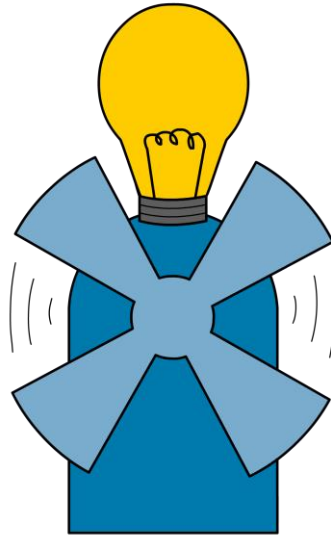


studirko.com.hr

predstavlja:



**C# za FPZ
(13.01.2025.)**

Za studirko.com.hr napisao:

Slaven Špigl

Uvodna riječ

Poštovani,

skripta koju čitate dio je popratnih materijala koji dolaze uz serijal video tutorijala za učenje programiranja. Cijeli serijal „C# za FPZ“, kao i posljednju inačicu ove skripte, možete naći na web stranici studirko.com.hr u sekciji „Tutorijali“.

Svi zadaci su prepisani (uz minimalne izmjene) s ispita i kolokvija iz kolegija „Algoritmi i programiranje“ na Fakultetu prometnih znanosti u Zagrebu. Izmjene se uglavnom odnose na ispravku gramatičkih i logičkih grešaka (krive formule, pogrešan način pisanja naredbi i sl.) koje su se pojavile u ispitima, ali ima i manjih izmjena koje sam radio radi očuvanja konzistentnosti pisanja zadataka.

Neke ispitne rokove nisam mogao pronaći sa svim zadacima pa sam, radi popunjavanja rupa u gradivu, sam smislio neke od zadataka (npr. peti zadatak iz ispitnog roka 13.07.17. - grupa B).

Skripta će se mijenjati i dopunjavati s ispitima koje uspijem nabaviti, tako da pratite o kojoj se inačici skripte radi kako bi imali najnovije materijale. Verzija skripte označena je na naslovnoj stranici datumom zadnje izmjene i brojem izdanja skripte.

Ako nađete grešku u skripti, ako imate prijedlog, kritiku ili zadatak koji bi htjeli znati riješiti javite nam se preko kontakt forme na stranici studirko.com.hr.

Želim Vam ugodno učenje,

Slaven Špigl

Sadržaj

Ispitni rokovi – zadaci	1
Ispit #1 (XX.XX.XX. - grupa A)	1
Ispit #2 (XX.XX.XX. - grupa B)	2
Ispit #3 (XX.XX.XX. - grupa C)	4
Ispit #4 (23.06.14. - grupa A)	6
Ispit #5 (23.06.14. - grupa B)	8
Ispit #6 (23.06.14. - grupa C) i (03.07.14 – grupa A)	9
Ispit #7 (10.07.14. - grupa A)	10
Ispit #8 (02.07.15. - grupa A)	11
Ispit #9 (16.05.16. - grupa A)	13
Ispit #10 (16.05.16. - grupa B)	14
Ispit #11 (26.06.17. - grupa A)	15
Ispit #12 (26.06.17. - grupa B)	16
Ispit #13 (06.07.17. - grupa A)	17
Ispit #14 (06.07.17. - grupa B)	19
Ispit #15 (13.07.17. - grupa A)	21
Ispit #16 (13.07.17. - grupa B)	23
Ispit #17 (21.06.18. – grupa A)	25
Ispit #18 (21.06.18. – grupa B).....	26
Ispit #19 (05.07.18. – grupa A)	27
Ispit #20 (05.07.18. – grupa B).....	29
Ispit #21 (12.07.18. – grupa A)	31
Ispit #22 (12.07.18. – grupa B).....	33
Ispit #23 (06.09.18. - grupa A)	35
Ispit #24 (06.09.18. - grupa B)	37
Ispit #25 (17.11.18. - grupa A)	39
Ispit #26 (07.02.19. - grupa A)	40
Ispit #27 (21.02.19. - grupa A)	41
Ispit #28 (13.04.19. - grupa A)	42
Ispit #29 (13.06.19. - grupa A)	44
Ispit #30 (13.06.19. - grupa B)	46
Ispit #31 (02.07.19. - grupa A)	48

Ispit #32 (02.07.19. - grupa B)	50
Ispit #33 (11.07.19. - grupa A)	52
Ispit #34 (11.07.19. - grupa B)	55
Ispit #35 (29.08.19. - grupa A)	59
Ispit #36 (29.08.19. - grupa B)	61
Ispit #37 (06.02.20. - grupa A)	63
Ispit #38 (20.02.20. - grupa A)	65
Ispit #39 (02.06.20. - grupa A)	68
Ispit #40 (15.06.20. - grupa A)	73
Ispit #41 (26.06.20. - grupa A)	76
Ispit #42 (09.07.20. - grupa A)	79
Ispit #43 (09.07.20. - grupa B)	83
Ispit #44 (09.07.20. - grupa C)	86
Ispit #45 (31.08.20. - grupa A)	88
Ispit #46 (09.09.20. - grupa A)	90
Ispit #47 (09.09.20. - grupa B)	92
Ispit #48 (21.11.20. - grupa B)	94
Ispit #49 (04.02.21. - grupa A)	96
Ispit #50 (18.02.21. - grupa A)	99
Ispit #51 (17.04.21. - grupa A)	103
Ispit #52 (17.04.21. - grupa B)	106
Ispit #53 (17.04.21. - grupa C)	108
Ispit #54 (10.06.21. - grupa A)	110
Ispit #55 (10.06.21. - grupa B)	113
Ispit #56 (10.06.21. - grupa C)	116
Ispit #57 (28.06.21. - grupa A)	119
Ispit #58 (28.06.21. - grupa B)	121
Ispit #59 (28.06.21. - grupa C)	123
Ispit #60 (08.07.21. - grupa A)	125
Ispit #61 (08.07.21. - grupa B)	129
Ispit #62 (08.07.21. - grupa C)	132
Ispit #63 (02.09.21. - grupa A)	134
Ispit #64 (02.09.21. - grupa B)	137
Ispit #65 (02.09.21. - grupa C)	141

Ispit #66 (06.11.21. - grupa A)	143
Ispit #67 (03.02.22. - grupa A)	146
Ispit #68 (17.02.22. - grupa A)	149
Ispit #69 (26.03.22. - grupa A)	151
Ispit #70 (23.06.22. - grupa A)	155
Ispit #71 (23.06.22. - grupa B)	158
Ispit #72 (07.07.22. - grupa A)	162
Ispit #73 (07.07.22. - grupa B)	165
Ispit #74 (01.09.22. - grupa A)	168
Ispit #75 (01.09.22. - grupa B)	171
Ispit #76 (08.09.22. - grupa A)	174
Ispit #77 (08.09.22. - grupa B)	175
Ispit #78 (02.02.23. - grupa B)	179
Ispit #79 (16.02.23. - grupa A)	182
Ispit #80 (25.03.23. - grupa A)	187
Ispit #81 (15.06.23. - grupa A)	190
Ispit #82 (06.07.23. - grupa A)	193
Ispit #83 (06.07.23. - grupa B)	195
Ispit #84 (31.08.23. - grupa A)	197
Ispit #85 (07.09.23. - grupa B)	198
Ispit #88 (22.02.24. - grupa A)	201
Ispit #89 (06.04.24. - grupa A)	203
Ispit #90 (27.06.24.)	206
Ispit #91 (XX.XX.24. - demonstrature)	209
Ispit #92 (11.07.24. - demonstrature)	212
Ispit #93 (11.07.24.)	214
Ispit #94 (28.08.24. - demonstrature)	217
Ispit #95 (29.08.24.)	219
Ispit #96 (09.09.24. - demonstrature)	221
Ispit #97 (10.09.24.)	223
Dodatni zadaci	226
Ispitni rokovi – rješenja	227
Ispit #1 (XX.XX.XX. - grupa A)	227
Zadatak 1	227

Zadatak 2.....	227
Zadatak 3.....	228
Zadatak 4.....	228
Zadatak 5.....	229
Ispit #2 (XX.XX.XX. - grupa B)	230
Zadatak 1.....	230
Zadatak 2.....	231
Zadatak 3.....	231
Zadatak 4.....	232
Zadatak 5.....	233
Ispit #3 (XX.XX.XX. - grupa C)	235
Zadatak 1.....	235
Zadatak 2.....	236
Zadatak 3.....	236
Zadatak 4.....	237
Zadatak 5.....	238
Ispit #9 (16.05.16. - grupa A)	240
Zadatak 1.....	240
Zadatak 2.....	241
Zadatak 3.....	242
Zadatak 4.....	243
Zadatak 5.....	244
Ispit #10 (16.05.16. - grupa B)	245
Zadatak 1.....	245
Zadatak 2.....	246
Zadatak 3.....	247
Zadatak 4.....	248
Zadatak 5.....	249
Ispit #11 (26.06.17. - grupa A)	250
Zadatak 1.....	250
Zadatak 2.....	251
Zadatak 3.....	252
Zadatak 4.....	253
Zadatak 5.....	254

Ispit #12 (26.06.17. - grupa B)	255
Zadatak 1.....	255
Zadatak 2.....	256
Zadatak 3.....	257
Zadatak 4.....	258
Zadatak 5.....	259
Ispit #13 (06.07.17. - grupa A)	260
Zadatak 1.....	260
Zadatak 2.....	261
Zadatak 3.....	262
Zadatak 4.....	263
Zadatak 5.....	264
Ispit #14 (06.07.17. - grupa B)	265
Zadatak 1.....	265
Zadatak 2.....	266
Zadatak 3.....	267
Zadatak 4.....	268
Zadatak 5.....	269
Ispit #15 (13.07.17. - grupa A)	270
Zadatak 1.....	270
Zadatak 2.....	271
Zadatak 3.....	272
Zadatak 4.....	273
Zadatak 5.....	274
Ispit #16 (13.07.17. - grupa B)	275
Zadatak 1.....	275
Zadatak 2.....	276
Zadatak 3.....	277
Zadatak 4.....	278
Zadatak 5.....	279
Ispit #17 (21.06.18. – grupa A)	280
Zadatak 1.....	280
Zadatak 2.....	282
Zadatak 3.....	284

Zadatak 4.....	285
Zadatak 5.....	286
Ispit #18 (21.06.18. – grupa B).....	287
Zadatak 1.....	287
Zadatak 2.....	289
Zadatak 3.....	290
Zadatak 4.....	291
Zadatak 5.....	292
Ispit #19 (05.07.18. – grupa A)	293
Zadatak 1.....	293
Zadatak 2.....	294
Zadatak 3.....	295
Zadatak 4.....	296
Zadatak 5.....	297
Ispit #20 (05.07.18. – grupa B).....	298
Zadatak 1.....	298
Zadatak 2.....	299
Zadatak 3.....	300
Zadatak 4.....	301
Zadatak 5.....	302
Ispit #21 (12.07.18. – grupa A)	303
Zadatak 1.....	303
Zadatak 2.....	304
Zadatak 3.....	305
Zadatak 4.....	306
Zadatak 5.....	307
Ispit #22 (12.07.18. – grupa B).....	308
Zadatak 1.....	308
Zadatak 2.....	309
Zadatak 3.....	310
Zadatak 4.....	311
Zadatak 5.....	312
Ispit #23 (06.09.18. - grupa A)	313
Zadatak 1.....	313

Zadatak 2.....	314
Zadatak 3.....	315
Zadatak 4.....	316
Zadatak 5.....	317
Ispit #24 (06.09.18. - grupa B)	318
Zadatak 1.....	318
Zadatak 2.....	319
Zadatak 3.....	320
Zadatak 4.....	321
Zadatak 5.....	322
Ispit #25 (17.11.18. - grupa A)	323
Zadatak 1.....	323
Zadatak 2.....	324
Zadatak 3.....	325
Zadatak 4.....	326
Zadatak 5.....	327
Ispit #26 (07.02.19. - grupa A)	328
Zadatak 1.....	328
Zadatak 2.....	329
Zadatak 3.....	330
Zadatak 4.....	331
Zadatak 5.....	332
Ispit #27 (21.02.19. - grupa A)	333
Zadatak 1.....	333
Zadatak 2.....	334
Zadatak 3.....	335
Zadatak 4.....	336
Zadatak 5.....	337
Ispit #28 (13.04.19. - grupa A)	338
Zadatak 1.....	338
Zadatak 2.....	339
Zadatak 3.....	341
Zadatak 4.....	342
Zadatak 5.....	343

Ispit #29 (13.06.19. - grupa A)	344
Zadatak 1	344
Zadatak 2	345
Zadatak 3	346
Zadatak 4	348
Zadatak 5	349
Ispit #30 (13.06.19. - grupa B)	351
Zadatak 1	351
Zadatak 2	352
Zadatak 3	353
Zadatak 4	355
Zadatak 5	356
Ispit #31 (02.07.19. - grupa A)	357
Zadatak 1	357
Zadatak 2	358
Zadatak 3	359
Zadatak 4	360
Zadatak 5	361
Ispit #32 (02.07.19. - grupa B)	362
Zadatak 1	362
Zadatak 2	363
Zadatak 3	364
Zadatak 4	365
Zadatak 5	366
Ispit #33 (11.07.19. - grupa A)	367
Zadatak 1	367
Zadatak 2	368
Zadatak 3	369
Zadatak 4	370
Zadatak 5	372
Ispit #34 (11.07.19. - grupa B)	373
Zadatak 1	373
Zadatak 2	374
Zadatak 3	375

Zadatak 4.....	376
Zadatak 5.....	378
Ispit #35 (29.08.19. - grupa A)	379
Zadatak 1.....	379
Zadatak 2.....	380
Zadatak 3.....	381
Zadatak 4.....	382
Zadatak 5.....	383
Ispit #36 (29.08.19. - grupa B)	384
Zadatak 1.....	384
Zadatak 2.....	385
Zadatak 3.....	386
Zadatak 4.....	387
Zadatak 5.....	388
Ispit #38 (20.02.20. - grupa A)	389
Zadatak 1.....	389
Zadatak 2.....	390
Zadatak 3.....	392
Zadatak 4.....	393
Zadatak 5.....	394
Ispit #39 (02.06.20. - grupa A)	395
Zadatak 1.....	395
Zadatak 2.....	396
Zadatak 3.....	397
Zadatak 4.....	399
Zadatak 5.....	400
Ispit #40 (15.06.20. - grupa A)	402
Zadatak 1.....	402
Zadatak 2.....	403
Zadatak 3.....	405
Zadatak 4.....	407
Zadatak 5.....	409
Ispit #41 (26.06.20. - grupa A)	410
Zadatak 1.....	410

Zadatak 2.....	412
Zadatak 3.....	416
Zadatak 4.....	418
Zadatak 5.....	419
Ispit #42 (09.07.20. - grupa A)	421
Zadatak 1.....	421
Zadatak 2.....	423
Zadatak 3.....	424
Zadatak 4.....	426
Zadatak 5.....	427
Ispit #43 (09.07.20. - grupa B)	429
Zadatak 1.....	429
Zadatak 2.....	431
Zadatak 3.....	432
Zadatak 4.....	434
Zadatak 5.....	435
Ispit #44 (09.07.20. - grupa C)	437
Zadatak 1.....	437
Zadatak 2.....	439
Zadatak 5.....	440
Ispit #45 (31.08.20. - grupa A)	442
Zadatak 1.....	442
Zadatak 2.....	443
Zadatak 3.....	444
Zadatak 4.....	445
Zadatak 5.....	446
Ispit #46 (09.09.20. - grupa A)	447
Zadatak 1.....	447
Zadatak 2.....	448
Zadatak 3.....	449
Zadatak 4.....	450
Zadatak 5.....	452
Ispit #47 (09.09.20. - grupa B)	453
Zadatak 1.....	453

Zadatak 2.....	454
Zadatak 3.....	455
Zadatak 4.....	456
Zadatak 5.....	458
Ispit #48 (21.11.20. - grupa B)	459
Zadatak 1.....	459
Zadatak 2.....	460
Zadatak 3.....	461
Zadatak 4.....	462
Zadatak 5.....	464
Ispit #49 (04.02.21. - grupa A)	465
Zadatak 1.....	465
Zadatak 2.....	466
Zadatak 3.....	467
Zadatak 4.....	469
Zadatak 5.....	471
Ispit #50 (18.02.21. - grupa A)	473
Zadatak 1.....	473
Zadatak 2.....	474
Zadatak 3.....	475
Zadatak 4.....	477
Zadatak 5.....	479
Ispit #51 (17.04.21. - grupa A)	480
Zadatak 1.....	480
Zadatak 2.....	483
Zadatak 3.....	485
Zadatak 4.....	486
Zadatak 5.....	487
Ispit #60 (08.07.21. - grupa A)	489
Zadatak 1.....	489
Zadatak 2.....	490
Zadatak 3.....	491
Zadatak 4.....	494
Zadatak 5.....	495

Ispit #61 (08.07.21. - grupa B)	498
Zadatak 1.....	498
Zadatak 2.....	499
Zadatak 3.....	501
Zadatak 4.....	503
Zadatak 5.....	505
Ispit #62 (08.07.21. - grupa C)	507
Zadatak 1.....	507
Zadatak 2.....	509
Zadatak 3.....	510
Zadatak 4.....	511
Ispit #63 (02.09.21. - grupa A)	513
Zadatak 1.....	513
Zadatak 2.....	514
Zadatak 3.....	519
Zadatak 4.....	520
Zadatak 5.....	522
Ispit #64 (02.09.21. - grupa B)	524
Zadatak 1.....	524
Zadatak 2.....	525
Zadatak 3.....	528
Zadatak 4.....	529
Zadatak 5.....	531
Ispit #65 (02.09.21. - grupa C)	534
Zadatak 1.....	534
Zadatak 2.....	535
Zadatak 3.....	537
Zadatak 4.....	538
Zadatak 5.....	540
Ispit #76 (08.09.22. - grupa A)	542
Zadatak 2.....	542
Zadatak 3.....	544
Ispit #77 (08.09.22. - grupa B)	546
Zadatak 1.....	546

Zadatak 2.....	547
Zadatak 3.....	549
Zadatak 4.....	551
Zadatak 5.....	554
Ispit #78 (02.02.23. - grupa B)	556
Zadatak 1.....	556
Zadatak 2.....	557
Zadatak 3.....	560
Zadatak 5.....	561
Ispit #79 (02.02.23. - grupa A)	563
Zadatak 1.....	563
Zadatak 2.....	566
Zadatak 5.....	568
Ispit #81 (15.06.23. - grupa A)	571
Zadatak 1.....	571
Zadatak 2.....	572
Zadatak 3.....	573
Zadatak 4.....	575
Zadatak 5.....	577
Ispit #82 (06.07.23. - grupa A)	578
Zadatak 1.....	578
Zadatak 2.....	579
Zadatak 3.....	580
Zadatak 4.....	582
Ispit #83 (06.07.23. - grupa B)	583
Zadatak 1.....	583
Zadatak 2.....	584
Zadatak 3.....	585
Ispit #84 (31.08.23. - grupa A)	586
Zadatak 2.....	586

Ispitni rokovi – zadaci

Ispit #1 (XX.XX.XX. - grupa A)

1. Nadopunite tablicu s odgovarajućim tipovima podataka potrebnim za spremanje navedenih vrijednosti.

Vrijednost	Tip podatka
"false"	
"2"	
2	
'2'	

2. Napišite konzolnu aplikaciju koja učitava jedan cijeli broj i provjerava da li je taj broj djeljiv s 3. Ako je broj djeljiv ispisuje poruku „Broj je djeljiv s 3“, a ako nije „Broj nije djeljiv s 3“.

3. Korištenjem algoritma za jednostavno sortiranje umetanjem (insertion sort) nadopunite ostatak koraka algoritma.

91	33	56	48	88	66	6
91	33	56	48	88	66	6

4. Napišite konzolnu aplikaciju koja traži unos središnjeg kuta i radijusa kružnice (realni brojevi) te računa površinu kružnog isječka prema formuli:

$$P = \frac{r^2 \pi \alpha}{360}$$

5. Kreirajte polje od 200 elemenata koji su cijeli brojevi generatorom slučajnih brojeva od 20 do 30 i ispišite zadnji element polja.



Ispit #2 (XX.XX.XX. - grupa B)

1. Napišite konzolnu aplikaciju koja učitava dva cijela broja s tipkovnice (a i b). Ako je broj a neparan program ispisuje razliku brojeva a i b . Inače ispisuje umnožak učitanih brojeva. Ispis rezultata mora biti u formatu kao u primjeru.

Primjer: ako je za a učitani broj 0, a za b je učitani broj 2, ispis mora biti „Razlika brojeva 9 i 2 je 7“.

2. Napišite konzolnu aplikaciju koja ispisuje sve brojeve između 1 i n koji nisu djeljivi ni sa 3 ni sa 7, pri čemu se prirodan broj n učitava s tipkovnice.

3. Napišite metodu *ZbrojKvadrata* koja kao argument prima tri realna broja te vraća zbroj kvadrata predanih brojeva. Potrebno je napisati deklaraciju metode *ZbrojKvadrata*.

4. Nadopunite sljedeći program na način da učitajte cijeli broj n s tipkovnice te prebrojite koliko elemenata u polju je veće ili jednako n .

Primjer: ako se s tipkovnice učitava broj 6 potrebno je ispisati „2 broja u polju su veća ili jednaka 6“.

Koristite petlje prema vlastitom izboru.

```
static void Main(string[] args)
{
    int[] polje = new int[5] { 1, 2, 6, 7, 4 };

    //Nadopuni

}
```



5. Nadopunite kod u glavnoj metodi tako da se doda jedan novi student JMBAG-a 0135123123, Pero Perić. Uvećajte njegov broj ECTS bodova za 5.

```
static void Main(string[] args)
{
    //Nadopuni

}

public class Student
{
    public string JMBAG { get; set; }
    public string Prezime { get; set; }
    public string Ime { get; set; }
    public int Ects { get; set; }

    public Student(string jmbag, string prezime, string ime)
    {
        JMBAG = jmbag;
        Prezime = prezime;
        Ime = ime;
        Ects = 0;
    }

    public void UvecajEcts(int ects)
    {
        Ects += ects;
    }
}
```



Ispit #3 (XX.XX.XX. - grupa C)

1. Napišite konzolnu aplikaciju koja učitava realan broj r koji predstavlja polumjer kruga. Ako je polumjer kruga manji ili jednak nuli aplikacija ispisuje "Niste unijeli dobar polumjer!", inače ispisuje površinu i opseg kruga kao u primjeru.

Npr. ako je za r učitani broj 2.5 ispis mora biti sljedeći: "Površina kruga polumjera 2.5 je $P = 19.635$, a opseg je $O = 15.7!$ ".

2. Napišite metodu *Suma* koja kao argument prima dva prirodna broja te ispisuje sumu svih brojeva između dva predana broja (granice su uključujuće). Potrebno je napisati deklaraciju i implementaciju metode *Suma*.

3. Napišite konzolnu aplikaciju koja učitava sadržaj datoteke „*data.txt*“ u kojoj su upisani cijeli brojevi svaki u zasebnom retku. Program treba na kraju ispisati sumu svih učitanih cijelih brojeva.

Napomena: nije potrebno u program dodati *using System.IO* niti *try-catch* blok.

4. Napišite konzolnu aplikaciju koja učitava 1000 realnih vrijednosti u jednodimenzionalno polje te ispisuje koliko je elemenata polja negativno (prebrojava broj negativnih vrijednosti).



5. Nadopunite sljedeći kod unutar Main metode na način da stvorite novi kolegij naziva „ALGPRO“ na zavodu „ITS“ koji nosi 7 ECTS-a i ima 270 studenata. Potom pozovite metodu s kojom će se povećati broj studenata za 1.

```
static void Main(string[] args)
{
    //Nadopuni

}

public class Kolegij
{
    private string naziv;
    private string zavod;
    private int ects;
    private int brojStudenata;

    public Kolegij(string ZAVOD, string NAZIV, int ECTS,int BROJSTUDENATA)
    {
        naziv = NAZIV;
        zavod = ZAVOD;
        ects = ECTS;
        brojStudenata = BROJSTUDENATA;
    }

    public void UvecajBrojStudenata1(int brStud)
    {
        brojStudenata += brStud;
    }

    private void UvecajBrojStudenata2()
    {
        brojStudenata++;
    }
}
```



Ispit #4 (23.06.14. - grupa A)

1. Što će biti rezultat izvršavanja programa ispod:

```
static void Main(string[] args)
{
    if (5 < 7 && (4 != 2 && 4 >= 4))
    {
        Console.WriteLine("Istina");
    }
    else
    {
        Console.WriteLine("Laž");
    }

    Console.ReadKey();
}
```

2. Napišite konzolnu aplikaciju koja učitava dva prirodna broja. Unesene brojeve je potrebno zbrojiti i ispis formatirati kao u primjeru.

Primjer: ako su uneseni brojevi 2 i 3 potrebno je ispisati „2 + 3 = 5“.

3. Napišite konzolnu aplikaciju koja učitava prirodan broj i ispisuje je li uneseni broj djeljiv s brojem 3 ili s brojem 8.

4. Nadopunite sljedeći program na način da pronađete aritmetičku sredinu vrijednosti elemenata u polju i ispišete tu vrijednost u konzolu.

```
static void Main(string[] args)
{
    int[] polje = new int[4] { 6, 8, 2, 3 };

    //Nadopuni
}
```



5. Nadopunite sljedeći kod metodom "*Ubrzaj*" unutar klase *Auto* na način da njezini parametri odgovaraju pozivu unutar *Main* dijela programa. U tijelu metode potrebno je povećati atribut *brzina* za vrijednost parametra kojeg metoda primi.

U metodi *Ubrzaj* potrebno je i ispisati vrijednost za koju se povećao atribut *brzina*, kao i vrijednost atributa *brzina* nakon uvećavanja.

```
static void Main(string[] args)
{
    Auto auto = new Auto();
    auto.Ubrzaj(8);
}

public class Auto
{
    public int id { get; set; }
    public int Model { get; set; }
    public int Marka { get; set; }
    public int Brzina { get; set; }

    //Nadopuni

}
```



Ispit #5 (23.06.14. - grupa B)

1. Što će biti rezultat izvršavanja programa ispod:

```
static void Main(string[] args)
{
    if (5 > 7 || (4 != 2 && 4 > 4))
    {
        Console.WriteLine("Istina");
    }
    else
    {
        Console.WriteLine("Laž");
    }

    Console.ReadKey();
}
```

2. Napišite konzolnu aplikaciju koja učitava tri realna broja koja predstavljaju katete i hipotenuzu pravokutnog trokuta. Potrebno je izračunati i ispisati opseg i površinu tog trokuta u proizvoljnom formatu.

3. Napišite konzolnu aplikaciju koja učitava prirodan broj i ispisuje je li uneseni broj djeljiv s brojem 3 i s brojem 8.

4. Nadopunite sljedeći program tako da elemente unutar polja sortirate od najmanjeg do najvećeg. Nakon sortiranja ispišite sve elemente iz polja na ekran tako da svaki element bude ispisan u novom retku.

```
static void Main(string[] args)
{
    int[] polje = new int[5] { 4, 5, 6, 7, 2 };

    //Nadopuni

}
```

5. Napišite konzolnu aplikaciju koja učitava rečenicu i ispisuje koliko puta se u upisanoj rečenici pojavljuje samoglasnik „a“. Napomena: potrebno je brojati i veliko i malo slovo.



Ispit #6 (23.06.14. - grupa C) i (03.07.14 – grupa A)

1. Što će biti rezultat izvršavanja programa ispod:

```
static void Main(string[] args)
{
    if (5 > 7 || (4 != 2 && 4 >= 4))
    {
        Console.WriteLine("Istina");
    }
    else
    {
        Console.WriteLine("Laž");
    }
}
```

2. Napišite konzolnu aplikaciju koja učitava dva realna broja, izračunava njihov umnožak te ga zaokružuje na najbliži cijeli broj. Rezultat ispišite u konzolu.

3. Napišite konzolnu aplikaciju koja učitava koordinate točaka T_1 i T_2 . Svaka točka predstavljena je s dvije koordinate x i y koje su realni brojevi. Izračunajte udaljenost između točaka sljedećom formulom $\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$ i ispišite izračunati udaljenost u konzolu.

Napomena: zadana formula je bila nešto drugačija: $\sqrt{(x_1 - y_1) + (x_2 - y_2)}$. Odlučio sam se napisati ispravnu formulu koja je standardna matematička formula za računanje udaljenosti točaka u ravni. To je tzv. formula za izračun euklidske udaljenosti.

4. Nadopunite sljedeći program na način da pronađete maksimalnu vrijednost elemenata u polju i ispišete tu vrijednost u konzolu.

```
static void Main(string[] args)
{
    int[] polje = new int[6] { 12, 10, 6, 8, 2, 1 };

    //Nadopuni

}
```

5. Izradite program koji preko tipkovnice učitava 20 cijelih brojeva i sprema ih u listu. Zatim je potrebno iz liste obrisati prva dva elementa te ispisati ostatak sadržaja liste u konzolu.



Ispit #7 (10.07.14. - grupa A)

Napomena: nedostaje prvi zadatak

2. Napišite konzolnu aplikaciju koja učitava jedan cijeli broj i provjerava da li je taj broj djeljiv s 3. Ako je broj djeljiv ispisuje poruku „*Broj je djeljiv s 3*“, a ako nije „*Broj nije djeljiv s 3*“.

3. Napišite konzolnu aplikaciju u kojoj ćete kreirati polje koje je potrebno napuniti s proizvoljnim brojevima (nije potreban unos s tipkovnice i treba biti najmanje 5 brojeva).

Polje je potrebno sortirati *insertion sort* algoritmom u posebnoj metodi (ime metode je proizvoljno).

Metoda za sortiranje ujedno i ispisuje sve korake sortiranja kako bi proces sortiranja bio vidljiv u konzoli.

4. Napišite konzolnu aplikaciju koja traži unos središnjeg kuta i radijusa kružnice (realni brojevi) te računa površinu kružnog isječka prema formuli:

$$P = \frac{r^2 \pi \alpha}{360}$$

5. Kreirajte polje od 200 elemenata koji su cijeli brojevi generatorom slučajnih brojeva od 20 do 30 i ispišite zadnji element polja.



Ispit #8 (02.07.15. - grupa A)

1. Što će biti rezultat izvršavanja programa ispod:

```
static void Main(string[] args)
{
    int a = 5;
    int b = Smanji(a);
    if (a > b)
    {
        Console.WriteLine(a - b);
    }
    else
    {
        Console.WriteLine(a + b);
    }
}

static int Smanji(int c)
{
    int g = c - 10;
    return g;
}
```

2. Napišite konzolnu aplikaciju koja učitava dva cijela broja i ispisuje na konzolu korijen prvog broja i kvadrat drugog broja. Format ispisa rezultata mora izgledati kao na sljedećem primjeru.

Primjer: ako su uneseni brojevi 4 i 3 potrebno je ispisati „Korijen broja 4 je 2, kvadrat broja 3 je 9“.

3. Napišite konzolnu aplikaciju koja ispisuje sve neparne brojeve između broja a i broja b (ako su a i b neparni brojevi uključeni su u ispis neparnih brojeva). Cijeli brojevi a i b učitavaju se preko tipkovnice. Format ispisa rezultata mora izgledati kao na sljedećem primjeru.

Primjer: ako su uneseni brojevi 2 i 10 potrebno je ispisati „Neparni brojevi između 2 i 10 su 3,5,7,9“.

4. Napišite konzolnu aplikaciju koja će tražiti unos pet cijelih brojeva u listu. Nakon unosa četvrtog broja potrebno je obrisati drugi broj iz liste, ali potrebno je nastaviti s učitavanjem sljedećeg (petog) broja.

Na kraju je potrebno ispisati sadržaj liste u proizvoljnom formatu.



5. Nadopunite sljedeći zadatak tako što ćete upisati ime i prezime studenta *Pero Perić*, njegov JMBAG *0135231212* i dodati mu 5 ECTS bodova.

```
static void Main(string[] args)
{
    //Nadopuni

}

public class Student
{
    public string JMBAG { get; set; }
    public string Prezime { get; set; }
    public string Ime { get; set; }
    public int Ects { get; set; }

    public Student(string jmbag, string prezime, string ime)
    {
        JMBAG = jmbag;
        Prezime = prezime;
        Ime = ime;
        Ects = 0;
    }

    public void UvecajEcts(int ects)
    {
        Ects += ects;
    }
}
```



Ispit #9 (16.05.16. - grupa A)

1. Napišite konzolnu aplikaciju koja učitava polumjer kruga r (decimalni broj).
Ako je polumjer kruga $r \leq 0$, aplikacija ispisuje „Niste unijeli dobar polumjer!“.
Ako je polumjer kruga $r == 2.5$, aplikacija ispisuje površinu kruga 19.63“.
Ako je polumjer kruga $r > 0 \ \&\& \ r \neq 2.5$, računa površinu kruga po unesenom polumjeru.
2. Napišite konzolnu aplikaciju koja ispisuje sve prirodne brojeve između n i m (učitavaju se s tipkovnice).
3. Napišite metodu Umnožak koja kao argument prima prirodni broj te vraća umnožak svih prirodnih brojeva do tog broja. Potrebno je napisati i deklaraciju metode Umnožak.
4. Nadopunite sljedeći program na način da pronađete sumu parnih vrijednosti elemenata u polju i ispišite je u konzolu (petlja po izboru).

```
static void Main(string[] args)
{
    int[] polje = new int[5] { 8, 2, 1, 9, 7 };

    //Nadopuni

}
```

5. Nadopunite sljedeći zadatak tako što ćete upisati ime i prezime studenta Pero Perić, njegov JMBAG 0135231212 i dodati mu 5 ECTS bodova.

```
static void Main(string[] args)
{
    //Nadopuni

}

public class Student
{
    public string JMBAG { get; set; }
    public string Prezime { get; set; }
    public string Ime { get; set; }
    public int Ects { get; set; }

    public Student(string jmbag, string prezime, string ime)
    {
        JMBAG = jmbag;
        Prezime = prezime;
        Ime = ime;
        Ects = 0;
    }

    public void UvecajEcts(int ects)
    {
        Ects += ects;
    }
}
```



Ispit #10 (16.05.16. - grupa B)

1. Napišite konzolnu aplikaciju koja učitava dva broja a i b.

Ako je broj a $\geq$ b, treba izračunati njihovu sumu.

Ako je a < b, onda treba izračunati njihovu razliku.

2. Napišite konzolnu aplikaciju koja ispisuje sve parne brojeve između 1 i n. Prirodni broj n učitajte s tipkovnice.

3. Napišite metodu Suma koja kao argument prima dva prirodna broja te vraća sumu svih brojeva između dva prirodna broja. Potrebno je napisati i deklaraciju metode Suma.

4. Nadopunite sljedeći program na način da pronađete maksimalnu vrijednost elemenata u polju i ispišete tu vrijednost u konzolu.

```
static void Main(string[] args)
{
    int[] polje = new int[6] { 12, 10, 6, 8, 2, 1 };

    //Nadopuna

}
```

5. Nadopunite sljedeći kod metodom "Ubrzaj" unutar klase Auto na način da njezini parametri odgovaraju pozivu unutar Main dijela programa. U tijelu metode potrebno je povećati atribut brzinu za vrijednost parametra kojeg metoda primi.

```
static void Main(string[] args)
{
    Auto auto1 = new Auto();
    auto1.Ubrzaj(5);

    Console.ReadKey();
}

public class Auto
{
    public int id { get; set; }
    public int Model { get; set; }
    public int Marka { get; set; }
    public int Brzina { get; set; }

    //Nadopuni

}
```



Ispit #11 (26.06.17. - grupa A)

1. Napišite konzolnu aplikaciju koja učitava broj bodova s ispita unutar intervala [0,5]. Ako učitani bodovi nisu unutar zadanog intervala program ispisuje "Niste unijeli bodove u intervalu [0,5]".

Ako su ispravno učitani bodovi potrebno je ispisati odgovarajuću poruku: ako je broj bodova unutar intervala $[0,2>$ (0 uključujuće, 2 isključujuće) potrebno je ispisati "Student je pao na ispitu!". Inače je potrebno ispisati "Student je prošao na ispitu!".

2. Napišite metodu Ispis koja ispisuje sve brojeve djeljive s 4 u intervalu $[m - n]$, pri čemu su m i n argumenti koji se predaju metodi. Nije potrebno vršiti provjeru da li su učitane ispravne granice intervala. Potrebno je napisati implementaciju i deklaraciju metode Ispis. Poziv unutar Main metode nije potreban.

3. Napišite konzolnu aplikaciju koja učitava sadržaj datoteke "dat.txt" u kojoj su upisani cijeli brojevi svaki u zasebnom retku. Program treba ispisati sadržaj datoteke s ispisom svakog od učitanih cijelih brojeva u zasebnom retku.

4. Napišite konzolnu aplikaciju koja učitava 10 cjelobrojnih vrijednosti u jednodimenzionalno polje, te računa i ispisuje prosječnu vrijednost učitanih elemenata polja.

5. Nadopunite sljedeći kod s metodama UvecajKilometre i PromijeniVlasnika tako da njihovi parametri odgovaraju pozivu iz Main metode. Metoda UvecajKilometre treba uvećati kilometre za zadani broj koji se predaje preko parametra, dok metoda PromijeniVlasnika treba promijeniti ime vlasnika u ime koje se predaje preko parametra.

```
static void Main(string[] args)
{
    Auto a = new Auto("Insignia", 50000, "Marko");
    a.UvecajKilometre(10000);
    a.PromijeniVlasnika("Ivo");
}

public class Auto
{
    private string imeAuta;
    private int kilometri;
    private string imeVlasnika;

    public Auto(string IMEAUTA, int KILOMETRI, string IMEVLASNIKA)
    {
        imeAuta = IMEAUTA;
        kilometri = KILOMETRI;
        imeVlasnika = IMEVLASNIKA;
    }

    //Nadopuni
}
```



Ispit #12 (26.06.17. - grupa B)

1. Napišite konzolnu aplikaciju koja učitava dva cijela broja s tipkovnice, a i b. Ako je broj a manji ili jednak b program ispisuje sumu brojeva, a ako je a veći od b program ispisuje razliku brojeva. Ispis rezultata mora biti u formatu kao u primjeru. Npr. ako je za a učitani broj 8, a za b je učitani broj 3 ispis mora biti sljedeći: "Razlika brojeva 8 i 3 je 5."
2. Napišite metodu Suma koja kao argument prima dva prirodna broja m i n te vraća sumu svih brojeva između ta dva broja (granice su uključujuće). Nije potrebno vršiti provjeru da li su učitani ispravni brojevi. Potrebno je napisati implementaciju i deklaraciju metode Suma. Poziv unutar Main metode nije potreban.
3. Napišite konzolnu aplikaciju koja učitava realan broj x s tipkovnice. U datoteku naziva "dat.txt" potrebno je zapisati dvije vrijednosti: vrijednost učitanih broja x i dvostruko veću vrijednost učitanih broja x. Svaki zapis mora biti u zasebnom retku.
4. Napišite konzolnu aplikaciju koja učitava 10 cjelobrojnih vrijednosti u jednodimenzionalno polje, te pronalazi i ispisuje minimalnu vrijednost učitanih elemenata polja.
5. Nadopunite sljedeći kod odgovarajućim konstruktorom unutar klase Student na način da njezini parametri odgovaraju pozivu unutar Main metode.

```
static void Main(string[] args)
{
    Student st = new Student("Ivo", "Ivić", "ALGPRO", 3);
}

public class Student
{
    private string Ime;
    private string Prezime;
    private string Kolegij;
    private int Ocjena;

    //Nadopuni

}
```



Ispit #13 (06.07.17. - grupa A)

1. Napišite konzolnu aplikaciju koja učitava dva realna broja D i a koji predstavljaju diskriminantu i vrijednost prvog koeficijenta kvadratne funkcije.

Ako su vrijednosti D i a manje od nule potrebno je ispisati: "Za D i a funkcija ne siječe x-os te su joj sve vrijednosti negativne!".

Ako je D manje od nule i a veće od nule potrebno je ispisati: "Za D i a funkcija ne siječe x-os te su joj sve vrijednosti pozitivne!".

Inače je potrebno ispisati: "Funkcija siječe x-os!".

Primjer: ako se za D učitava vrijednost -1.25 , a za a vrijednost 0.5 , potrebno je ispisati:

"Za $D = -1.25$ i $a = 0.5$ funkcija ne siječe x-os te su joj sve vrijednosti pozitivne!".

2. Napišite metodu `PotenciranjeRazlike` koja kao argument prima tri realna broja. Metoda razliku prva dva broja potencira s trećim brojem te vraća rezultat izvršene operacije.

Potrebno je napisati deklaraciju i implementaciju metode `PotenciranjeRazlike`.

3. Napišite konzolnu aplikaciju koja učitava sadržaj datoteke "data.txt" u kojoj su upisani realni brojevi svaki u zasebnom retku. Program treba na kraju ispisati sumu svih učitanih realnih brojeva.

Napomena: nije potrebno u programu dodati `using System.IO`.

4. Napišite konzolnu aplikaciju koja učitava 1000 cjelobrojnih vrijednosti u jednodimenzionalno polje te ispisuje one indekse polja čija je vrijednost jednaka 0.



5. Nadopunite sljedeći kod unutar Main metode na način da stvorite novi grad "Zagreb" s koordinatama x=15.9773, y=45.8129 i brojem stanovnika 800 000.

Potom pozovite metodu koja će ispisati informacije vezane za stvoreni grad.

```
static void Main(string[] args)
{
    //Nadopuni

}

public class Grad
{
    private string naziv;
    private double x;
    private double y;
    private int brojStanovnika;

    public Grad(double X, double Y, string NAZIV, int BROJSTANOVNIKA)
    {
        naziv = NAZIV;
        x = X;
        y = Y;
        brojStanovnika = BROJSTANOVNIKA;
    }

    private void ispis1()
    {
        Console.WriteLine("Naziv grada: " + naziv + ", x= " + x + ", y= "
            + y + ", brSt= " + brojStanovnika);
    }

    public void ispis2()
    {
        Console.WriteLine("Naziv grada: " + naziv + ", x= " + x + ", y= "
            + y + ", brSt= " + brojStanovnika);
    }
}
```



Ispit #14 (06.07.17. - grupa B)

1. Napišite konzolnu aplikaciju koja učitava dva realna broja x i y koji predstavljaju točku $T(x,y)$ u kartezijevom koordinatnom sustavu. Program treba ispisati da li se točka nalazi u prvom, drugom ili niti jednom od navedenih kvadranta. Format ispisa mora biti kao u primjeru.

Primjer: ako se za x učitava vrijednost -3.2 , a za y vrijednost -2 potrebno je ispisati: "Točka $T(-3.2,-2)$ se ne nalazi niti u prvom niti u drugom kvadrantu!"

Napomena: pripadnost točke nekom kvadrantu određuje se na temelju pozitivnih i negativnih vrijednosti x i y .

2. Napišite metodu ZbrojPotencija koja kao argument prima tri realna broja. Metoda prvi i drugi potencira s trećim brojem, potom navedene vrijednosti zbraja te ispisuje rezultat u konzolu.

Potrebno je napisati deklaraciju i implementaciju metode ZbrojPotencija.

3. Napišite konzolnu aplikaciju koja redom učitava 10 cjelobrojnih vrijednosti s konzole i odmah iz zapisuje u datoteku "podaci.txt".

4. Napišite konzolnu aplikaciju koja učitava 1000 realnih vrijednosti u jednodimenzionalno polje, te ispisuje sve elemente polja čiji su indeksi parni brojevi.



5. Nadopunite sljedeći kod unutar Main metode na način da stvorite novi kolegij naziva "ALGPRO" na zavodu "ITS" koji nosi 7 ECTS-a i ima 270 studenata. Potom pozovite metodu kojom će se povećati broj studenata za 1.

```
static void Main(string[] args)
{
    //Nadopuni

    Console.ReadKey();
}

public class Kolegij
{
    private string naziv;
    private string zavod;
    private int ects;
    private int brojStudenata;

    public Kolegij(string ZAVOD, int ECTS, string NAZIV, int
    BROJSTUDENATA)
    {
        naziv = NAZIV;
        zavod = ZAVOD;
        ects = ECTS;
        brojStudenata = BROJSTUDENATA;
    }

    public void UvecajBrojStudenata1(int brStud)
    {
        brojStudenata += brStud;
    }

    private void UvecajBrojStudenata2()
    {
        brojStudenata++;
    }
}
```



Ispit #15 (13.07.17. - grupa A)

1. Napišite konzolnu aplikaciju koja učitava dva realna broja r_1 i r_2 s tipkovnice koji predstavljaju polumjere odgovarajućih krugova s površinama P_1 i P_2 .

Ako je r_1 veće ili jednako r_2 potrebno je izračunati površinu kružnog vijenca $P = P_1 - P_2$, inače je potrebno izračunati površinu kružnog vijenca $P = P_2 - P_1$.

Format ispisa mora biti kao što je navedeno u primjeru.

Npr. ako je $r_1 = 3$, a $r_2 = 2$ potrebno je ispisati:

"Površina kružnog vijenca koju čine krugovi polumjera $r_1 = 3$ i $r_2 = 2$ je $P = 15.708$ ".

2. Napišite metodu Umnožak koja kao argument prima prirodan broj te ispisuje umnožak svih prirodnih brojeva do uključujući tog broja.

Potrebno je napisati deklaraciju i implementaciju metode Umnožak.

3. Napišite konzolnu aplikaciju koja redom učitava 10 realnih vrijednosti s konzole i odmah ih zapisuje u datoteku "data.txt".

Napomena: nije potrebno u program dodati `Using.System.IO` niti `try-catch` blok.

4. Napišite konzolnu aplikaciju koja učitava 1000 cijelih vrijednosti u jednodimenzionalno polje, te ispisuje sumu onih elemenata koji su djeljivi sa 3.



5. Nadopunite sljedeći kod unutar Main metode na način da stvorite novi grad "Zagreb" s koordinatama x=15.9773, y=45.8129 i brojem stanovnika 800 000. Potom pozovite metodu koja će ispisati informacije vezane za stvoreni grad.

```
static void Main(string[] args)
{
    //Nadopuni

    Console.ReadKey();
}

public class Grad
{
    private string naziv;
    private double x;
    private double y;
    private int brojStanovnika;

    public Grad(double X, double Y, string NAZIV, int BROJSTANOVNIKA)
    {
        naziv = NAZIV;
        x = X;
        y = Y;
        brojStanovnika = BROJSTANOVNIKA;
    }

    private void ispis1()
    {
        Console.WriteLine("Naziv grada: " + naziv + ", x= " + x + ", y= " + y
            + ", brSt= " + brojStanovnika);
    }

    public void ispis2()
    {
        Console.WriteLine("Naziv grada: " + naziv + ", x= " + x + ", y= " + y
            + ", brSt= " + brojStanovnika);
    }
}
```



Ispit #16 (13.07.17. - grupa B)

1. Napišite konzolnu aplikaciju koja učitava realan broj r koji predstavlja polumjer kruga. Ako je polumjer kruga manji ili jednak nuli aplikacija ispisuje "Niste unijeli dobar polumjer!", inače ispisuje površinu i opseg kruga kao u primjeru.

Npr. ako je za r učitani broj 2.5 ispis mora biti sljedeći: "Površina kruga polumjera 2.5 je $P = 19.625$, a opseg je $O = 15.7!$ ".

2. Napišite metodu Suma koja kao argument prima dva prirodna broja te ispisuje sumu svih brojeva između dva predana broja (granice su uključujuće).

Potrebno je napisati deklaraciju i implementaciju metode Suma.

3. Napišite konzolnu aplikaciju koja učitava sadržaj datoteke "data.txt" u kojoj su upisani cijeli brojevi svaki u zasebnom retku.

Program treba na kraju ispisati sumu svih učitanih brojeva.

Napomena: nije potrebno u program dodati Using.System.IO niti try-catch blok.

4. Napišite konzolnu aplikaciju koja učitava 1000 realnih vrijednosti u jednodimenzionalno polje, te ispisuje koliko je elemenata polja negativno (prebrojava broj negativnih vrijednosti).



5. Nadopunite sljedeći kod unutar klase Grad na način da stvorite metodu PovecajBrojStanovnika koja će povećati broj stanovnika grada za 100 000.

Potom je instanci klase Grad stvorenoj u Main metodi potrebno povećati broj stanovnika pomoću metode PovecajBrojStanovnika.

```
static void Main(string[] args)
{
    Grad g = new Grad(15.9773, 45.8129, "Zagreb", 800000);

    //Nadopuni

    Console.ReadKey();
}

public class Grad
{
    private string naziv;
    private double x;
    private double y;
    private int brojStanovnika;

    public Grad(double X, double Y, string NAZIV, int BROJSTANOVNIKA)
    {
        naziv = NAZIV;
        x = X;
        y = Y;
        brojStanovnika = BROJSTANOVNIKA;
    }

    private void ispis1()
    {
        Console.WriteLine("Naziv grada: " + naziv + ", x= " + x + ", y= "
            + y + ", brSt= " + brojStanovnika);
    }

    public void ispis2()
    {
        Console.WriteLine("Naziv grada: " + naziv + ", x= " + x + ", y= "
            + y + ", brSt= " + brojStanovnika);
    }

    //Nadopuni

}
```



Ispit #17 (21.06.18. – grupa A)

1. Napišite program koji učitava četiri realne vrijednosti s tipkovnice, redom x_1 , y_1 , x_2 i y_2 koje predstavljaju dvije točke u koordinatnom sustavu $T_1 (x_1, y_1)$ i $T_2 (x_2, y_2)$, odnosno dužinu $\overline{T_1 T_2}$. Ako su sve unesene vrijednosti unutar intervala $[-10, 10]$ potrebno je izračunati i ispisati točku polovišta dužine u formatu ispisa kao u primjeru, inače je potrebno ispisati „Nisu unesene ispravne vrijednosti!“.

Provjeru da li je pojedina vrijednost unutar intervala $[-10, 10]$ potrebno je napisati u zasebnoj metodi. Provjera koja prima samo jedan argument te vraća rezultat provjere.

Polovište se određuje kao $P(x, y) = P(\frac{x_1+x_2}{2}, \frac{y_1+y_2}{2})$.

Primjerice, ako se redom učitaju vrijednosti -1.5, 3.0, 4.0, -2.0 potrebno je ispisati: „Polovište učitane dužine je: P (1.25, 0.5)“.

2. Napišite konzolnu aplikaciju koja učitava sadržaj datoteke bodovi.txt u kojoj su upisani realni brojevi koji predstavljaju bodove studenata na ispitu, svaki u zasebnom retku.

Program treba na kraju ispisati koliko je studenata pristupilo ispitu, prosječan, minimalan i maksimalan broj bodova.

Napomena: potrebno je koristiti try-catch blok, te nije potrebno dodati imenski prostor System.IO.

3. Napišite konzolnu aplikaciju koja generira 100 slučajnih cijelih brojeva u intervalu $[5, 99]$ te sprema brojeve ovisno o njihovoj parnosti u dvije liste: parnu i neparnu.

U svakoj iteraciji provjerava se parnost dohvaćenog slučajnog broja, te ako je broj paran dodaje ga se u listu parnih, a ako je broj neparan u listu nepranih brojeva.

4. Napišite program koji učitava dvije cjelobrojne vrijednosti n i m koje predstavljaju broj redaka i stupaca matrice $M = n \times m$. Potom je potrebno kreirati matricu M i popuniti ju slučajnim cijelim brojevima unutar intervala $[1, 10]$. Na kraju je sadržaj matrice potrebno ispisati u konzolu u proizvoljnom formatu.

5. Napišite kod klase Student koja sadrži javne attribute Ime, Prezime i JMBAG, te privatni atribut Ocjena. Konstruktor klase prima argumente imena, prezimena i jmbag, a ocjenu inicijalizira na vrijednost 1.

Potrebno je također napisati i javnu metodu postaviOcjenu koja kao argument prima ocjenu te postavlja studentu ocjenu predanu preko argumenta.



Ispit #18 (21.06.18. – grupa B)

1. Napišite program koji učitava četiri cijela broja s tipkovnice, redom x_1 , y_1 , x_2 i y_2 koji predstavljaju dvije točke u koordinatnom sustavu $T_1 (x_1, y_1)$ i $T_2 (x_2, y_2)$, odnosno dužinu $\overline{T_1 T_2}$. Ako su sve unesene vrijednosti unutar intervala $[-20, 20]$ potrebno je izračunati i ispisati udaljenost između dvije točke u formatu ispisa kao u primjeru, inače je potrebno ispisati „Nisu unesene ispravne vrijednosti!“.

Provjeru da li je pojedina vrijednost unutar intervala $[-20, 20]$ potrebno je napisati u zasebnoj metodi Provjera koja prima samo jedan argument te vraća rezultat provjere.

Udaljenost se određuje kao $d(T_1, T_2) = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$.

Primjerice, ako se redom učitaju vrijednosti -1, 3, 4, -2 potrebno je ispisati: „Dužina iznosi: 7.07“.

2. Napišite konzolnu aplikaciju koja učitava cijele brojeve iz konzole sve dok se ne unese vrijednost 0. Uneseni brojevi spremaju se u datoteku brojevi.txt, svaki u zasebnom retku.

Na kraju je u datoteku brojevi.txt potrebno upisati i prosjek učitanih brojeva (računajući i zadnje unesenu 0).

Napomena: potrebno je koristiti try-catch blok.

3. Napišite konzolnu aplikaciju koja 100 slučajnih cijelih brojeva unutar intervala $[1, 100]$ sprema u dvije liste ovisno o tome da li su brojevi djeljivi s 5 ili ne.

U svakoj iteraciji provjerava se da li je broj djeljiv s 5 ili ne, ako je dodaje ga se u listu djeljivih s 5, inače ga se dodaje u listu nedjeljivih s 5.

4. Napišite program koji učitava 9 cjelobrojnih vrijednosti u kvadratnu matricu (3×3) . Pritom je potrebno koristiti petlje po izboru. Nakon što su učitani svi elementi matrice potrebno je ispisati elemente glavne dijagonale matrice. Format ispisa mora biti kao u primjeru. Primjerice, ako su redom učitane vrijednosti po redcima:

1, 2, 3

4, 5, 6

7, 8, 9

Potrebno je ispisati: „Elementi glavne dijagonale matrice su: 1, 5, 9“

5. Napišite kod klase Student koja sadrži javne attribute Ime, Prezime i JMBAG, te privatni atribut Ocjena. Konstruktor klase prima argumente imena, prezimena i jmbag, a ocjenu inicijalizira na vrijednost -1.

Potrebno je također napisati i javnu metodu Ispis koja ispisuje ime, prezime i ocjenu studenta u formatu: „Student Ime Prezime iz kolegija Algoritmi i programiranje je ostvario ocjenu ocjena!“.



Ispit #19 (05.07.18. – grupa A)

1. Nadopunite sljedeći kod metodama UvecajKilometre i PromijeniVlasnika tako da njihovi parametri odgovaraju pozivu iz main metode.

Metoda UvecajKilometre treba uvećati kilometre za zadani broj koji se predaje preko argumenta, dok metoda PromijeniVlasnika treba promijeniti ime vlasnika u ime koje se predaje preko argumenta.

```
static void Main(string[] args)
{
    Auto a = new Auto("Insignia", 50000, "Marko");
    a.UvecajKilometre(10000);
    a.PromijeniVlasnika("Ivo");
}

public class Auto
{
    private string imeAuta;
    private int kilometri;
    private string imeVlasnika;

    public Auto(string IMEAUTA, int KILOMETRI, string IMEVLASNIKA)
    {
        imeAuta = IMEAUTA;
        kilometri = KILOMETRI;
        imeVlasnika = IMEVLASNIKA;
    }

    //Nadopuni
}
```



2. Napišite konzolnu aplikaciju koja učitava sadržaj datoteke paketi.txt u kojoj su upisani realni brojevi koji predstavljaju volumen paketa (m^3) koje je potrebno dostaviti vozilima koja imaju volumni kapacitet $10 m^3$.

Program u proizvoljnom formatu treba ispisati ukupni volumen koji je potrebno prevesti i broj vozila koji je potrebno koristiti.

Napomena: za zaokruživanje na prvi veći cijeli broj koristiti funkciju `Math.Ceiling(double value)` unutar ugrađene matematičke klase.

Potrebno je koristiti try-catch blok, te nije potrebno dodati imenski prostor `System.IO`.

3. Napišite konzolnu aplikaciju koja generira 1000 slučajnih cijelih brojeva koji predstavljaju brzine vozila unutar intervala $[20, 100]$ km/h na određenom prosjeku prometnice.

Ograničenje na prometnici je 80 km/h.

Potrebno je u zasebnu listu spremi brzine vozila čija je vrijednost iznad ograničenja i u proizvoljnom formatu ispisati koliko ima takvih zabilježenih brzina (većih od ograničenja).

4. Napišite konzolnu aplikaciju koja učitava JMBAG (niz znakova) studenata i broj bodova na ispitu (realan broj) u jednodimenzionalna polja (dva polja). Ispitni rok prijavilo je 75 studenata.

Nakon učitavanja potrebno je u proizvoljnom formatu ispisati JMBAG i broj bodova onih studenata koji imaju više od 12 bodova.

5. Napišite konzolnu aplikaciju koja učitava četiri realna broja x_1, y_1, x_2, y_2 koji predstavljaju geografske koordinate u stupnjevima.

Potrebno je napisati metodu udaljenost koja prima učitane geografske koordinate te vraća udaljenost u kilometrima između njih.

U glavnom programu potrebno je ispisati poruku u formatu kao u primjeru. Za izračun udaljenosti koristiti euklidsku udaljenost, te koristiti $1^\circ = 111.32$ km.

Primjer: ako se redom učitaju vrijednosti: 15.97, 45.8, 18.68, 45.55 potrebno je ispisati: „Udaljenost između točaka (15.97, 45.8) i (18.68, 45.55) iznosi 302.95 km!“.

Izračun euklidske udaljenosti između dvije točke $T_1 (x_1, y_1)$ i $T_2 (x_2, y_2)$ računa se prema izrazu:

$$d(T_1, T_2) = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$



Ispit #20 (05.07.18. – grupa B)

1. Nadopunite sljedeći kod unutar main metode na način da stvorite novi grad Zagreb s koordinatama $x = 15.9773$, $y = 45.8129$ i brojem stanovnika 800000.

Potom pozovite metodu koja će ispisati informacije vezane za stvoreni grad.

```
static void Main(string[] args)
{
    //Nadopuni

    Console.ReadKey();
}

public class Grad
{
    private string naziv;
    private double x;
    private double y;
    private int brojStanovnika;

    public Grad(double X, double Y, string NAZIV, int BROJSTANOVNIKA)
    {
        naziv = NAZIV;
        x = X;
        y = Y;
        brojStanovnika = BROJSTANOVNIKA;
    }

    private void ispis1()
    {
        Console.WriteLine("Naziv grada: " + naziv + ", x = ", +x + ", y = "
            + y + ", brSt=" + brojStanovnika);
    }

    public void ispis2()
    {
        Console.WriteLine("Naziv grada: " + naziv + ", x = ", +x + ", y = "
            + y + ", brSt=" + brojStanovnika);
    }
}
```



2. Napišite konzolnu aplikaciju koja učitava sadržaj datoteke vozila.txt u kojoj su upisani realni brojevi koji predstavljaju udaljenost koje su vozila prošla u određenom danu. Jedan redak odgovara prijeđenoj udaljenosti jednog vozila.

Potrebno je odrediti maksimalnu i prosječnu prijeđenu udaljenost tog dana. Rezultat je potrebno ispisati u proizvoljnom formatu.

Potrebno je koristiti try-catch blok, te nije potrebno dodati imenski prostor System.IO.

3. Nadopunite kod na način da se iz zadane liste vozniPark koji predstavljaju tipove vozila u voznom parku, slučajno odabere 5 tipova vozila za današnju dostavu te ih se doda u listu tipovi.

Vrijednosti se mogu ponavljati.

```
static void Main(string[] args)
{
    List<int> vozniPark = new List<int>(
        new int[] { 1, 2, 5, 6, 7, 9, 15, 100 });

    //Nadopuni

    Console.ReadKey();
}
```

4. Napišite konzolnu aplikaciju koja učitava matricu udaljenosti između n gradova.

Potrebno je prvo preko tipkovnice unijeti vrijednost n, potom kreirati matricu $M = n \times n$, te ju popuniti realnim brojevima koji predstavljaju udaljenosti između gradova.

Na kraju je potrebno ispisati sadržaj matrice u proizvoljnom formatu.

5. Napišite konzolnu aplikaciju koja učitava tri realna broja s, t i v_{ogr} koji označavaju redom duljinu dionice prometnice [m], vrijeme putovanja vozila po dionici [s] i ograničenje brzine na dionici prometnice [km/h].

Potrebno je napisati metodu jednolikaBrzina koja kao argumente prima duljinu dionice i vrijeme putovanja vozila po dionici, a vraća izračunatu jednoliku brzinu kretanja u km/h (pretvorba: $1 \text{ m/s} = 3.6 \text{ km/h}$).

U glavnom dijelu programa potrebo je provjeriti da li je izračunata brzina veća od učitane brzine ograničenja, te ako je ispisati poruku u formatu kao u primjeru.

Primjer: ako se redom učitaju vrijednosti 1000, 40, i 80 potrebno je ispisati: „Vozilo se kreće brzinom 90 km/h, odnosno iznad ograničenja brzine!“.



Ispit #21 (12.07.18. – grupa A)

1. Nadopunite sljedeći kod unutar main metode na način da stvorite novi grad Zagreb s koordinatama $x = 15.9773$, $y = 45.8129$ i brojem stanovnika 800000.

Potom u konzolu ispišite x koordinatu grada (vrijednost atributa unutar instance).

```
static void Main(string[] args)
{
    //Nadopuni

    Console.ReadKey();
}

public class Grad
{
    private string naziv;
    public Tocka t;
    private int brojStanovnika;

    public Grad(Tocka T, string NAZIV, int BROJSTANOVNIKA)
    {
        t = T;
        naziv = NAZIV;
        brojStanovnika = BROJSTANOVNIKA;
    }
}

public class Tocka
{
    public double x;
    public double y;

    public Tocka(double X, double Y)
    {
        x = X;
        y = Y;
    }
}
```



2. Napišite konzolnu aplikaciju koja učitava mase paketa (kg, cijeli broj) koje je potrebno dostaviti sve dok se ne unese masa paketa manja ili jednaka nuli (zadnji paket se ne pribrojava).

Paketi se spremaju u vozila masenog kapaciteta 500 kg.

Program u proizvoljnom formatu treba ispisati ukupnu masu koju je potrebno prevesti i broj vozila koji je potrebno koristiti.

Napomena: za zaokruživanje na prvi veći cijeli broj koristiti funkciju `Math.Ceiling(double value)`.

3. Napišite konzolnu aplikaciju koja generira 1000 realnih brojeva unutar intervala $[0, 100>$ koji predstavljaju simulirane vrijednosti brzine te ih zapisuje u datoteku `brzine.txt`.

Nije potrebno dodati imenski prostor `System.IO`.

4. Napišite konzolnu aplikaciju koja učitava sadržaj datoteke `brojVozila.txt` u jednodimenzionalno polje.

U datoteci je u zasebnom retku za svaki dan u mjesecu (31 redak) zapisan broj zabilježenih vozila (realan broj) u tome danu na određenom segmentu prometnice.

Potrebno je izračunati prosječan broj zabilježenih vozila te ga ispisati u konzolu u proizvoljnom formatu.

Potrebno je koristiti `try-catch` blok, te nije potrebno dodati imenski prostor `System.IO`.

5. Napišite metodu `Najveci` koja prima tri realna broj `a`, `b` i `c` te u konzolu ispisuje najveći od tri predana broja.

Unutar glavnog dijela programa potrebno je iz konzole učitati brojeve te pozvati metodu.

Format ispisa bora biti kao u primjeru.

Primjer: ako se redom učitaju vrijednosti 10, -11.25, 11.05 potrebno je ispisati: „Najveći od tri učitana broja 10, -11.25, 11.05 je 11.05.“.



Ispit #22 (12.07.18. – grupa B)

1. Nadopunite sljedeći kod unutar Main metode na način da stvorite novi auto naziva Insignia, kojem je vlasnik Marko, te koji se trenutno nalazi na GPS lokaciji $x = 15.9773$, $y = 45.8129$.

Potom u konzolu ispišite vrijednost y GPS lokacije vozila (vrijednost atributa unutar instance).

```
static void Main(string[] args)
{
    //Nadopuni

    Console.ReadKey();
}

public class GPSLokacija
{
    public double x;
    public double y;

    public GPSLokacija(double X, double Y)
    {
        x = X;
        y = Y;
    }
}

public class Auto
{
    private string imeAuta;
    private string imeVlasnika;
    public GPSLokacija lokacija;

    public Auto(string IMEAUTA, string IMEVLASNIKA, GPSLokacija LOKACIJA)
    {
        imeAuta = IMEAUTA;
        lokacija = LOKACIJA;
        imeVlasnika = IMEVLASNIKA;
    }
}
```



2. Napišite konzolnu aplikaciju koja predstavlja punjenje vozila s paketima.

Mase paketa (realni brojevi) se učitavaju iz konzole sve dok se ne prekorači nosivost vozila koja iznosi 400 kg.

Kada se prekorači nosivost potrebno je odbaciti učitani paket i ispisati poruku „Odbacujem paket zbog prekoračenja mase! “.

Na kraju je potrebno ispisati ukupnu masu koja se prevozi vozilom u proizvoljnom formatu.

3. Napišite konzolnu aplikaciju koja generira 900 realnih brojeva unutar intervala [0, 10> koji predstavljaju simulirane vrijednosti brzine te ih zapisuje u datoteku simuliraneBrzine.txt.

Nije potrebno dodati imenski prostor System.IO.

4. Napišite konzolnu Aplikaciju koja učitava sadržaj datoteke vozilaUdaljenosti.txt u jednodimenzionalno polje.

U datoteci su u zasebnom retku za svaki dan u godini (365 redaka) zapisane udaljenosti (cijeli broj) koje je vozilo toga dana ostvarilo.

Potrebno je izračunati prosječnu prijeđenu udaljenost vozila kroz godinu te ju ispisati u konzolu u proizvoljnom formatu.

Potrebno je koristiti try-catch blok, te nije potrebno dodati imenski prostor System.IO.

5. Napišite metodu Najmanji koja prima tri cijela broja a, b i c te u konzolu ispisuje najmanji od tri predana broja.

Unutar glavnog dijela programa potrebno je iz konzole učitati brojeve te pozvati metodu.

Format ispisa mora biti kao u primjeru.

Primjer: ako se redom učitaju vrijednosti 10, -11, 11 potrebno je ispisati: „Najmanji od tri učitana broja 10, -11 i 11 je -11.“.



Ispit #23 (06.09.18. - grupa A)

1. Nadopunite sljedeći kod unutar Main metode na način da stvorite novi grad Zagreb s koordinatama $x = 15.9773$, $y = 45.8129$ i brojem stanovnika 800000.

Potom pozovite metodu koja će ispisati informacije vezane na stvoreni grad.

```
static void Main(string[] args)
{
    //Nadopuni

    Console.ReadKey();
}

public class Grad
{
    private string naziv;
    private double x;
    private double y;
    private int brojStanovnika;

    public Grad(double Y, double X, string NAZIV, int BROJSTANOVNIKA)
    {
        naziv = NAZIV;
        x = X;
        y = Y;
        brojStanovnika = BROJSTANOVNIKA;
    }

    private void ispis1()
    {
        Console.WriteLine("Naziv grada: " + naziv + ", x = " + x + ", y = "
            + y + ", brSt = " + brojStanovnika);
    }

    public void ispis2()
    {
        Console.WriteLine("Naziv grada: " + naziv + ", x = " + x + ", y = "
            + y + ", brSt = " + brojStanovnika);
    }
}
```



2. Napišite konzolnu aplikaciju koja iz datoteke poslovnice.txt učitava mjesečni ostvareni prihod određene logističke tvrtke.

Za svaku poslovnicu u zasebnom retku navedena je ukupna mjesečna zarada kao realan broj u kunama.

Potrebno je u drugu datoteku poboljsanja.txt zapisati indekse redaka onih poslovnica čija je mjesečna zarada bila manja od 50000 kn.

Smatra se da prvi redak, odnosno prva poslovnica u datoteci poslovnice.txt ima indeks 0, druga 1 itd.

3. Napišite konzolnu aplikaciju koja generira 1000 slučajnih cijelih brojeva unutar intervala [80, 200] koji predstavljaju brzine vozila na prometnici [km/h].

U svakoj iteraciji provjerava se da li je generirana brzina veća od 140 km/h, te ako je dodaje ju se u listu prekoracenjaBrzina.

Na kraju je potrebno ispisati koliko je bilo brzina većih od 140 km/h.

4. Napišite program koji učitava 6 cjelobrojnih vrijednosti iz konzole u pravokutnu matricu (2 × 3). Pritom je potrebno koristiti petlje po izboru.

Nakon što su učitani svi elementi matrice potrebno je ispisati one elemente čija je vrijednost veća od 3. Primjerice, ako se učitava matrica:

1, 2, 4

3, 5, 6

potrebno je ispisati:

Elementi matrice veći od 3 su: 4 5 6

5. Napišite konzolnu aplikaciju koja učitava dva realna broja m i n, koji označavaju redom masu jednog paketa [kg] i broj paketa koji je potrebno ukrcati u kamion.

Potrebno je izračunati ukupnu masu koju je potrebno ukrcati u kamion te pozvati metodu ProvjeraMase.

Metoda kao argument prima iznos mase koji je potrebno ukrcati, te provjerava i ispisuje poruku da li je premašena dopuštena nosiva masa vozila u iznosu od 1000 kg. Metoda ne vraća rezultat, a format ispisa treba biti kao u primjerima.

Primjer: ako se redom učitaju vrijednosti 100 i 15 potrebno je ispisati: „Ukupna masa iznosi 1500 kg, te premašuje dopuštenu nosivost vozila.“.

Ako se redom učitaju vrijednosti 10 i 10 potrebno je ispisati: „Ukupna masa iznosi 100 kg, te ne premašuje dopuštenu nosivost vozila.“.



Ispit #24 (06.09.18. - grupa B)

1. Nadopunite sljedeći kod s metodama UvecajKilometre i PromijeniVlasnika tako da njihovi parametri odgovaraju pozivu iz Main metode.

Metoda UvecajKilometre treba uvećati kilometre za zadani broj koji se predaje preko argumenta, dok metoda PromijeniVlasnika treba promijeniti ime vlasnika u ime koje se predaje preko argumenta.

```
static void Main(string[] args)
{
    Auto a = new Auto("Insignia", 50000, "Marko");
    a.UvecajKilometre(10000);
    a.PromijeniVlasnika("Ivo");

    Console.ReadKey();
}

public class Auto
{
    private string imeAuta;
    private int kilometri;
    private string imeVlasnika;

    public Auto(string IMEAUTA, int KILOMETRI, string IMEVLASNIKA)
    {
        imeAuta = IMEAUTA;
        kilometri = KILOMETRI;
        imeVlasnika = IMEVLASNIKA;
    }

    //Nadopuni
}
```



2. Napišite konzolnu aplikaciju koja iz datoteke udaljenosti.txt učitava godišnju prijeđenu udaljenost vozila u voznom parku. Za svako vozilo u zasebnom retku navedena je ukupna godišnja prijeđena udaljenost kao realan broj u kilometrima.

Potrebno je u drugu datoteku servisVozila.txt zapisati indekse redaka onih vozila čija je godišnja prijeđena udaljenost veća od 50000 km. Smatra se da prvi redak, odnosno prvo vozilo u datoteci udaljenosti.txt ima indeks 0, drugi 1 itd.

3. Napišite konzolnu aplikaciju koja generira 1000 slučajnih cijelih brojeva unutar intervala $[0, 5>$ koji predstavljaju simulirane vrijednosti veličina paketa $[m^3]$ na traci koje je potrebno spremiti u vozilo.

U svakoj iteraciji provjerava se da li je generirana veličina paketa manja od $3 m^3$, te ako je dodaje ju se u listu paketi.

Na kraju je potrebno ispisati koliko je bilo paketa čija je veličina manja od $3 m^3$.

4. Napišite program koji učitava 6 cjelobrojnih vrijednosti iz konzole u pravokutnu matricu (3×2) . Pritom je potrebno koristiti petlje po izboru.

Nakon što su učitani svi elementi matrice potrebno je ispisati one elemente čija je vrijednost manja od -1. Primjerice ako se učitava matrica:

-1, -2

-3, 3

4, -5

potrebno je ispisati:

Elementi matrice manji od -1 su: -2 -3 -5

5. Napišite konzolnu aplikaciju koja učitava dva realna broja v i s , koji označavaju redom prosječnu brzinu kretanja vozila $[km/h]$ i duljinu prometnice $[km]$.

Potrebno je napisati metodu VrijemePutovanja koja kao argumente prima prosječnu brzinu kretanja vozila i duljinu prometnice, a kao rezultat vraća vrijeme u satima potrebno za prolazak prometnice.

U glavnom dijelu programa potrebno je provjeriti da li je vrijeme putovanja veće od 2 h ili ne, te ispisati poruku kao u primjerima.

Primjer: Ako se redom učitaju vrijednosti 100 i 280 potrebno je ispisati: „Vrijeme putovanja iznosi 2.8 h, te vozilo neće na vrijeme stići na odredište.“.

Ako se redom učitaju vrijednosti 100 i 150 potrebno je ispisati: „Vrijeme putovanja iznosi 1.5 h, te će vozilo na vrijeme stići na odredište.“.



Ispit #25 (17.11.18. - grupa A)

1. Napišite kod klase Student koja sadrži javne attribute Ime, Prezime i JMBAG, te privatni atribut Ocjena.

Konstruktor klase prima argumente imena, prezimena i jmbag, a ocjenu inicijalizira na vrijednost 1.

Potrebno je također napisati i javnu metodu PostaviOcjenu koja kao argument prima ocjenu te postavlja studentu zadanu ocjenu.

2. Napišite program koji učitava šest realnih vrijednosti s tipkovnice, redom x1, y1, t1, x2, y2 i t2 koje predstavljaju geografske koordinate (x, y) i vremenski trenutak zapisa signala t (u satima [h]).

Pomoću geografskih koordinata T1 (x1, y1) i T2 (x2, y2) potrebno je izračunati udaljenost između točaka. Udaljenost se računa kao Euklidska udaljenost $d(T_1, T_2) = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$ pomnožena s vrijednošću 111.32 kako bi se dobio iznos u kilometrima.

Potrebno je izračunati i ispisati prosječnu brzinu kretanja vozila te ispisati da li se vozilo kreće iznad ili ispod ograničenja koje iznosi 100 km/h. Format ispisa mora biti kao u primjeru.

Primjerice ako se redom učitaju vrijednosti 15.7, 45.3, 4, 16.6, 44.2, 5.5 potrebno je ispisati: „Prosječna brzina kretanja vozila iznosi: 105.47 km/h! Vozilo se kreće iznad ograničenja!“.

3. Napišite konzolnu aplikaciju koja generira 1000 slučajnih cijelih brojeva u intervalu [5, 20] koji predstavljaju simulirane vrijednosti očitanih duljina vozila s radara.

Potrebno je prebrojati koliko je vrijednosti duljina vozila unutar intervala [5, 10>, a koliko unutar intervala [10, 20].

Program treba na kraju u proizvoljnom zapisu zapisati navedene dvije vrijednosti u datoteku vozila.txt.

4. Napišite konzolnu aplikaciju koja učitava sadržaj datoteke dostava_17_11_2018.txt u kojoj su upisani realni brojevi koji predstavljaju udaljenosti (u kilometrima) koje su dostavna vozila u tvrtki prošla u današnjem danu.

Potrebno je izračunati koliko je ukupno utrošeno kuna na dostavu ako je potrošnja 0.6 kn/km. Format ispisa je proizvoljan.

Potrebno je koristiti try-catch blok.

5. Napišite konzolnu aplikaciju koja učitava matricu udaljenosti između n gradova.

Potrebno je prvo preko tipkovnice unijeti vrijednost n, potom kreirati matricu $M = n \times n$, te ju popuniti realnim brojevima koji predstavljaju udaljenosti između gradova.

Na kraju je potrebno ispisati sadržaj matrice u proizvoljnom formatu.



Ispit #26 (07.02.19. - grupa A)

1. Napišite kod klase Kolegij koji sadrži privatne attribute Naziv, ECTS, Nositelj i Broj_studenata.

Konstruktor klase prima argumente naziva, ects-a i nositelja, a broj studenata inicijalizira na vrijednost 0.

Potrebno je napisati i javnu metodu Ispis koja ispisuje naziv, ects-e, nositelja i broj studenata u formatu: „Kolegij Naziv, ima ECTS ects-a, nositelj kolegija je Nositelj, a kolegij je upisalo Broj_studenata studenata!“. Umjesto vrijednosti u kurzivu potrebno je upisati odgovarajuće vrijednosti.

2. Napišite program koji učitava šest realnih vrijednosti s tipkovnice, redom x1, y1, t1, x2, y2 i t2 koje predstavljaju geografske koordinate (x, y) i vremenski trenutak zapisa signala t (u satima [h]).

Pomoću geografskih koordinata T1 (x1, y1) i T2 (x2, y2) potrebno je izračunati udaljenost između točaka. Udaljenost se računa kao Euklidska udaljenost $d(T_1, T_2) = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$ pomnožena s vrijednošću 111.32 kako bi se dobio iznos u kilometrima.

Potrebno je izračunati i ispisati prosječnu brzinu kretanja vozila te ispisati da li se vozilo kreće iznad ili ispod ograničenja koje iznosi 100 km/h. Format ispisa mora biti kao u primjeru.

Primjerice ako se redom učitaju vrijednosti 15.7, 45.3, 4, 16.6, 44.2, 5.5 potrebno je ispisati: „Prosječna brzina kretanja vozila iznosi: 105.47 km/h! Vozilo se kreće iznad ograničenja!“.

3. Napišite konzolnu aplikaciju koja generira 1000 slučajnih cijelih brojeva u intervalu [5, 20] koji predstavljaju simulirane vrijednosti očitanih duljina vozila s radara.

Potrebno je prebrojati koliko je vrijednosti duljina vozila unutar intervala [5, 10>, a koliko unutar intervala [10, 20].

Program treba na kraju u proizvoljnom zapisu zapisati navedene dvije vrijednosti u datoteku vozila.txt.

4. Napišite konzolnu aplikaciju koja učitava sadržaj datoteke brzineNaPrometnici.txt u kojoj su upisani cijeli brojevi svaki u zasebnom retku koji predstavljaju izmjerene brzine vozila na prometnici.

Nakon što se učitava vrijednost brzine -1 potrebno je završiti učitavanje podataka i u konzolu ispisati broj zabilježenih brzina i prosječnu brzinu u proizvoljnom formatu.

Napomena: potrebno je koristiti try-catch blok, te nije potrebno dodati imenski prostor System.IO.

5. Napišite konzolnu aplikaciju koja učitava matricu udaljenosti između n gradova.

Potrebno je prvo preko tipkovnice unijeti vrijednost n, potom kreirati matricu $M = n \times n$, te ju popuniti realnim brojevima koji predstavljaju udaljenosti između gradova.

Na kraju je potrebno ispisati sadržaj matrice u proizvoljnom formatu.



Ispit #27 (21.02.19. - grupa A)

1. Napišite kod klase Student koja sadrži javne attribute Ime, Prezime i JMBAG, te privatni atribut Ocjena.

Konstruktor klase prima argumente imena, prezimena i jmbag, a ocjenu inicijalizira na vrijednost 1.

Potrebno je također napisati i javnu metodu PostaviOcjenu koja kao argument prima ocjenu te postavlja studentu zadanu ocjenu.

2. Napišite konzolnu aplikaciju koja učitava dva realna broja D i a, koji predstavljaju diskriminantu i vrijednost prvog koeficijenta kvadratne funkcije.

Ako su vrijednosti D i a manje od nule potrebno je ispisati „Za D i a funkcija ne siječe x-os te su joj sve vrijednosti negativne!“.

Ako je D manje od nule, a a veće od nule potrebno je ispisati „Za D i a funkcija ne siječe x-os te su joj sve vrijednosti pozitivne!“.

Inače je potrebno ispisati „Funkcija siječe x-os!“.

Primjer: ako se za D učitava vrijednost -1.25 , a za a vrijednost 0.5 , potrebno je ispisati: „Za $D = -1.25$ i $a = 0.5$ funkcija ne siječe x-os te su joj sve vrijednosti pozitivne!“.

3. Napišite konzolnu aplikaciju koja generira 1000 slučajnih cijelih brojeva u intervalu $[5, 20]$ koji predstavljaju simulirane vrijednosti očitanih duljina vozila s radara.

Potrebno je prebrojati koliko je vrijednosti duljina vozila unutar intervala $[5, 10>$, te izračunati prosječnu duljinu vozila čija je duljina u intervalu $[5, 10>$.

Program treba na kraju u proizvoljnom zapisu zapisati navedene dvije vrijednosti u datoteku vozila.txt.

4. Napišite konzolnu aplikaciju koja učitava sadržaj datoteke brzineNaPrometnici.txt u kojoj su upisani cijeli brojevi svaki u zasebnom retku koji predstavljaju izmjerene brzine vozila na prometnici.

Nakon što se učitava vrijednost brzine -1 potrebno je završiti učitavanje podataka i u konzolu ispisati broj zabilježenih brzina i prosječnu brzinu u proizvoljnom formatu.

Napomena: potrebno je koristiti try-catch blok, te nije potrebno dodati imenski prostor System.IO.

5. Napišite konzolnu aplikaciju koja učitava matricu udaljenosti između n gradova.

Potrebno je prvo preko tipkovnice unijeti vrijednost n, potom kreirati matricu $M = n \times n$, te ju popuniti realnim brojevima koji predstavljaju udaljenosti između gradova.

Na kraju je potrebno ispisati sadržaj matrice u proizvoljnom formatu.



Ispit #28 (13.04.19. - grupa A)

1. Nadopunite sljedeći kod unutar Main metode na način da stvorite novi grad Zagreb s koordinatama $x = 15.9773$, $y = 45.8129$ i brojem stanovnika 800000.

```
static void Main(string[] args)
{
    //Nadopuni

}

public class Grad
{
    private string naziv;
    public Tocka t;
    private int brojStanovnika;

    public Grad(int BROJSTANOVNIKA, string NAZIV, Tocka T)
    {
        t = T;
        naziv = NAZIV;
        brojStanovnika = BROJSTANOVNIKA;
    }
}

public class Tocka
{
    public double x;
    public double y;

    public Tocka(double Y, double X)
    {
        x = X;
        y = Y;
    }
}
```



2. Napišite konzolnu aplikaciju koja učitava cijeli broj koji predstavlja izbor u izborniku: 1 - površina pravokutnika, 2 - opseg kružnice.

Ukoliko se unese 1 potrebno je dodatno zatražiti korisnika unos stranica pravokutnika a i b (realni brojevi) te izračunati i ispisati površinu pravokutnika samo ako su i a i b veći od 0. Ukoliko a i b nisu veći od nula potrebno je ispisati „Neispravan unos!“.

Ukoliko se unese 2 potrebno je korisnika zatražiti unos polumjera kružnice r (realan broj) te izračunati i ispisati opseg kružnice ($O = 2\pi r$).

Ukoliko se ne unese 1 ili 2 za izbor potrebno je ispisati „Neispravan unos!“.

Format ispisa mora biti kao u primjeru, tako da se ispišu učitani parametri i izračunata vrijednost.

Primjer: ako se učitava prvo vrijednost 1, a potom vrijednosti 5.5 i 6.7 potrebno je ispisati: „Površina pravokutnika određenog stranicama a=5.5 i b=6.7 iznosi P=36.85!“.

3. Napišite metodu Prosjek koja kao argument prima listu cijelih brojeva. Metoda računa prosječnu vrijednost elemenata u listi i vraća rezultat.

4. Napišite konzolnu aplikaciju koja učitava sadržaj datoteke brzine.txt u kojoj su upisani cijeli brojevi svaki u zasebnom retku koji predstavljaju izmjerene brzine vozila na prometnici.

Potrebno je stvoriti dvije liste, jednu za brzine manje od ograničenja brzine koje iznosi 60 km/h i jednu za brzine veće ili jednake 60 km/h. Nakon što se učitava vrijednost brzine potrebno je učitanu brzinu dodati u odgovarajuću listu.

Na kraju je potrebno pozvati metodu prosjek iz zadatka 3 za svaku od lista kako bi se ispisala prosječna brzina lista u proizvoljnom formatu.

Napomena: potrebno je koristiti try-catch blok, te nije potrebno dodati imenski prostor System.IO.

5. Napišite konzolnu aplikaciju koja učitava sadržaj za n gradova pri čemu se n unosi preko tipkovnice.

Za svaki grad unosi se naziv grada (niz znakova) i udaljenost u kilometrima od trenutne lokacije (realan broj).

Prema tome potrebno je kreirati dva jednodimenzionalna polja te ih popuniti odgovarajućim sadržajem.

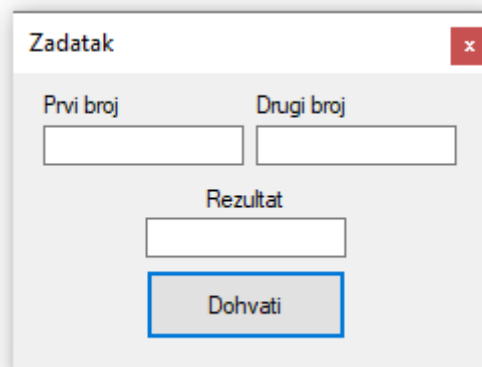
Potrebno je ispisati naziv i pripadajuću udaljenost za 4 slučajno dohvaćena grada iz polja (mogu biti i isti gradovi). Format ispisa je proizvoljan.



Ispit #29 (13.06.19. - grupa A)

1. Nadopunite sljedeći kod unutar metode BtnDohvati_Click na način da izračunate cjelobrojni ostatak dijeljenja prvog i drugog cijelog broja unesenih u TextBoxove txtPrviBroj i txtDrugiBroj.

Rezultat je također cijeli broj koji je potrebno zapisati u TextBox txtRezultat. Sučelje je prikazano slikom.

The image shows a Windows application window with the title 'Zadatak'. Inside the window, there are two text boxes at the top, one labeled 'Prvi broj' and one labeled 'Drugi broj'. Below these, there is a third text box labeled 'Rezultat'. At the bottom of the window, there is a button labeled 'Dohvati'. The window has a standard Windows title bar with a close button (X) in the top right corner.

```
private void BtnDohvati_Click(object sender, EventArgs e)
{
    //Nadopuni

}
```

2. Potrebno je napisati metodu IspisiDio koja kao argument prima tri vrijednosti: niz znakova s, početni indeks b (cijeli broj) i krajnji indeks e (cijeli broj).

Metoda u konzolu treba ispisati znak po znak predanog znakovnog niza počevši od znaka na indeksu b do znaka na indeksu e (oba znaka su uključujuća).

Ispis se treba vršiti na način da se znakovi ispisuju u istom retku (bez prelaska pokazivača u novi red).



3. Napišite konzolnu aplikaciju koja preko konzole učitava vrijeme putovanja t u satima (realan broj) i ID tipa vozila (cijeli broj).

ID tipa vozila dan je stupcem ID u tablici.

ID	Opis	Brzina [°/h]
1	Motor	$v_x = v_y = 0.7622$
2	Automobil	$v_x = v_y = 0.5717$
3	Kamion	$v_x = v_y = 0.5082$

Početna pozicija vozila u geografskim stupnjevima je $T(x_p, y_p) = (15.97790, 45.810857)$.

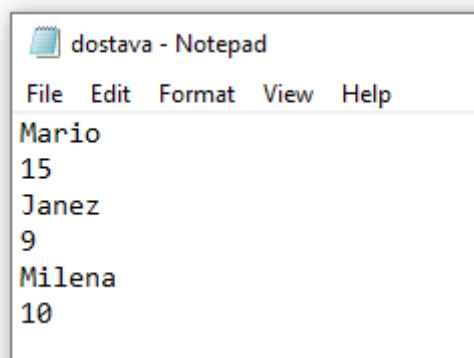
Ovisno o tome koji je ID vozila unesen potrebno je ispisati poziciju na kojoj će se vozilo naći nakon što se vozi t vremena, pri čemu su brzine zadane u tablici, a konačna pozicija $T(x_k, y_k)$ se računa koristeći izraze:

$$x_k = v_x * t + x_p$$

$$y_k = v_y * t + y_p$$

Format ispisa mora biti kao u primjeru. Primjerice, ukoliko se učitava vrijeme od 2 sata i vozilo pod ID-em 1, potrebno je ispisati: „Motor će se nakon 2 sata vožnje nalaziti na poziciji T (17.502305, 47.335257)!“.

4. Napišite konzolnu aplikaciju koja učitava sadržaj datoteke dostava.txt u kojoj su upisani podaci dostave na način prikazan slikom, odnosno ime dostavljača u jednom retku, pa u sljedećem retku količina paketa (cijeli broj) koju je taj dostavljač dostavio.



Potrebno je u konzolu ispisati količinu i ime dostavljača koji je taj dan dostavio najveći broj paketa. Format ispisa je proizvoljan.

Napomena: potrebno je koristiti try-catch blok, te nije potrebno dodati imenski prostor System.IO.

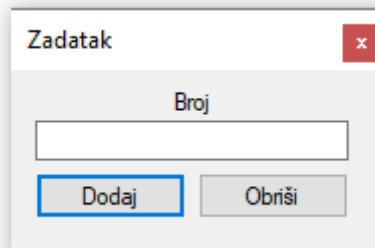
5. Napišite konzolnu aplikaciju za simulaciju bacanja kocke. Bacanje jedne kocke može rezultirati s jednom od šest slučajnih vrijednosti (cijeli brojevi) 1, 2, 3, 4, 5 ili 6.

Potrebno je simulirati tisuću (1000) bacanja jedne kocke te na kraju u proizvoljnom formatu u datoteku vjerojatnosti.txt ispisati vjerojatnosti pojave svake od vrijednosti 1, 2, 3, 4, 5 i 6.



Ispit #30 (13.06.19. - grupa B)

1. Nadopunite sljedeći kod unutar odgovarajućih metoda na način da klikom na gumb Dodaj dodate u listu vrijednost broja upisanu u TextBox txtBroj, a klikom na gumb Obriši obrišete iz liste vrijednost broja upisanu u TextBox txtBroj. Sučelje je prikazano slikom.



```
//Nadopuni
```

```
private void BtnDodaj_Click(object sender, EventArgs e)
{
}
```

```
private void BtnObriši_Click(object sender, EventArgs e)
{
}
```

2. Napišite metodu prebrojiA koja preko argumenata prima niz znakova s. Metoda treba prebrojati koliko znakova u nizu znakova s je jednako znaku (slovu) a te vratiti navedeni rezultat (cijeli broj).

3. Napišite konzolnu aplikaciju koja preko konzole učitava vrijeme putovanja t u satima (realan broj) i ID tipa vozila (cijeli broj). ID tipa vozila dan je stupcem ID u tablici.

ID	Opis	Akceleracija [km/h ²]
3	Motor	72000
2	Automobil	36000
1	Kamion	14400

Početna brzina svakog vozila je $v_p = 10$ km/h. Ovisno o tome koji je ID vozila unesen potrebno je izračunati koliku će brzinu imati vozilo nakon što se vozi t vremena, pri čemu su akceleracije zadane u tablici 1, a konačna brzina v_k se računa koristeći izraz:

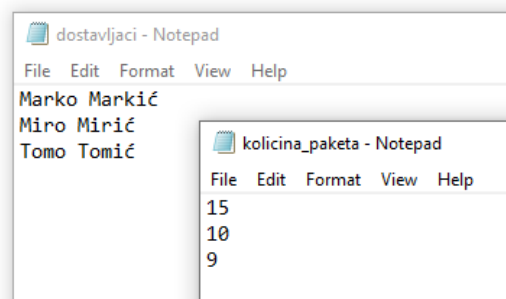
$$v_k = a * t + v_p$$

Također ukoliko je konačna brzina veća od 130 km/h potrebno je ispisati „Vozilo se vozi prebrzo!“. Format ispisa mora biti kao u primjeru.

Primjerice, ukoliko se učitava vrijeme od 0.0022 sata i vozilo pod ID-em 3, potrebno je ispisati „Motor će nakon 0.0022 sata vožnje imati brzinu 168.4 km/h! Vozilo se vozi prebrzo!“.



4. Napišite konzolnu aplikaciju koja učitava sadržaj dviju datoteka dostavljac.txt i kolicina_paketa.txt. U datoteci dostavljac.txt upisana su imena i prezimena dostavljača, a u istom retku u datoteci kolicina_paketa.txt upisana je količina paketa koju je odgovarajući dostavljač dostavio.



Potrebno je u konzolu ispisati ime i prezime dostavljača te količinu paketa koju je dostavljač dostavio. Format ispisa mora biti kao u primjeru.

Napomena: potrebno je koristiti try-catch blok, te nije potrebno dodati imenski prostor System.IO. Za primjer sa slike potrebno je ispisati:

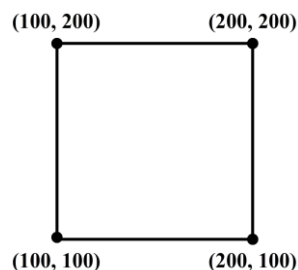
„Marko Marić je dostavio 15 paketa!

Miro Mirić je dostavio 10 paketa!

Tomo Tomić je dostavio 9 paketa!“

5. Napišite konzolnu aplikaciju za simulaciju odredišnih parova.

Početna točka je ishodište $T_p(x_p, y_p) = (0, 0)$. Potrebno je dohvatiti 100 slučajnih točaka $T_k(x_k, y_k)$ (odredišta) unutar kvadrata prikazanog slikom.



Smatra se da su koordinate x i y cijeli brojevi, te da su granice kvadrata uključujuće.

Euklidska udaljenost između početne točke i pojedinog odredišta računa se izrazom.

$$D(T_p, T_k) = \sqrt{(x_k - x_p)^2 + (y_k - y_p)^2}$$

U svakom koraku u datoteku udaljenosti.txt potrebno je ispisivati u proizvoljnom formatu udaljenost između polazišta i trenutno dohvaćenog odredišta.

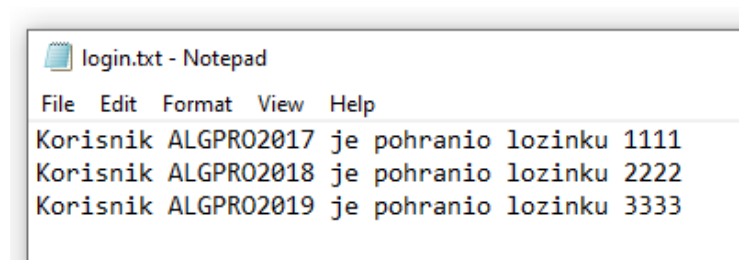


Ispit #31 (02.07.19. - grupa A)

1. Potrebno je napisati metodu Generiraj koja kao argument prima dvije vrijednosti: cijeli broj x i bool vrijednost flag.

Metoda vraća listu cijelih brojeva. Ako je primljena vrijednost varijable flag istina (true) metoda vraća listu veličine x popunjenu brojevima 1, a ako je vrijednost laž (false) metoda vraća listu veličine x popunjenu brojevima 0.

2. Napišite konzolnu aplikaciju koja omogućuje korisniku zapisivanje 3 proizvoljna korisnička imena i lozinke u datoteku. Format zapisa korisničkog imena i lozinke u datoteci prikazan je primjerom na slici.



Napomena: potrebno je koristiti try-catch blok, te nije potrebno dodati imenski prostor System.IO.

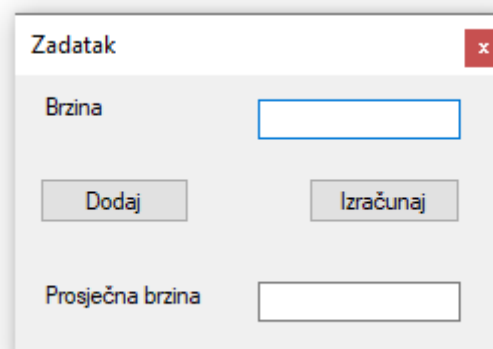
3. Nadopunite sljedeći kod unutar odgovarajućih metoda na način da se klikom na gumb Dodaj upisuje u listu vrijednost brzine automobila upisana u TextBox txtBrzina.

```
//Nadopuni  
  
private void BtnDodaj_Click(object sender, EventArgs e)  
{  
  
}  
  
private void BtnIzracunaj_Click(object sender, EventArgs e)  
{  
  
}
```

Klikom na gumb Izračunaj izračunava se i ispisuje prosječna vrijednost svih brzina u listi u TextBox txtProsječnaBrzina.

Prosječna vrijednost mora biti zaokružena na dva decimalna mjesta.

Sučelje je prikazano slikom.



4. Napišite program koji učitava ocjene za pet studenata te računa prosjek ocjena i ispisuje odgovarajuću poruku.

Nije potrebno vršiti provjeru da li je učitana dobra ocjena, očekuje se od korisnika da unese ispravnu ocjenu.

Ako je prosjek ocjena u intervalu $[0 - 1.5>$ potrebno je ispisati "Nedovoljan (1)!".

Ako je prosjek ocjena u intervalu $[1.5 - 2.5>$ potrebno je ispisati "Dovoljan (2)!".

Ako je prosjek ocjena u intervalu $[2.5 - 3.5>$ potrebno je ispisati "Dobar (3)!".

Ako je prosjek ocjena u intervalu $[3.5 - 4.5>$ potrebno je ispisati "Vrlo dobar (4)!".

Inače je potrebno ispisati "Odličan (5)!".

5. Napišite konzolnu aplikaciju koja učitava prirodni broj n sa konzole te potom kreira matricu $M = n \times n$.

Matricu je potrebno popuniti slučajnim prirodnim brojevima u intervalu $[1, 30]$.

Zatim je potrebno u jednodimenzionalno polje veličine n zapisati prosječne vrijednosti elemenata svakog retka matrice.



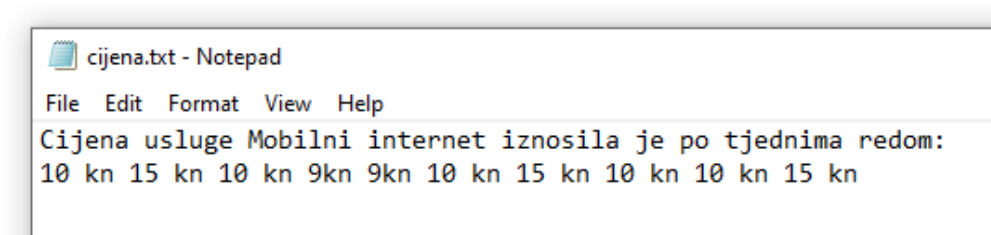
Ispit #32 (02.07.19. - grupa B)

1. Napišite metodu Ispitaj koja preko argumenata prima listu cijelih brojeva, a vraća broj negativnih elemenata liste.

Ako nisu pronađeni negativni elementi u listi, metoda ispisuje na konzolu „Nisu pronađeni negativni elementi“ te vraća broj 0.

2. Napišite konzolnu aplikaciju koja omogućuje korisniku pohranu proizvoljnog naziva usluge i 10 cijena te usluge po tjednima u datoteku.

Primjer formata zapisa naziva usluge (u primjeru Mobilni internet) i cijena u datoteci prikazan je slikom.



Napomena: potrebno je koristiti try-catch blok, te nije potrebno dodati imenski prostor System.IO.

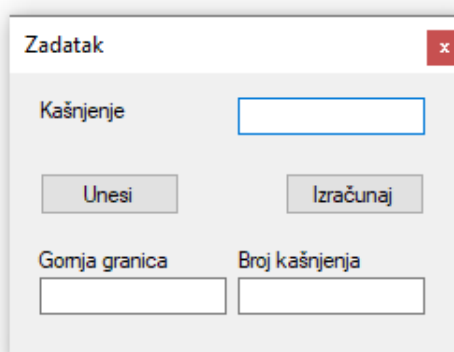
3. Nadopunite sljedeći kod unutar odgovarajućih metoda na način da se klikom na gumb Unesi dodaje u listu vrijednost kašnjenja zrakoplova upisana u TextBox txtKasnjenja.

```
//Nadopuni  
  
private void BtnUnesi_Click(object sender, EventArgs e)  
{  
  
}  
  
private void BtnIzracunaj_Click(object sender, EventArgs e)  
{  
  
}
```

Klikom na gumb Izračunaj prebrojavaju se sva kašnjenja u listi koja su veća od vrijednosti gornje granice upisane u TextBox txtGornjaGranica.

Broj kašnjenja ispisuje se u TextBox txtbrKasnjenja.

Sučelje je prikazano slikom.



4. Napišite program koji učitava broj bodova unutar intervala $[0, 5]$. Ako učitani bodovi nisu unutar zadanog intervala program ispisuje "Niste unijeli bodove u intervalu $[0, 5]$!".

Ako su ispravno učitani bodovi potrebno je ispisati odgovarajuću ocjenu.

Ako je broj bodova u intervalu $[0 - 1.5>$ potrebno je ispisati "Nedovoljan (1)!".

Ako je broj bodova u intervalu $[1.5 - 2.5>$ potrebno je ispisati "Dovoljan (2)!".

Ako je broj bodova u intervalu $[2.5 - 3.5>$ potrebno je ispisati "Dobar (3)!".

Ako je broj bodova u intervalu $[3.5 - 4.5>$ potrebno je ispisati "Vrlo dobar (4)!".

Inače je potrebno ispisati "Odličan (5)!".

5. Napišite konzolnu aplikaciju koja učitava prirodni broj n s konzole te potom kreira matricu $M = (n - 1) \times (n + 1)$.

Matricu je potrebno popuniti slučajnim prirodnim brojevima u intervalu $[30, 59]$.

Zatim je potrebno u jednodimenzionalno polje upisati n slučajno odabranih elemenata iz matrice M .



Ispit #33 (11.07.19. - grupa A)

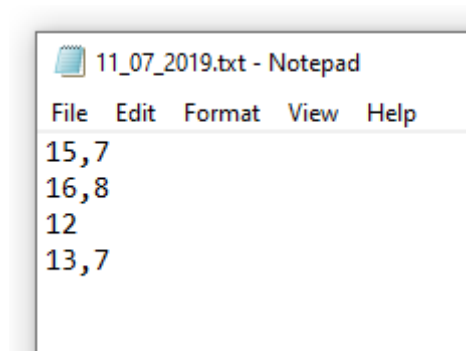
1. Napišite kod klase Grad koja sadrži privatne attribute naziv, broj stanovnika (cijeli broj) i javne attribute x i y (realni brojevi) koji predstavljaju geografsku dužinu, odnosno širinu.

Konstruktor klase prima argumente za sve attribute klase.

Potrebno je napisati i javnu metodu PovecajBrStanovnika koja povećava vrijednost atributa broja stanovnika za vrijednost predanu kao argument metode i ispisuje poruku: „Grad naziv_grada ima novi broj stanovnika broj_stanovnika!“.

2. Napišite metodu Ucitaj koja kao argument prima niz znakova (riječ) koji predstavljaju naziv jedne datoteke.

Sve datoteke koje se obrađuju neovisno o nazivu sadrže u zasebnim recima količine paketa u kilogramima (realni brojevi) koje je taj dan potrebno dostaviti kao što je to prikazano na slici, a naziv datoteke dogovara datumu za koji se vrši dostava npr. 11_07_2019.txt.



Metoda treba vratiti ukupnu količinu paketa koju je potrebno dostaviti za predani naziv datoteke koja se učitava.

Napomena: nije potrebno koristiti try-catch blok, te nije potrebno dodati imenski prostor System.IO.

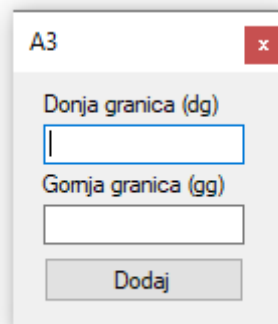


3. Nadopunite sljedeći kod unutar odgovarajuće metode na način da se klikom na gumb Dodaj u listu vrijednost brzine automobila dodaje nova brzina čija se vrijednost određuje slučajno unutar intervala [dd, gg>.

```
//Nadopuni  
  
private void BtnDodaj_Click(object sender, EventArgs e)  
{  
  
}
```

Donja i gornja granica upisuju se preko TextBoxova naziva txtDonja i txtGornja.

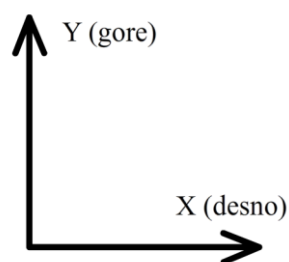
Sučelje je prikazano slikom.



4. Napišite program koji simulira kretanje zmiје kao u igrici Snake.

Program treba učitavati niz znakova s konzole koji predstavljaju kretanje zmiје, sve dok ne učitava nevaljani niz znakova. Nevaljani niz znakova je svaki niz koji nije iz skupa niza znakova {gore, dolje, lijevo, desno}.

Početna pozicija zmiје je određena koordinatom $T_0(x, y) = (0, 0)$ pri čemu su x i y cijeli brojevi. Koordinatni sustav prikazan je na slici.



Nakon učitavanja valjanog znakovnog niza potrebno je pomaknuti koordinatu zmiје za vrijednost 5 u smjeru određenom učitanim znakovnim nizom.

Na kraju je potrebno ispisati konačnu poziciju (koordinatu) zmiје u formatu kao u primjeru.

Primjer: ako se redom učitavaju vrijednosti gore, desno, gore, desno, dolje, lijevo, kraj potrebno je ispisati: „Zmija se na kraju nalazi na poziciji (5,5)!“.



5. Napišite konzolnu aplikaciju koja učitava prirodni broj n s konzole te potom kreira polje veličine n u koje će se spremati studenti na kolegiju Algoritmi i programiranje (klasa Student).

Potrebno je popuniti polje s instancama klase Student čiji atributi ime i bodovi imaju vrijednosti koje se unose preko konzole.

```
public class Student
{
    private string ime;
    public double bodovi;

    public Student(string IME, double BODOVI)
    {
        ime = IME;
        bodovi = BODOVI;
    }
}

class Program
{
    static void Main(string[] args)
    {
        //Nadopuni
    }
}
```



Ispit #34 (11.07.19. - grupa B)

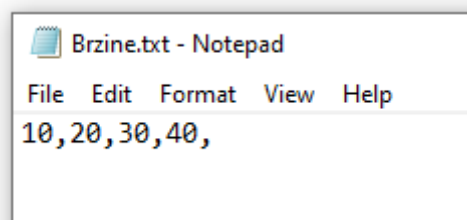
1. Napišite kod klase Dostava koja sadrži javne atribute naziv, x i y (realni brojevi) koji predstavljaju geografsku dužinu, odnosno širinu te privatni atribut obavljenaDostava (binarna vrijednost istina ili laž).

Konstruktor klase prima argumente naziva, geografske dužine x i geografske širine y, a atribut obavljenaDostava postavlja na neistinu, jer još nije obavljena dostava.

Potrebno je napisati i javnu metodu ObaviDostavu koja postavlja atribut obavljene dostave na istinu i ispisuje poruku: „Dostava za naziv_dostave je obavljena!“.

2. Napišite metodu Ispis koja kao argument prima niz znakova s i broj simulacija brzina n (cijeli broj). Metoda ne vraća rezultat.

Niz znakova s predstavlja naziv datoteke u koju će se spremiti vrijednosti, npr. Brzine.txt. Potrebno je iz konzole učitati n vrijednosti brzina (realni brojevi) i zapisivati ih u datoteku sve skupa u jednom retku, odvojene zarezom (kao u primjeru na slici).



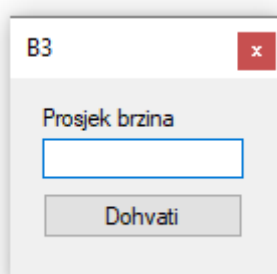
Napomena: potrebno je koristiti try-catch blok, te nije potrebno dodati imenski prostor System.IO.



3. Nadopunite sljedeći kod unutar odgovarajućih metoda na način da se klikom na gumb Dohvati dohvate 2 (dvije) slučajne brzine zapisane u listi brzine te se prosjek vrijednosti dohvaćenih brzina zapiše u TextBox txtProsjeKBrzina.

```
//Nadopuni  
List<int> brzine = new List<int>();  
  
private void BtnDohvati_Click(object sender, EventArgs e)  
{  
  
}
```

Pretpostavka je da je u listi već zapisano 100 brzina. Sučelje je prikazano slikom.



Nije potrebno paziti da budu dohvaćene različite vrijednosti brzina iz liste.



4. Napišite program koji simulira vožnju automobila po trkaćoj stazi. Program treba učitavati niz znakova s konzole koji predstavljaju kretanje automobila, sve dok automobil ne prođe udaljenost od 300 m koliko iznosi staza.

Pretpostavka je da korisnik učitava novi znakovni niz s konzole svakih $t = 5$ sekundi te da se neće učitati krivi znakovni niz.

Moguće je učitati niz znakova iz skupa ubrzaj, uspori, koji odgovaraju ubrzanju odnosno usporenju automobila. Svaki put kada se učitava novi znakovni niz potrebno je izračunati brzinu i udaljenost, ako je učitani znakovni niz uspori primjenjuju se izrazi koji se odnose na usporenje, a ako je učitani znakovni niz ubrzaj primjenjuju se izrazi koji se odnose na ubrzanje.

Prijeđeni put se računa prema izrazu:

$$s_k = \begin{cases} s_p + v_p t + \frac{a}{2} t^2, & \text{ubrzanje} \\ s_p + v_p t - \frac{a}{2} t^2, & \text{usporenje} \end{cases}$$

Brzina se računa prema izrazu:

$$v_k = \begin{cases} v_p + at, & \text{ubrzanje} \\ v_p - at, & \text{usporenje} \end{cases}$$

Početna brzina, akceleracija i udaljenost su postavljeni na vrijednost 0 (realni brojevi). v_p i s_p su početna brzina i prijeđena udaljenost na početku intervala, a v_k i s_k su krajnja brzina i prijeđena udaljenost na kraju intervala.

Na kraju je potrebno ispisati vrijeme prolaska staze u formatu kao u primjeru.

Vrijeme se računa prema vremenu učitavanja, odnosno ako je primjerice već nakon prvog učitavanja prijeđena udaljenost veća od 300 m, vrijeme prolaska staze iznosi 5 sekundi (jer je "prošlo" 5 sekundi). Akceleracija za ubrzanje i usporenje iznosi $a = 2.78 \text{ m/s}^2$.

Primjer: ako se redom učitaju vrijednosti ubrzaj, ubrzaj, uspori, ubrzaj potrebno je ispisati: „Vrijeme prolaska staze iznosi 20 sekundi!“ (opaska: jer je prošlo 4 učitavanja).



5. Napišite konzolnu aplikaciju koja učitava prirodni broj n s konzole te potom kreira polje veličine n u koje će se spremati točke (klasa Točka).

Potrebno je popuniti polje s instancama klase Točka čiji atributi x i y imaju vrijednosti koje se unose preko konzole.

```
public class Točka
{
    public int x;
    public int y;

    public Točka(int X, int Y)
    {
        x = X;
        y = Y;
    }
}

class Program
{
    static void Main(string[] args)
    {
        //Nadopuni
    }
}
```



Ispit #35 (29.08.19. - grupa A)

1. Nadopunite sljedeći kod unutar Main metode na način da stvorite novi auto naziva Insignia, kojem je vlasnik Marko, te koji se trenutno nalazi na GPS lokaciji $x = 15.9773$, $y = 45.8129$.

Potom u konzolu ispišite vrijednost x GPS lokacije vozila i naziv automobila (vrijednosti atributa unutar instance).

```
static void Main(string[] args)
{
    //Nadopuni

}

public class GPSLokacija
{
    public double x;
    public double y;
    public GPSLokacija(double Y, double X)
    {
        x = X;
        y = Y;
        y = Y;
    }
}

public class Auto
{
    public string imeAuta;
    public string imeVlasnika;
    public GPSLokacija lokacija;

    public Auto(GPSLokacija LOKACIJA, string IMEVLASNIKA, string IMEAUTA)
    {
        imeAuta = IMEAUTA;
        lokacija = LOKACIJA;
        imeVlasnika = IMEVLASNIKA;
    }
}
```

2. Napišite konzolnu aplikaciju koja učitava sadržaj datoteke potrosnjaInterneta.txt kojoj se pristupa preko relativne putanje.

U datoteci je zapisano 12 realnih brojeva svaki u zasebnom retku koji predstavljaju izmjerenu mjesečnu potrošnju mobilnog interneta u GB.

Potrebno je ispisati prosječnu mjesečnu potrošnju korisnika u sljedećem formatu: „Prosječna mjesečna potrošnja korisnika je x GB!“.

Napomena: potrebno je koristiti try-catch blok, te nije potrebno dodati imenski prostor System.IO.



3. Napišite program koji učitava tri realna broja s tipkovnice a, b i c koji redom predstavljaju stranice kvadra. Potrebno je napisati metodu IzracunajDijagonalu koja kao argumente prima stranice kvadra, a vraća izračunatu duljinu dijagonale kvadra.

U glavnom dijelu programa potrebno je ispisati duljinu dijagonale samo ako su sve stranice kvadra veće od 0. Format ispisa mora biti kao u primjeru.

Primjer: ako se redom učitaju vrijednosti 2.5, 3 i 4 potrebno je ispisati: „Duljina dijagonale kvadra određenog stranicama a = 2.5, b = 3 i c = 4 je 5.59!“.

Izraz za izračun dijagonale kvadra dan je izrazom:

$$D = \sqrt{a^2 + b^2 + c^2}$$

4. Napišite program koji učitava podatke o nazivima smjerova na fakultetu i broju predmeta na smjeru u dva polja.

U prvo polje se upisuju nazivi smjerova (niz znakova), a u drugo polje se upisuju cijeli brojevi koji predstavljaju broj predmeta na odgovarajućem smjeru. Na Fakultetu postoji 10 različitih smjerova.

Na kraju učitavanja u proizvoljnom formatu potrebno je ispisati naziv smjera koji ima najviše predmeta.

5. Napišite konzolnu aplikaciju koja učitava 1000 cijelih slučajnih brojeva unutar intervala <0, 100>.

U dvije liste spremaju se brojevi ovisno o njihovoj vrijednosti. Ako je slučajan broj djeljiv i s 2 i s 3 te nije djeljiv s 5 dodaje ga se u jednu listu, inače ga se dodaje u drugu listu.

Na kraju je potrebno ispisati u konzolu koliko ima elemenata u pojedinoj listi u proizvoljnom formatu.



Ispit #36 (29.08.19. - grupa B)

1. Nadopunite sljedeći kod unutar Main metode na način da stvorite novi grad Zagreb s koordinatama $x = 15:9773$, $y = 45:8129$ i brojem stanovnika 800000.

Potom u konzolu ispišite x i y koordinatu grada (vrijednost atributa unutar instance).

```
static void Main(string[] args)
{
    //Nadopuni

    Console.ReadKey();
}

public class Grad
{
    private string naziv;
    public Tocka t;
    private int brojStanovnika;

    public Grad(int BROJSTANOVNIKA, string NAZIV, Tocka T)
    {
        t = T;
        naziv = NAZIV;
        brojStanovnika = BROJSTANOVNIKA;
    }
}

public class Tocka
{
    public double x;
    public double y;

    public Tocka(double Y, double X)
    {
        x = X;
        y = Y;
    }
}
```

2. Napišite konzolnu aplikaciju koja učitava cjelobrojnu vrijednosti n s tipkovnice koja predstavlja broj redaka i stupaca kvadratne matrice $M = n \times n$.

Potrebno je popuniti matricu M s rezultatima množenja prirodnih brojeva počevši od vrijednosti 1, odnosno kao tablicu množenja.

Matricu je potrebno ispisati u formatu kao u primjeru. Primjerice, ako se za n učitava vrijednost 4, potrebno je ispisati matricu:

```
1,2,3,4
2,4,6,8
3,6,9,12
4,8,12,16
```



3. Napišite konzolnu aplikaciju koji učitava tri realna broja s tipkovnice a, b i c koji predstavljaju stranice trokuta.

Potrebno je provjeriti da li su sve unesene vrijednosti veće od nule, te ako jesu potrebno je izračunati i ispisati površinu trokuta prema formatu u primjeru koristeći Heronovu formulu danu jednadžbama 1 i 2, a ako nisu potrebno je ispisati „Unesene su krive vrijednosti stranica trokuta!“.

Provjeru da li su učitane ispravne vrijednosti potrebno je napisati u zasebnoj metodi Provjera koja prima tri argumenta (stranice trokuta) i vraća rezultat provjere.

Ako se primjerice redom unesu vrijednosti 3, 4 i 5 potrebno je ispisati: „Površina trokuta određenog stranicama a = 3, b = 4 i c = 5 iznosi: 6!“.

$$(1) P = \sqrt{s(s-a)(s-b)(s-c)}$$

$$(2) s = \frac{a+b+c}{2}$$

4. Napišite konzolnu aplikaciju koja učitava 100 imena i prezimena korisnika (niz znakova) i pripadajuće mjesečne potrošnje interneta (realan broj).

U datoteku korisnici.txt potrebno je zapisivati imena i prezimena korisnika. Svaki korisnik u zasebnom retku. U odgovarajućem retku u datoteci potrosnja.txt potrebno je zapisivati odgovarajuću mjesečnu potrošnju interneta korisnika. Datotekama se pristupa preko relativne putanje.

Napomena: potrebno je koristiti try-catch blok, nije potrebno dodati imenski prostor System.IO te se ime i prezime može učitati kao jedan znakovni niz.

5. Napišite konzolnu aplikaciju koja učitava 1000 cijelih slučajnih brojeva unutar intervala [0, 1000>.

U tri liste spremaju se brojevi ovisno o njihovoj vrijednosti. Ako je broj unutar intervala [0, 10> dodaje ga se u listu jednaZnamenka, ako je broj unutar intervala [10, 100>, dodaje ga se u listu dvijeZnamenke, inače ga se dodaje u listu triZnamenke.

Na kraju je potrebno ispisati koliko ima kojih brojeva, primjerice:

„Jednoznamenkastih brojeva ima 10!

Dvoznamenkastih brojeva ima 90!

Troznamenkastih brojeva ima 900!“.



Ispit #37 (06.02.20. - grupa A)

1. Napišite konzolnu aplikaciju koja učitava 1000 cijelih slučajnih brojeva unutar intervala $[0, 250>$, a koji predstavljaju trenutni broj automobila na nekom parkiralištu.

Ako je broj automobila na parkiralištu veći od 200 broj se sprema u listu *zagusenje*, a ako je manji od 200 i različit od 0 sprema se u listu *protocno*. U slučaju da je broj automobila 0 parkiralište je zatvoreno i taj broj se ne dodaje niti u jednu listu.

Na kraju je potrebno ispisati u konzolu koliko ima elemenata u pojedinoj listi u proizvoljnom formatu te koliko puta je parkiralište bilo zatvoreno.

2. Napišite konzolnu aplikaciju koja učitava podatke iz dvije datoteke *korisnici.txt* i *narudzbe.txt*. Primjer izgleda datoteka prikazan je na sljedećoj slici:

 korisnici.txt - Blok za pisanje

Datoteka	Uređivanje	Prikaži
Vita d.d.		115,5
mStart plus d.o.o.		2000,5
M.N.F. -PROMET d.o.o.		1601,5
eStart d.d.		250
IN d.o.o.		1300,5

 narudzbe.txt - Blok za pisanje

115,5
2000,5
1601,5
250
1300,5

U datoteci *korisnici.txt* zapisani su nazivi korisnika, a u datoteci *narudzbe.txt* u odgovarajućem retku količina njihove narudžbe (realan broj).

Konzolna aplikacija, nakon čitanja podataka iz datoteka, ispisuje na konzolu naziv korisnika s najvećom količinom narudžbe.

Datotekama se pristupa preko relativne putanje.

Napomena: potrebno je koristiti *try-catch* blok te nije potrebno dodati imenski prostor *System.IO*.



3. Napišite program koji učitava dva realna broja s tipkovnice r i v koji redom predstavljaju polumjer baze i visinu stošca.

Potrebno je napisati metodu *izracunajVolumen* koja kao argumente prima polumjer i visinu, a vraća izračunati volumen stošca.

U glavnom dijelu programa potrebno je ispisati volumen stošca samo ako su sve učitane vrijednosti veće od 0. Format ispisa mora biti kao u primjeru.

Primjer: ako se redom učitaju vrijednosti 2.5 i 4 potrebno je ispisati: „Volumen stošca određenog polumjerom $r = 2.5$ i visinom $v = 4$ je 26.18!“.

Izraz za izračun volumena stošca dan je izrazom:

$$V = \frac{r^2 \times \pi \times v}{3}$$

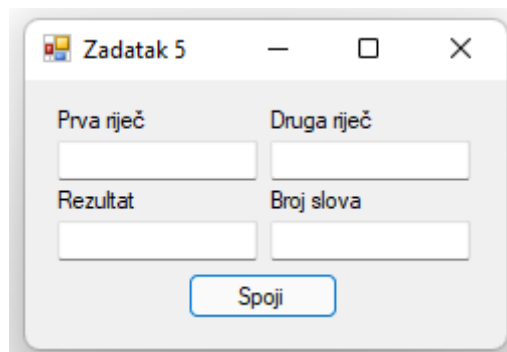
4. Napišite program koji učitava dva broja s tipkovnice n i m koji predstavljaju redom broj studenata koji je pristupio ispitu i broj zadataka na ispitu te kreira matricu veličine $n \times m$.

U matricu se redom upisuju s tipkovnice bodovi studenata na ispitu po pojedinom zadatku (jedan redak matrice su bodovi jednog studenta na ispitu za m zadataka).

Nakon učitavanja u proizvoljnom formatu potrebno je ispisati za svakog studenta indeks zadatka na kojem je student ostvario najviše bodova.

5. Nadopunite sljedeći kod unutar metode *btnSpoji_Click* na način da spojite nizove znakova upisane u *TextBox*ove *txtprvarijec* i *txtdrugarijec* u jedan i ispišete ga u *TextBox* *txtrezultat* te broj slova u spojenoj riječi u *TextBox* *xtbrojslova*.

Sučelje je prikazano sljedećom slikom:



```
private void btnSpoji_Click(object sender, EventArgs e)
{
    //Nadopuni
}
```



Ispit #38 (20.02.20. - grupa A)

1. Što će se ispisati u konzolu nakon izvršavanja sljedećeg programskog odsječka? Potrebno je naznačiti redoslijed ispisivanja.

```
static void Main(string[] args)
{
    PravokutanTrokut p = new PravokutanTrokut(4, 6);
    p.izracunajPovrsinu();
    Console.WriteLine(p.izracunajPovrsinu(2, 3));
    p.ispis();

    Console.ReadKey();
}

class PravokutanTrokut
{
    private double A;
    private double B;
    private double povrsina;

    public PravokutanTrokut(double a, double b)
    {
        A = a;
        B = b;
        povrsina = 0;
        ispis();
    }

    public double izracunajPovrsinu(int a, int b)
    {
        double POVRSINA = (b * a) / 2;
        ispis();
        return POVRSINA;
    }

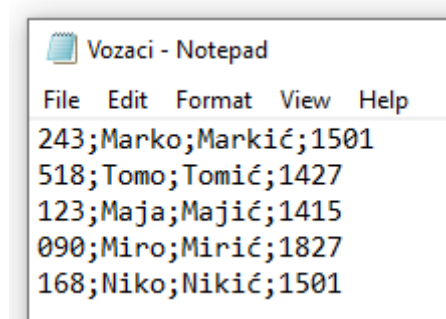
    public void izracunajPovrsinu()
    {
        povrsina = (A * B) / 2;
        ispis();
    }

    public void ispis()
    {
        Console.WriteLine(povrsina);
    }
}
```



2. Napišite C# kod metode PronadiNajbrzeg koja kao argument prima cijelu putanju do datoteke koja sadrži zapise o vozačima utrke i njihovim vremenima (u sekundama).

U jednom retku datoteke zapisana je informacija o vozaču i njegovom vremenu u formatu: ID_vozac;Ime;Prezime;Vrijeme kao što je prikazano na slici.



Metoda kao rezultat vraća ID vozača (cijeli broj) koji je ostvario najbolje vrijeme na utrci, a u konzolu u proizvoljnom formatu ispisuje ime, prezime i vrijeme vozača. Potrebno je koristiti try-catch blok te nije potrebno dodati imenski prostor System.IO.

Prilikom učitavanja za dijeljenje podataka u retku potrebno je koristiti ugrađenu metodu Split nad tipom podatka string. Primjer poziva metode Split nad retkom "Ovo-je-primjer!", s crticom '-' kao znakom za dijeljenje retka je sljedeći:

```
string linija = "Ovo-je-primjer!";
string[] dijeloviLinije = linija.Split('-');
```

3. Napišite kod C# konzolne aplikacije koja predstavlja igru pogađanja cijelog broja. Prvo je potrebno dohvatiti slučajni cijeli broj x unutar intervala [0, 100>. Potom je preko tipkovnice potrebno učitavati cijeli broj z sve dok se ne pogodi traženi slučajni broj x.

Ako je uspješno pogođen traženi broj x program se zaustavlja i potrebno je u konzolu ispisati poruku „Nakon y pokušaja uspješno je pogođen broj x!“, pri čemu x predstavlja vrijednost broja koji se pogađa, a y vrijednost broja pokušaja.

Ako nije pogođen broj x potrebno je ispisati sugestiju korisniku koju vrijednost broja da upiše sljedeću i to: ukoliko je učitani broj z manji od x potrebno je ispisati poruku „Traženi broj je veći od broja z!“, inače ako je z veći od x potrebno je ispisati poruku „Traženi broj je manji od broja z!“ (z predstavlja vrijednost učitanoog broja).

4. Napišite kod C# programa koji u dvije liste učitava JMBAG studenata i broj bodova na ispitu (realan broj) studenata sve dok se ne učitava "prazan" JMBAG (pojašnjenje: JMBAG u listu JMBAG-ova, a broj bodova u listu bodova).

Potom program u konzolu treba ispisati JMBAG-ove i bodove onih studenata čiji je broj bodova veći od prosjeka. Format ispisa je proizvoljan.



5. Što će se ispisati u konzolu nakon izvršavanja sljedećeg programskog odsječka? Potrebno je naznačiti redoslijed ispisivanja. Kao točno rješenje priznaje se samo ono u kojem je vidljiv postupak, odnosno prikazane su vrijednosti varijabli u svakoj iteraciji.

```
int[, ] polje = new int[2, 3];
polje[0, 0] = polje.Length;
polje[0, 1] = polje.GetLength(0);
polje[1, 0] = polje.GetLength(1);
polje[1, 1] = polje.GetLength(1) * polje.Length;

for (int j = polje.GetLength(1) - 1; j > 0; j--)
{
    for (int i = polje.GetLength(0) - 1; i >= 0; i--)
    {
        Console.WriteLine(polje[i, j]);
    }
}
```



Ispit #39 (02.06.20. - grupa A)

1. Napišite kod metode provjeraSifre koja kao argument prima šifru prometne nesreće (cijeli broj), a kao rezultat vraća istinu ili laž ovisno o tome da li je šifra ispravna ili ne. Šifra je ispravna ako je šifra peteroznamenasti broj i ako je suma znamenki šifre veća od 10.

Primjeri:

243 - neispravna šifra jer nije peteroznamenasti broj

15002 - neispravna šifra jer je suma znamenki 8

83010 - ispravna šifra jer je suma znamenki 12 i peteroznamenasti je broj

2. Napišite kod dvije klase Lokacija i PrometnaNesreca.

Klasa Lokacija sadrži javne atribute x i y (realni brojevi) koji predstavljaju geografsku dužinu, odnosno širinu. Konstruktor klase Lokacija prima argumente za sve atribute klase.

Klasa PrometnaNesreca sadrži javne atribute Sifra (cijeli broj) koji predstavlja jedinstveni identifikator nesreće, Postaja (niz znakova) koji predstavlja naziv policijske postaje koja je evidentirala nesreću, DatumVrijeme (niz znakova) koji predstavlja datum i vrijeme kada se dogodila prometna nesreća i atribut LokacijaNesrece (tip klase Lokacija) koji predstavlja mjesto gdje se nesreća dogodila.

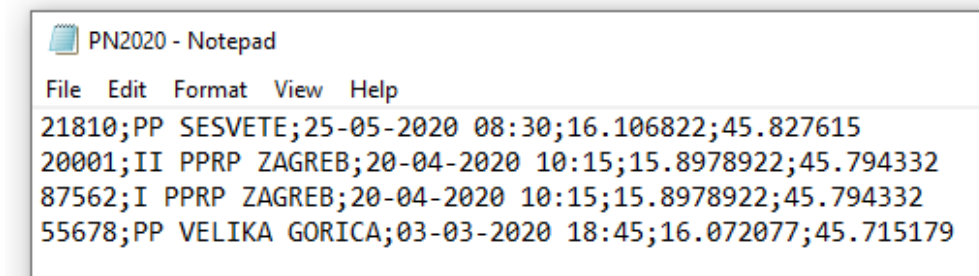
Unutar klase potrebno je napisati i javnu metodu Ispis koja ispisuje opis prometne nesreće u formatu: „Prometnu nesreću pod šifrom Sifra koja se dogodila DatumVrijeme na lokaciji (X, Y) evidentirala je policijska postaja Postaja!“. Vrijednosti Sifra, DatumVrijeme, X, Y i Postaja potrebno je dohvatiti iz odgovarajućih atributa klase.

Primjer ispisa: Prometnu nesreću pod šifrom 21810 koja se dogodila 25-05-2020 08:30 na lokaciji (16.106822, 45.827615) evidentirala je policijska postaja PP SESVETE!



3. Napišite C# program koji učitava datoteku o prometnim nesrećama preko apsolutne putanje. Datoteka naziva PN2020.txt spremljena je u direktorij E:\PN.

U jednom retku datoteke zapisana je informacija o šifri prometne nesreće (cijeli broj), policijskoj postaji (niz znakova), datumu-vremenu (niz znakova), geografskoj dužini X (realan broj) i geografskoj širini Y (realan broj), međusobno odijeljenih znakom ";" kao što je prikazano na slici.



Podatke je potrebno učitati u listu listaNesreca, na način da se u listu dodaje kreirana instanca klase PrometnaNesreca (zadatak 2) na odgovarajući način. U listu se dodaju samo one prometne nesreće koje imaju VALJANU šifru. Za provjeru valjanosti šifre koristiti poziv metode provjeraSifre (zadatak 1).

Na kraju je u konzolu potrebno ispisati koliko ima učitanih valjanih prometnih nesreća u proizvoljnom formatu.

Potrebno je koristiti try-catch blok te nije potrebno dodati imenski prostor System.IO.

Prilikom učitavanja za dijeljenje podataka u retku potrebno je koristiti ugrađenu metodu Split nad tipom podatka string. Primjer poziva metode Split nad retkom "Ovo-je-primjer!", s crticom '-' kao znakom za dijeljenje retka je sljedeći:

```
string linija = "Ovo-je-primjer!";  
string[] dijeloviLinije = linija.Split('-');
```



4. Napišite kod konzolne aplikacije koja služi za ispis koliko je prometna nesreća PN1 na lokaciji (16.106822, 45.827615) daleko od lokacije koja se učitava preko konzole. Potrebno je s konzole učitati geografsku dužinu x (realan broj) i geografsku širinu y (realan broj) koji predstavljaju željenu lokaciju i ispisati koliko je PN1 udaljena od učitane lokacije. Za izračun udaljenosti potrebno je koristiti Euklidsku udaljenost prema formuli:

$$d(T_1(x_1, y_1), T_2(x_2, y_2)) = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

Za ispis udaljenosti potrebno je koristiti kategorizaciju danu tablicom:

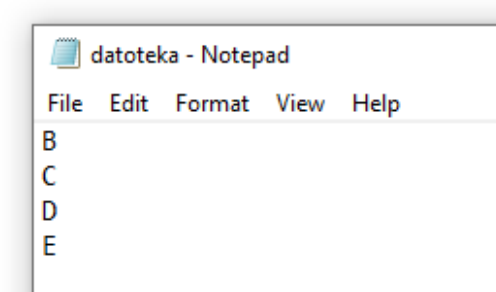
Udaljenost	Tekst
[0, 0.025]	blizu
<0.025, 0.2]	daleko
<0.2, ∞]	jako daleko

Ispis formatirati na sljedeći način: „Učitana lokacija (x, y) je tekst zadanoj lokaciji prometne nesreće PN1!“, pri čemu je potrebno ispisati odgovarajuće vrijednosti x, y i tekst, primjerice: „Učitana lokacija (16.106802,45.827625) je blizu zadanoj lokaciji prometne nesreće PN1!“.



5. Što će se ispisati nakon izvršavanja sljedećeg programskog odsječka? Potrebno je naznačiti redoslijed ispisivanja.

Pretpostavka je da je datoteka „datoteka.txt“ spremljena u projektni folder kako bi joj se moglo pristupiti preko relativne putanje, a njezin sadržaj je prikazan sljedećom slikom:



```
try
{
    StreamReader citanje = new StreamReader("datoteka.txt");
    string s = "A";

    while (!citane.EndOfStream)
    {
        Console.WriteLine(s);
        s = citanje.ReadLine();
        s = citanje.ReadLine();
        Console.WriteLine(s);
    }
    citanje.Close();
    Console.WriteLine("F");
}
catch (Exception ex)
{
    Console.WriteLine("G");
}
Console.WriteLine("H");
```



6. Što će se ispisati u konzolu nakon izvršavanja sljedećeg programskog odsječka?

Potrebno je naznačiti redoslijed ispisivanja. Kao točno rješenje priznaje se samo ono u kojem je vidljiv postupak i sadržaj varijabli u svakoj iteraciji.

```
static void Main(string[] args)
{
    int[] p = new int[3];
    p[0] = p.Length;
    p[1] = -1;
    p[p.Length - 1] = p.Length / 2;
    Metoda(p);
    for (int i = 0; i < p.Length; i++)
    {
        Console.WriteLine(p[i]);
    }
}
private static void Metoda(int[] q)
{
    for (int i = 0; i < 2; i++)
    {
        q[i] = i;
    }
}
```



Ispit #40 (15.06.20. - grupa A)

1. Napišite C# kod metode ObradaDatuma koja kao argumente prima mjesec (cijeli broj) i dan (cijeli broj) koji predstavljaju datum u 2020. godini. Kao rezultat metoda vraća polje cijelih brojeva od 2 elementa.

Prvi element predstavlja oznaku dana u tjednu i treba poprimiti vrijednost: 1 - ukoliko se radi o radnom danu u tjednu (ponedjeljak, . . . , petak) ili 2 - ukoliko se radi o neradnom danu u tjednu (subota, nedjelja).

Drugi element predstavlja broj dana od početka 2020. godine. Pretpostavka je da svaki mjesec ima točno 30 dana i da je 01.01.2020. bila srijeda.

Primjeri:

Dan - 1, Mjesec - 1 metoda treba vratiti vrijednosti 1 i 1

Dan - 1, Mjesec - 2 metoda treba vratiti vrijednosti 1 i 31
Dan - 1, Mjesec - 3 metoda treba vratiti vrijednosti 2 i 61

2. Napišite C# kod za učitavanje podataka o zapisima radara preko konzole. Za jedan zapis s radara potrebno je učitati vrijednost dana (cijeli broj), mjeseca (cijeli broj) i brzine (realan broj).

U jednodimenzionalno polje naziva brzine potrebno je učitati 100 VALJANIH zapisa brzina (ne svih nego samo valjanih). Zapis je valjan ukoliko je dan na koji se dogodio zapis bio radni (ponedjeljak, . . . , petak) i ukoliko je broj dana proteklih od početka 2020. godine veći od 100.

Za dohvat navedenih vrijednosti potrebno je koristiti metodu ObradaDatuma iz Zadatka 1.

Na kraju je potrebno ispisati vrijednost maksimalne valjane brzine i pripadajućeg dana i mjeseca, te vrijednost prosječne valjane brzine u formatu kao u primjeru.

Primjer: „Maksimalna valjana brzina iznosi 93.2 km/h i zabilježena je 15.06.2020! Prosječna valjana brzina iznosi 35,72856666 km/h!“.



3. Nadopunite C# kod Windows forme koja služi za vizualizaciju podataka učitanih u jednodimenzionalno polje realnih brojeva naziva brzine (zadatak 2).

Pritiskom na gumb Ispis naziva btnIspis potrebno je u kontrolu tipa RichTextBox naziva txtIspis ispisati sve brzine iz polja brzine koje se nalaze unutar intervala <dg, gg>, pri čemu se granice dg i gg (realni brojevi) unose preko kontrola tipa TextBox naziva txtDonja i txtGornja.

Svaka brzina mora biti u zasebnom retku, a oznaka za prelazak u novi red je "\n". Sučelje je prikazano na sljedećoj slici:

```
//Brzine su spremljene u polje naziva brzine (zadatak 2)
public Form1()
{
    InitializeComponent();
}

private void btnIspis_Click(object sender, EventArgs e)
{
    //Nadopuni
}
```



4. Što će se ispisati u konzolu nakon izvršavanja sljedećeg programskog odsječka. Potrebno je naznačiti redoslijed ispisivanja i vrijednosti varijabli u svakom pozivu metoda.

```
static void Main(string[] args)
{
    double a = 4;
    Metoda1(a);
    Console.WriteLine(a);
}

static void Metoda1(double a)
{
    a *= 2;
    Console.WriteLine(a);
    double b = Metoda3(a);
    Console.WriteLine(b);
}

static void Metoda2()
{
    Console.WriteLine("Test1");
}

static double Metoda3(double a)
{
    Console.WriteLine("Test1");
    return Math.Pow(a, 2);
}
```

5. Što će se ispisati u konzolu nakon izvršavanja sljedećeg programskog odsječka. Potrebno je naznačiti redoslijed ispitivanja. Kao točno rješenje priznaje se samo ono u kojem je vidljiv postupak i sadržaj varijabli u svakoj iteraciji.

```
List<int> lista = new List<int>();
lista.Add(-1);
lista.AddRange(new int[] { -1, 0, 1 });
lista.RemoveAt(lista.Count - 2);
lista.Insert(lista.Count - 2, 2);
for (int i = 0; i < lista.Count; i++)
{
    Console.WriteLine(lista[i]);
}
```



Ispit #41 (26.06.20. - grupa A)

1. Što će se ispisati u konzolu nakon izvršavanja sljedećeg programskog odsječka. Potrebno je naznačiti redoslijed ispisivanja. Kao točno rješenje priznaje se samo ono u kojem je vidljiv postupak i sadržaj varijabli.

```
int a = 16 % 7;
int b = 5 / a;
int c, d;

if (a >= 2 && (a != 3 ^ a == 2))
{
    c = ++a;
    d = b % 2;
}
else
{
    c = a++;
    d = a * 3 - 2;
}

Console.WriteLine(a + ", " + b + ", " + c + ", " + d);
```

2. Što će se ispisati u konzolu nakon izvršavanja sljedećeg programskog odsječka. Potrebno je naznačiti redoslijed ispisivanja. Kao točno rješenje priznaje se samo ono u kojem je vidljiv postupak i sadržaj varijabli u svakoj iteraciji.

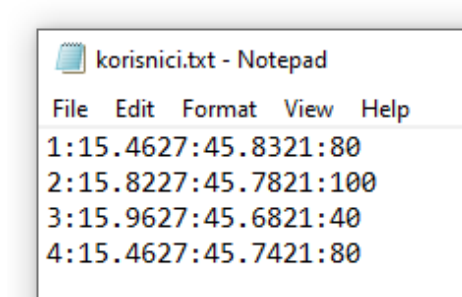
```
int i, j;

for (i = 1; i <= 3; i++)
{
    for (j = 1; j <= 4; j++)
    {
        if (j % 2 != 0)
            continue;
        Console.WriteLine(j);
        if (j != i)
            break;
    }
}
```



3. Napišite C# kod za učitavanje podataka o 100 korisnika dostave koji se nalaze unutar datoteke korisnici.txt, kojoj se treba pristupiti preko relativne putanje.

U jednom retku datoteke zapisana je informacija o šifri korisnika (cijeli broj), geografskoj dužini X (realan broj), geografskoj širini Y (realan broj) i količini (cijeli broj) paketa koji je potrebno dostaviti tom korisniku, međusobno odijeljenih znakom ":", kao što je prikazano na slici.



Podatke je potrebno učitati u polje korisnici koje treba sadržavati instance klase Korisnik prikazane u nastavku.

```
public class Korisnik
{
    public int sifra;
    public double x;
    public double y;
    public int kolicina;

    public Korisnik(int KOLICINA, int SIFRA, double X, double Y)
    {
        sifra = SIFRA;
        x = X;
        y = Y;
        kolicina = KOLICINA;
    }
}
```

Na kraju je u konzolu u proizvoljnom formatu potrebno ispisati kolika je ukupna količina paketa koju je potrebno prevesti, te koliki je minimalan broj vozila potreban za prijevoz ako je poznato da je kapacitet vozila 200.

Za izračun gornje granice za broj vozila potrebno je koristiti ugrađenu metodu Math.Ceiling koja zaokružuje realan broj na najbliži veći cijeli broj, ali i dalje vraća realan broj.

Potrebno je koristiti try-catch blok te nije potrebno dodati imenski prostor System.IO. Prilikom učitavanja za dijeljenje podataka u retku potrebno je koristiti ugrađenu metodu Split nad tipom podatka string. Primjer poziva metode Split nad retkom "Ovo-je-primjer!", s crticom '-' kao znakom za dijeljenje retka je sljedeći:

```
string linija = "Ovo-je-primjer!";
string[] dijeloviLinije = linija.Split('-');
```



4. Napišite C# kod za raspoređivanje korisnika iz polja korisnici (Zadatak 3) po vozilima. Raspoređivanje po vozilima vršit će se na način da se počevši od početka polja korisnici redom dodaju u vozilo dok god nije premašen kapacitet vozila u iznosi od 200 (niti jedan korisnik nema količinu 200).

Nakon prekoračenja kapaciteta vozila potrebno je trenutni raspored korisnika po vozilu zapisati u datoteku raspored.txt (svaki raspored korisnika u zasebnom retku) te potom "otvoriti" novo vozilo, dodati trenutnog korisnika u to novo vozilo i ponoviti postupak sve dok se ne obrade svi korisnici iz polja.

Raspored korisnika unutar vozila potrebno je bilježiti kao niz odgovarajućih šifri korisnika odvojenih znakom ';'. Za primjer sa slike 1 prvi redak u datoteci raspored.txt trebao bi biti 1;2; (jer je zbroj korisnika pod šiframa 1 i 2 180, a dodavanje sljedećeg korisnika pod šifrom 3 premašilo bi kapacitet 200). Prema tome drugi redak u datoteci započeo bi s 3;4; itd.

Nije potrebno koristiti try-catch blok te nije potrebno dodati imenski prostor System.IO, a datoteci se pristupa preko relativne putanje.

5. Napišite C# kod koji s konzole učitava geografsku točku T_1 s dužinom x_1 (realan broj) i širinom y_1 (realan broj), te potom računa udaljenost do slučajne geografske točke T_2 s dužinom x_2 unutar intervala $[0, 20.18>$ i širinom y_2 unutar intervala $[0, 51.28>$.

Učitavanje koordinata točke T_1 i dohvat slučajnih koordinata točke T_2 potrebno je ponavljati sve dok udaljenost između točaka nije manja od 10 km.

Za izračun udaljenosti u kilometrima potrebno je koristiti sljedeću jednadžbu:

$$d(T_1(x_1, y_1), T_2(x_2, y_2)) = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} * 112.32$$

U svakoj iteraciji potrebno je ispisati koliko je učitana točka daleko od slučajno generirane točke prema tablici:

Udaljenost	Tekst
< 50	jako blizu
[50, 150>	blizu
> 150	daleko

Format ispisa treba biti kao u primjeru. Primjerice neka se u jednoj iteraciji petlje učitava $T_1(x_1, y_1) = (12.789, 45.789)$ i slučajno dohvati $T_2(x_2, y_2) = (15.7401, 38.306)$ potrebno je ispisati: „Učitana točka T1 (12.789, 45.789) je daleko slučajnoj točki T2 (15.7401, 38.306)!“.



Ispit #42 (09.07.20. - grupa A)

1. Što će se ispisati u konzolu nakon izvršavanja sljedećeg programskog odsječka. Potrebno je naznačiti redoslijed ispisivanja. Kao točno rješenje priznaje se samo ono u kojem je vidljiv postupak i sadržaj varijabli u svakom pozivu metode.

```
static void Main(string[] args)
{
    Ispis(4);
    Console.ReadKey();
}

private static void Ispis(int x)
{
    Console.WriteLine(x);
    if (x < 2)
    {
        return;
    }
    else if (x <= 4)
    {
        Ispis(x - 2);
    }
    else
    {
        Ispis(x - 1);
    }
    Console.WriteLine(x);
}
```

2. Što će se ispisati u konzolu nakon izvršavanja sljedećeg programskog odsječka. Potrebno je naznačiti redoslijed ispisivanja. Kao točno rješenje priznaje se samo ono u kojem je vidljiv postupak i sadržaj varijabli u svakoj iteraciji.

```
string tekst = "Programiranje";
int i = 0;

try
{
    for (i = 1; i < tekst.Length; i *= 2)
    {
        Console.WriteLine(tekst[i]);
    }
}
catch (Exception ex)
{
    Console.WriteLine(tekst[tekst.Length / 3]);
}
Console.WriteLine(tekst[tekst.Length / 3]);
```

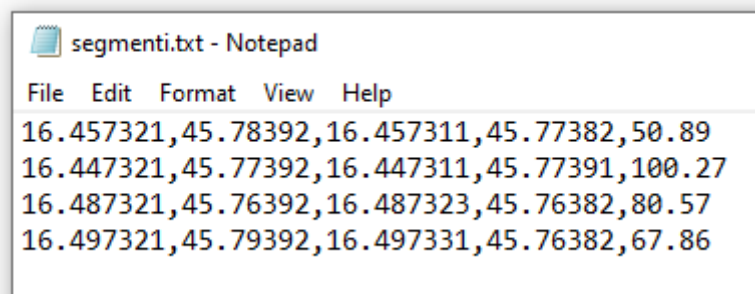


3. Napišite C# kod za učitavanje podataka o cestovnim segmentima u digitalnoj cestovnoj karti RH. Pojedini cestovni segment sadrži informacije o geografskoj dužini i širini početka (x_1, y_1) i kraja (x_2, y_2) (realni brojevi) i prosječnoj brzini v (realan broj) na cestovnom segmentu.

Podaci se trebaju učitavati iz datoteke segmenti.txt u listu segmenti, koja treba sadržavati instance klase Segment prikazane u nastavku.

```
public class Segment
{
    public double x1;
    public double y1;
    public double x2;
    public double y2;
    public double v;
}
```

U jednom retku datoteke zapisana je informacija o cestovnom segmentu u formatu: x_1, y_1, x_2, y_2, v kao što je prikazano na slici:



Instancu cestovnog segmenta potrebno je dodati u listu segmenti samo ako je brzina segmenta unutar intervala $<40, 60>$ ili $[80, 100>$. Na kraju je u konzolu u proizvoljnom formatu potrebno ispisati kolika je suma duljina svih valjanih cestovnih segmenata. Za izračun duljine pojedinog cestovnog segmenta potrebno je koristiti izraz:

$$d(T_1(x_1, y_1), T_2(x_2, y_2)) = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} * 111.32$$

Potrebno je koristiti try-catch blok te nije potrebno dodati imenski prostor System.IO. Prilikom učitavanja za dijeljenje podataka u retku potrebno je koristiti ugrađenu metodu Split nad tipom podataka string. Primjer poziva metode Split nad retkom „Ovo je primjer!“, s crticom '-' kao znakom za dijeljenje retka, je sljedeći:

```
string linija = "Ovo-je-primjer!";
string[] dijeloviLinije = linija.Split('-');
```



4. Napišite C# kod metode MapMatching koja kao argument prima 3 vrijednosti: geografsku dužinu x_1 , geografsku širinu y_1 i listu segmenata segmenti (zadatak 3).

Metoda za zadanu točku $T_1(x_1, y_1)$ treba pronaći najbliži cestovni segment prema kriteriju zadanom izrazom koji se nalazi na kraju ovog zadatka, pri čemu S_i označava prijeđeni cestovni segment iz liste, $P_{S_i}(x_p, y_p)$ točku polovišta segmenta S_i određenu koordinatama x_p i y_p (koju je potrebno izračunati), $d(T_1(x_1, y_1), P_{S_i}(x_p, y_p))$ udaljenost između dvije točke danu izrazom iz prethodnog zadatka (zadatak 3), a $KutIzmeđu(x_1, y_1, S_i)$ poziv metode KutIzmeđu koja vraća kut u stupnjevima između točke $T_1(x_1, y_1)$ i segmenta S_i .

Metodu KutIzmeđu nije potrebno implementirati, već se pretpostavlja da je ona već implementirana, ali je potrebno napisati točan poziv metode. Metoda MapMatching treba vratiti instancu cestovnog segmenta koji je najbliži zadanoj točki prema navedenom kriteriju.

$$F(T_1(x_1, y_1), S_i) = 1000 * d(T_1(x_1, y_1), P_{S_i}(x_p, y_p)) + 0.5 * KutIzmeđu(x_1, y_1, S_i)$$



5. Napišite C# kod koji s konzole učitava tri mase paketa u proizvodnom pogonu u kilogramima m_1 , m_2 i m_3 (cijeli brojevi).

Potom se na paletu maksimalne zapremnine 500 kg dodaje samo onaj paket koji od trenutno tri učitana ima najveću masu.

Učitavanje tri po tri paketa potrebno je ponavljati sve dok se najteži paket može dodati na paletu. U svakom koraku petlje potrebno je u formatu kao u primjeru ispisati kolika je masa paketa dodanog na paletu, te kolike su mase odbačenih paketa.

Primjerice: „Na paletu je stavljen paket mase 3 kg, a odbačeni su paketi mase 1 i 2 kg.“.

Na kraju je u formatu kao u primjeru potrebno ispisati koliko je paketa stavljeno na paletu i kolika je ukupna masa palete.

Primjer ispisa: „Na paletu je stavljeno ukupno 50 paketa, ukupne mase 490 kg!“.



Ispit #43 (09.07.20. - grupa B)

1. Što će se ispisati u konzolu nakon izvršavanja sljedećeg programskog odsječka. Potrebno je naznačiti redoslijed ispisivanja. Kao točno rješenje priznaje se samo ono u kojem je vidljiv postupak i sadržaj varijabli u svakom pozivu metode.

```
static void Main(string[] args)
{
    Ispis(5);
}

private static void Ispis(int x)
{
    if (x < 2)
    {
        Console.WriteLine(x);
        return;
    }
    else if (x <= 4)
    {
        Ispis(x - 2);
        Console.WriteLine(x);
    }
    else
    {
        Console.WriteLine(x);
        Ispis(x - 2);
    }
    Console.WriteLine(x);
}
```

2. Što će se ispisati u konzolu nakon izvršavanja sljedećeg programskog odsječka. Potrebno je naznačiti redoslijed ispisivanja. Kao točno rješenje priznaje se samo ono u kojem je vidljiv postupak i sadržaj varijabli u svakoj iteraciji.

```
string tekst = "Programiranje";
int i = 0;

try
{
    for (i = 1; i < 30; i *= 3)
    {
        Console.WriteLine(tekst[i]);
    }
}
catch (Exception ex)
{
    Console.WriteLine(tekst[i / 4]);
}
Console.WriteLine(tekst[i / 3]);
```



3. Napišite C# kod za učitavanje podataka o cestovnim segmentima u digitalnoj cestovnoj karti RH. Pojedini cestovni segment sadrži informacije o geografskoj dužini i širini početka (x1, y1) i kraja (x2, y2) (realni brojevi) i prosječnoj brzini v (realan broj) na cestovnom segmentu.

Podaci se trebaju učitavati s konzole u listu segmenti, koja treba sadržavati instance klase Segment prikazane u nastavku.

```
public class Segment
{
    public double x1;
    public double y1;
    public double x2;
    public double y2;
    public double v;

    public Segment(double X1, double Y1, double V, double X2, double Y2)
    {
        x1 = X1;
        x2 = X2;
        y1 = Y1;
        y2 = Y2;
        v = V;
    }
}
```

Instancu cestovnog segmenta potrebno je dodati u listu segmenti samo ako je brzina segmenta unutar intervala [0,30> ili [60,80]. Na kraju je u konzolu u proizvoljno formatu potrebno ispisati kolika je prosječna duljina svih valjanih cestovnih segmenata. Za izračun duljine cestovnog segmenta potrebno je koristiti sljedeći izraz:

$$d(T_1(x_1, y_1), T_2(x_2, y_2)) = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} * 111.32$$

4. Napišite C# kod metode MapMatching koja kao argument prima 3 vrijednosti: geografsku dužinu x₁, geografsku širinu y₁ i listu segmenata segmenti (zadatak 3).

Metoda za zadanu točku T₁(x₁, y₁) treba pronaći najbliži cestovni segment prema kriteriju zadanom izrazom koji se nalazi na kraju ovog zadatka, pri čemu S_i označava prijeđeni cestovni segment iz liste, P_{S_i}(x_p, y_p) točku polovišta segmenta S_i određenu koordinatama x_p i y_p (koju je potrebno izračunati), d(T₁(x₁, y₁), P_{S_i}(x_p, y_p)) udaljenost između dvije točke danu izrazom iz prethodnog zadatka (zadatak 3).

Na kraju metoda treba u proizvoljnom formatu ispisati sadržaj segmenta koji je najbliži točki T₁(x₁, y₁),

$$F(T_1(x_1, y_1), S_i) = d(T_1(x_1, y_1), P_{S_i}(x_p, y_p))$$



5. Napišite C# kod koji s konzole učitava tri mase paketa u proizvodnom pogonu u kilogramima m_1 , m_2 i m_3 (realni brojevi).

Potom se na paletu maksimalne zapremnine 100 kg dodaje samo onaj paket koji od trenutno tri učitana ima najmanju masu.

Učitavanje tri po tri paketa potrebno je ponavljati sve dok se najlakši paket može dodati na paletu. U svakom koraku petlje potrebno je u formatu kao u primjeru ispisati kolika je masa paketa dodanog na paletu, te kolike su mase odbačenih paketa.

Primjerice: „Na paletu je stavljen paket mase 0.5 kg, a odbačeni su paketi mase 1.2 i 2.2 kg.“.

Na kraju je u formatu kao u primjeru potrebno ispisati koliko je paketa stavljeno na paletu i kolika je ukupna masa palete.

Primjer ispisa: „Na paletu je stavljeno ukupno 50 paketa, ukupne mase 99.38 kg!“.



Ispit #44 (09.07.20. - grupa C)

Napomena: rok nije kompletan (nedostaju zadaci 3 i 4)

1. Što će se ispisati u konzolu nakon izvršavanja sljedećeg programskog odsječka. Potrebno je naznačiti redoslijed ispisivanja. Kao točno rješenje priznaje se samo ono u kojem je vidljiv postupak i sadržaj varijabli u svakom pozivu metode.

```
static void Main(string[] args)
{
    Ispis(4);
}

private static void Ispis(int x)
{
    Console.WriteLine(x);
    if (x > 5)
    {
        return;
    }
    else if (x > 2)
    {
        Ispis(x + 1);
    }
    else
    {
        Ispis(x + 2);
    }
    Console.WriteLine(x);
}
```

2. Što će se ispisati u konzolu nakon izvršavanja sljedećeg programskog odsječka. Potrebno je naznačiti redoslijed ispisivanja. Kao točno rješenje priznaje se samo ono u kojem je vidljiv postupak i sadržaj varijabli u svakoj iteraciji.

```
string tekst = "Programiranje";
int i = 0;

try
{
    for (i = tekst.Length - 1; i > 0; i -= 3)
    {
        Console.WriteLine(tekst[i]);
    }
}
catch (Exception ex)
{
    Console.WriteLine(tekst[tekst.Length / 2]);
}
Console.WriteLine(tekst[tekst.Length / 2]);
```



5. Napišite C# kod koji s konzole učitava tri mase paketa u proizvodnom pogonu u kilogramima m_1 , m_2 i m_3 (cijeli brojevi).

Potom se na paletu maksimalne zapremnine 500 kg od trenutno tri učitana člana dodaje paket koji je najbliži masi od 5 kg.

Učitavanje tri po tri paketa potrebno je ponavljati sve dok se odabrani paket može dodati na paletu. U svakom koraku petlje potrebno je u formatu kao u primjeru ispisati kolika je masa paketa dodanog na paletu, te kolike su mase odbačenih paketa.

Primjerice: „Na paletu je stavljen paket mase 3 kg, a odbačeni su paketi mase 1 i 2 kg.“.

Na kraju je u formatu kao u primjeru potrebno ispisati kolika je prosječna masa paketa stavljenog na paletu.

Primjer ispisa: „Prosječna masa paketa na paleti iznosi 3.2 kg!“.



Ispit #45 (31.08.20. - grupa A)

1. Što će se ispisati u konzolu nakon izvršavanja sljedećeg programskog odsječka. Potrebno je naznačiti redoslijed ispisivanja. Kao točno rješenje priznaje se samo ono u kojem je vidljiv postupak i sadržaj varijabli u svakoj iteraciji petlje.

```
int i = 1, j = 1;

do
{
    i += j;
    j = (j * 2) % 3;
    Console.WriteLine("(" + i + "," + j + ")");
} while (i < 10 && (j + i) <= 7);
```

2. Što će se ispisati u konzolu nakon izvršavanja sljedećeg programskog odsječka, ukoliko se s konzole učitava niz „mark“. Potrebno je naznačiti redoslijed ispisivanja. Kao točno rješenje priznaje se samo ono u kojem je vidljiv postupak i sadržaj varijabli u svakoj iteraciji.

```
string tekst = Console.ReadLine();

for (int i = tekst.Length-1; i >= 0; i--)
{
    switch (tekst[i])
    {
        case 'm':
            Console.WriteLine('o');
            break;
        case 'a':
            Console.WriteLine('t');
            break;
        case 'r':
            Console.WriteLine('a');
            break;
        case 'k':
            Console.WriteLine('m');
            break;
        default:
            Console.WriteLine('e');
            break;
    }
}
```



3. Napišite C# kod rekurzivne metode koja računa matematički izraz:

$$R(n) = R(n - 1) + R(n - 2) * R(n - 3), \forall n > 3$$

pri čemu je $R(0) = 1$, $R(1) = 2$, $R(2) = 3$.

Unutar Main metode potrebno je pozvati rekurzivnu metodu s argumentom $n = 6$ te ispisati rezultat.

4. Napišite C# kod programa koji služi za simulaciju proizvodne trake u pogonu. S konzole je potrebno učitavati volumene dvaju paketa (cijeli brojevi) v_1 i v_2 sve dok je suma volumena tih dvaju paketa manja od 20.

U svakoj iteraciji učitavanja potrebno je volumene oba paketa zapisati u datoteku volumeni.txt.

Na kraju je u proizvoljnom formatu potrebno ispisati koliko je bilo ukupno paketa.

Nije potrebno koristiti try-catch blok te nije potrebno dodati imenski prostor System.IO.

5. Napišite C# kod programa koji učitava cjelobrojnu vrijednost n koja predstavlja broj gradova. Potrebno je kreirati matricu $M = n \times n$ koja predstavlja udaljenosti između gradova i popuniti ju prema sljedećim pravilima:

- udaljenost između jednog te istog grada treba imati vrijednost nula (0) (pomoć: vrijednost glavne dijagonale matrice trebaju biti nula) i
- u svim ostalim slučajevima upisati slučajan realan broj unutar intervala $[0, 10>$,

Na kraju je sadržaj matrice potrebno ispisati u formatu kao u primjeru.

Neka je $n = 4$, a ispisana matrica popunjena na adekvatan način treba izgledati (vrijednosti su slučajne):

0	9,37	4,38	7,56
8,35	0	2,61	2,22
3,78	2,61	0	1,25
7,78	2,38	1,78	0



Ispit #46 (09.09.20. - grupa A)

1. Što će se ispisati u konzolu nakon izvršavanja sljedećeg programskog odsječka. Potrebno je naznačiti redoslijed ispisivanja. Kao točno rješenje priznaje se samo ono u kojem je vidljiv postupak i sadržaj varijabli.

```
DateTime t1 = DateTime.Now;
Thread.Sleep(200);
DateTime t2 = DateTime.Now;
Thread.Sleep(300);
Console.WriteLine((DateTime.Now - t1).TotalSeconds);
TimeSpan ts = t2 - t1;
Console.WriteLine(ts.TotalMilliseconds);
int godina = t1.Year;
int mjesec = t2.Month;
Console.WriteLine((godina + 1) / (mjesec + 1));
Console.WriteLine((godina + 1) % (mjesec + 1));
```

2. Što će se ispisati u konzolu nakon izvršavanja sljedećeg programskog odsječka. Potrebno je naznačiti redoslijed ispisivanja. Kao točno rješenje priznaje se samo ono u kojem je vidljiv postupak i sadržaj varijabli u svakoj iteraciji.

```
int[] p = new int[7];
for (int i = 0; i < p.Length - 1; i++)
{
    p[i + 1] = (i - 1) * 2;
}
foreach (int v in p)
{
    if (v < p.Length / 2)
    {
        Console.WriteLine(v);
    }
}
```

3. Napišite C# kod rekurzivne metode naziva Rekurzija koja kao argument prima cijeli broj n te ispisuje sve neparne brojeve manje ili jednake n.



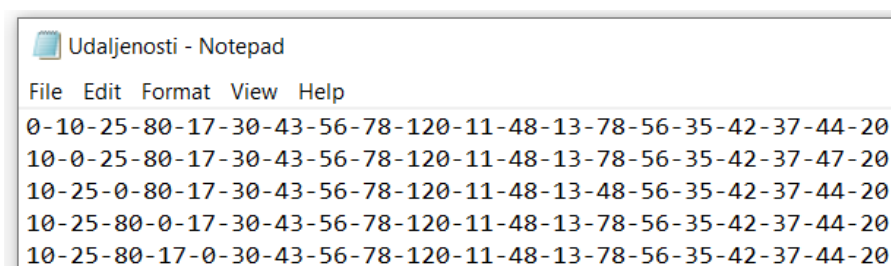
4. Napišite C# kod programa koji služi za simulaciju duljina vozila u softverskom alatu. S konzole je potrebno učitati niz znakova T koji predstavljaju za koji se tip vozila provodi simulacija (auto, kombi, bus ili sve – prema sljedećoj tablici).

Tip vozila T	Donja granica dg [km/h]	Gornja granica gg [km/h]	Trajanje t [s]
auto	0	5	0,4
kombi	6	10	2
bus	10	20	4
sve	0	20	10

Nakon što se učitava tip vozila potrebno je napraviti simulaciju u trajanju određenom stupcem Trajanje t [s]. Pretpostavka je da jedna iteracija petlje (bilo koje) traje 2 ms (milisekunda).

Duljine unutar simulacije potrebno je upisivati u datoteku naziva duljine_tip_vozila.txt, gdje umjesto „tip_vozila“ (u imenu datoteke) treba upisati odgovarajući naziv tipa vozila T. Unutar svake iteracije potrebno je dohvatiti slučajnu duljinu (cijeli broj) unutar intervala [dg, gg] za odgovarajući tip vozila i zapisati ju u datoteku.

5. Napišite C# kod programa koji učitava sadržaj datoteke Udaljenost.txt u matricu $M = 20 \times 20$. U datoteci je zapisano 400 vrijednosti udaljenosti (cijeli brojevi), po 20 u svakom retku, međusobno odvojenim znakom – (crtica) kao što je prikazano na sljedećoj slici.



```

File Edit Format View Help
0-10-25-80-17-30-43-56-78-120-11-48-13-78-56-35-42-37-44-20
10-0-25-80-17-30-43-56-78-120-11-48-13-78-56-35-42-37-47-20
10-25-0-80-17-30-43-56-78-120-11-48-13-48-56-35-42-37-44-20
10-25-80-0-17-30-43-56-78-120-11-48-13-78-56-35-42-37-44-20
10-25-80-17-0-30-43-56-78-120-11-48-13-78-56-35-42-37-44-20

```

Način punjenja matrice je proizvoljan, ali je potrebno svih 400 vrijednosti zapisati u matricu.

Na kraju je potrebno ispisati ukupnu sumu udaljenosti u matrici u proizvoljnom formatu i napraviti ispis elemenata na glavnoj dijagonali matrice.



Ispit #47 (09.09.20. - grupa B)

1. Što će se ispisati u konzolu nakon izvršavanja sljedećeg programskog odsječka. Potrebno je naznačiti redoslijed ispisivanja. Kao točno rješenje priznaje se samo ono u kojem je vidljiv postupak i sadržaj varijabli.

```
DateTime t1 = DateTime.Now;
Thread.Sleep(500);
DateTime t2 = DateTime.Now;
Thread.Sleep(100);
TimeSpan ts = t2 - t1;
Console.WriteLine(ts.TotalSeconds);
Console.WriteLine((t2 - t1).TotalMilliseconds);
int dan = t1.Day;
int mjesec = t2.Month;
Console.WriteLine((dan - 2) % mjesec);
Console.WriteLine((dan - 2) / mjesec);
```

2. Što će se ispisati u konzolu nakon izvršavanja sljedećeg programskog odsječka. Potrebno je naznačiti redoslijed ispisivanja. Kao točno rješenje priznaje se samo ono u kojem je vidljiv postupak i sadržaj varijabli u svakoj iteraciji.

```
int[] p = new int[6];
for (int i = 1; i < p.Length; i++)
{
    p[i - 1] = i / 2;
}
foreach (int v in p)
{
    if (v < p.Length / 2)
    {
        Console.WriteLine(v);
    }
}
```

3. Napišite C# kod rekurzivne metode naziva Rekurzija koja kao argument prima cijeli broj n te ispisuje sve parne brojeve manje ili jednake n, uključujući i 0.



4. Napišite C# kod programa koji služi za simulaciju brzina u softverskom alatu. S konzole je potrebno učitati niz znakova P koji predstavljaju za koje se područje provodi simulacija (autocesta, ulica, avenija ili sve – prema sljedećoj tablici).

Područje P	Donja granica dg [km/h]	Gornja granica gg [km/h]	Trajanje t [s]
ulica	0	40	0,5
avenija	40	90	1
autocesta	90	150	2
sve	0	150	5

Nakon što se učitava područje potrebno je napraviti simulaciju u trajanju određenom stupcem Trajanje t [s]. Pretpostavka je da jedna iteracija petlje (bilo koje) traje 1 ms (milisekunda).

Brzine unutar simulacije potrebno je upisivati u datoteku naziva brzine_područje.txt, gdje se umjesto područje (u imenu datoteke) treba upisati odgovarajući naziv područja P. Unutar svake iteracije potrebno je dohvatiti slučajnu brzinu (cijeli broj) unutar intervala <dg, gg> za odgovarajuće područje i zapisati ju u datoteku.

5. Napišite C# kod programa koji učitava sadržaj datoteke udaljenosti.txt preko relativne putanje u matricu $M = 10 \times 10$. U datoteci je zapisano 100 udaljenosti (realni brojevi) svaka u zasebnom retku. Način punjenja matrice je proizvoljan, ali je potrebno svih 100 vrijednosti zapisati u matricu.

Na kraju je potrebno ispisati prosječnu vrijednost udaljenosti u matrici u formatu kao u primjeru i napraviti ispis matrice stupac po stupac (proizvoljan ispis pojedinih elemenata).

Primjer ispisa prosječne vrijednosti: „Prosječna vrijednost udaljenosti u matrici $M = 10 \times 10$ iznosi 9.87!!!“, pri čemu je 9.87 izračunata vrijednost i treba ju zamijeniti s vrijednošću odgovarajuće varijable u Vašem kodu.



Ispit #48 (21.11.20. - grupa B)

1. Što će se ispisati u konzolu nakon izvršavanja sljedećeg programskog odsječka, ukoliko se s konzole učitava niz „marKA“. Potrebno je naznačiti redoslijed ispisivanja. Kao točno rješenje priznaje se samo ono u kojem je vidljiv postupak i sadržaj varijabli u svakoj iteraciji.

```
string tekst = Console.ReadLine();
for (int i = tekst.Length - 2; i > 0; i--)
{
    switch (tekst[i])
    {
        case 'a':
            Console.WriteLine('r');
            break;
        case 'm':
            Console.WriteLine('k');
            break;
        case 'r':
            Console.WriteLine('a');
            break;
        case 'k':
            Console.WriteLine('m');
            break;
        default:
            Console.WriteLine('l');
            break;
    }
}
```

2. Napišite C# kod rekurzivne metode koja računa matematički izraz:

$$R(n) = R(n - 3) + R(n - 1) * R(n - 2), \forall n \geq 3$$

pri čemu je $R(0) = 1$, $R(1) = 1$, $R(2) = -1$.

Unutar Main metode potrebno je pozvati rekurzivnu metodu s argumentom $n = 6$ te ispisati rezultat.

3. Što će se ispisati u konzolu nakon izvršavanja sljedećeg programskog odsječka. Potrebno je naznačiti redoslijed ispisivanja. Kao točno rješenje priznaje se samo ono u kojem je vidljiv postupak i sadržaj varijabli u svakoj iteraciji.

```
List<int> lista = new List<int>();
lista.AddRange(new int[] { -1, 0, 1, 2 });
lista.RemoveAt(lista.Count - 3);
lista.Insert(lista.Count - 3, 2);
lista.Add(-1);
foreach (int v in lista)
{
    if (v != 1)
    {
        Console.WriteLine(v);
    }
}
```



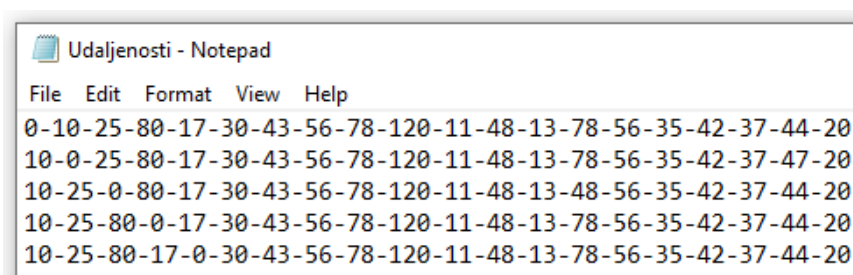
4. Napišite kod programa koji služi za simulaciju brzina u softverskom alatu. S konzole je potrebno učitati niz znakova P koji predstavljaju za koje se područje provodi simulacija (autocesta, ulica, avenija ili sve – prema sljedećoj tablici).

Područje P	Donja granica dg [km/h]	Gornja granica gg [km/h]	Trajanje t [s]
ulica	0	40	0,5
avenija	40	90	1
autocesta	90	150	2
sve	0	150	5

Nakon što se učita područje potrebno je napraviti simulaciju u trajanju određenom stupcem Trajanje t [s]. Pretpostavka je da jedna iteracija petlje (bilo koje) traje 1 ms (milisekunda).

Brzine unutar simulacije potrebno je upisivati u datoteku naziva brzine_područje.txt, gdje se umjesto područje (u imenu datoteke) treba upisati odgovarajući naziv područja P. Unutar svake iteracije potrebno je dohvatiti slučajnu brzinu (cijeli broj) unutar intervala <dg, gg> za odgovarajuće područje i zapisati ju u datoteku.

5. Napišite C# kod programa koji učitava sadržaj datoteke Udaljenosti.txt u matricu M = 20 x 20. U datoteci je zapisano 400 vrijednosti udaljenosti (cijeli brojevi), po 20 u svakom retku, međusobno odvojenih znakom – (crtica) kao što je prikazano na sljedećoj slici.



Način punjenja matrice je proizvoljan, ali je potrebno svih 400 vrijednosti zapisati u matricu.

Na kraju je potrebno ispisati ukupnu sumu udaljenosti u matrici u proizvoljnom formatu i napraviti ispis elemenata na glavnoj dijagonali matrice.



Ispit #49 (04.02.21. - grupa A)

1. Što će se ispisati u konzolu nakon izvršavanja sljedećeg programskog odsječka. Potrebno je naznačiti redoslijed ispisivanja. Kao točno rješenje priznaje se samo ono u kojem je vidljiv postupak i sadržaj varijabli u svakom pozivu metoda.

```
static void Main(string[] args)
{
    Console.WriteLine(metoda(8));
}
static int metoda(int x)
{
    if (x == 0)
    {
        return 1;
    }
    else if (x == 5)
    {
        return x * metoda(x - 5);
    }
    else if (x < 5)
    {
        return x - metoda(x + 1);
    }
    else if (x > 3)
    {
        return x + metoda(x / 2);
    }
    else
    {
        return metoda(x - 1);
    }
}
```

2. Što će se ispisati u konzolu nakon izvršavanja sljedećeg programskog odsječka. Potrebno je naznačiti redoslijed ispisivanja. Kao točno rješenje priznaje se samo ono u kojem je vidljiv postupak i sadržaj varijabli u svakoj iteraciji.

```
List<int> lista = new List<int>();
lista.Add(0);
lista.AddRange(new int[] { -2, 0, 3 });
lista.Remove(0);
lista.RemoveAt(lista.Count - 3);
for (int i = 0; i < lista.Count; i++)
{
    Console.WriteLine(lista[i]);
}
```



3. Napišite C# kod za simuliranje i obradu podataka o brzinama dobivenih s radara preko konzole. S konzole je potrebno učitati broj N koji predstavlja broj simuliranih brzina te popuniti jednodimenzionalno polje od N elemenata sa slučajnim vrijednostima unutar intervala $[0, 150]$.

Koristeći sortirano polje potrebno je implementirati filter koji će sve podatke koji nisu unutar intervala $[Q_1, Q_3]$ smatrati outlierima.

Indeksi kvartalnih granica u sortiranom polju računaju se kao $I_{Q_1} = 0.25 * N$ i $I_{Q_3} = 0.75 * N$, pri čemu je indeks potrebno zaokružiti na najbliži cijeli broj.

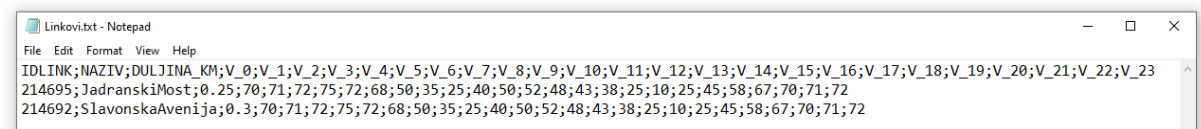
Na kraju je potrebno ispisati iznos dolje kvartalne granice Q_1 , gornje kvartalne granice Q_3 i broj outliera u sljedećem formatu: „Donja kvartalna granica iznosi Q_1 , a gornja kvartalna granica iznosi Q_3 ! Broj outliera: X “.

Deklaracija metode InsertionSort dana je u nastavku i nije ju potrebno implementirati već samo iskoristiti.

```
private static void InsertionSort(double[] polje)
```

4. Napišite C# kod za izračun vremena putovanja rute zadanog poljem ID-eva linkova (cestovnih segmenata) IDLinkovi i pripadajućih indeksa sata početka putovanja na linku zadanih poljem vrijemeIndeks, pri čemu indeks nula označava satni interval 00:00-01:00, a indeks 23 satni interval 23:00-00:00.

U datoteci Linkovi.txt prikazanoj na slici nalaze se svi linkovi na području Zagreba s podacima o ID-u linka (IDLINK), nazivu linka (NAZIV), duljini linka u kilometrima (DULJINA_KM) i brzini u km/h u određenom satnom intervalu (V_0 označava brzinu u satu 00:00-01:00, a V_{23} u satu 23:00-00:00).

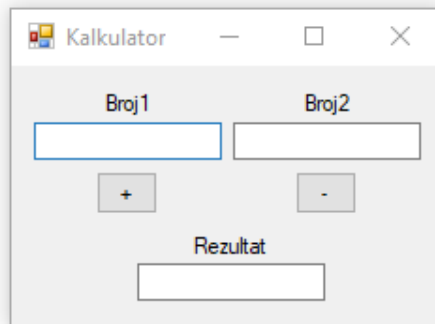


Potrebno je koristiti try-catch blok te nije potrebno dodati imenski prostor System.IO.

```
int[] IDLinkovi = new int[] { 214692, 214693, ..., 21717, 214695 };  
int[] vrijemeIndeks = new int[] { 0, 5, ..., 23, 10 };
```



5. Nadopunite C# kod Windows forme koja predstavlja kalkulator prikazan na slici koji radi zbrajanje i oduzimanje dvaju realnih brojeva.



Kontrole koje se koriste su: txtBroj1, txtBroj2, txtRezultat, labBroj1, labBroj2, labRezultat, btnPlus i btnMinus.

```
public partial class Form1 : Form
{
    public Form1()
    {
        InitializeComponent();
    }

    private void btnPlus_Click(object sender, EventArgs e)
    {
        //Nadopuni - 1
    }

    private void btnMinus_Click(object sender, EventArgs e)
    {
        //Nadopuni - 2
    }
}
```



Ispit #50 (18.02.21. - grupa A)

1. Što će se ispisati u konzolu nakon izvršavanja sljedećeg programskog odsječka. Potrebno je naznačiti redoslijed ispisivanja. Kao točno rješenje priznaje se samo ono u kojem je vidljiv postupak i sadržaj varijabli u svakom pozivu metoda.

```
static void Main(string[] args)
{
    Console.WriteLine(metoda(9));
}
static int metoda(int x)
{
    if (x == 0)
    {
        return 1;
    }
    else if (x == 5)
    {
        return x * metoda(x - 5);
    }
    else if (x < 5)
    {
        return x - metoda(x + 1);
    }
    else if (x > 3)
    {
        return x + metoda(x / 2);
    }
    else
    {
        return metoda(x - 1);
    }
}
```



2. Što će se ispisati u konzolu nakon izvršavanja sljedećeg programskog odsječka. Potrebno je naznačiti redoslijed ispisivanja. Kao točno rješenje priznaje se samo ono u kojem je vidljiv postupak i sadržaj varijabli u svakoj iteraciji.

```
int a = 3;
int b = 10;
int c, d;
if (b / 3 > 3)
{
    c = a % 2;
}
else
{
    c = b + 2 * a;
}
bool e = c % 4 == 0;
if (!(e ^ true))
{
    d = Convert.ToInt32(Math.Round(Math.Sqrt(c)));
}
else
{
    d = Convert.ToInt32(Math.Round(Math.Pow(c, 2)));
}
Console.WriteLine(a + "," + b + "," + c + "," + d);
```



3. Napišite C# kod za simuliranje i obradu podataka o brzinama dobivenih s radara preko konzole. S konzole je potrebno učitati broj N koji predstavlja broj simuliranih brzina te popuniti jednodimenzionalno polje od N elemenata sa slučajnim vrijednostima unutar intervala $[0, 200]$.

Koristeći sortirano polje potrebno je implementirati filter koji će sve podatke koji nisu unutar intervala $[Q_1, Q_3]$ smatrati outlierima.

Indeksi granica u sortiranom polju računaju se kao $I_{Q_1} = 0.15 * N$ i $I_{Q_3} = 0.85 * N$, pri čemu je indeks potrebno zaokružiti na najbliži cijeli broj.

Na kraju je potrebno ispisati iznos dolje granice Q_1 , gornje granice Q_3 i broj outliera u sljedećem formatu: „Donja granica iznosi Q_1 , a gornja granica iznosi Q_3 ! Broj outliera: X “.

Deklaracija metode BubbleSort dana je u nastavku i nije ju potrebno implementirati već samo iskoristiti.

```
private static void BubbleSort(double[] polje)
```

4. Napišite C# kod za učitavanje ID-a linka i indeksa sata s tipkovnice sve dok se za ID linka ne učitava broj manji ili jednak nuli.

U programu je već inicijalizirana lista linkovi s instancama klase LINK, te listu nije potrebno popuniti već samo iskoristiti. Klasa LINK dostupna je u nastavku, te sadrži attribute ID, NAZIV, DULJINA_KM, te polje BRZINE_U_SATU koje sadrži 24 zapisa brzina u km/h, pri čemu indeks nula (0) označava satni interval 00:00 – 01:00, a indeks 23 satni interval 23:00 – 00:00.

Za svaki učitani ID linka potrebno je pronaći odgovarajući instancu klase LINK u listi linkovi i izračunati vrijeme putovanja na linku ovisno o učitanoj satnoj indeksu.

Dodatno je u datoteku linkoviRute.txt koja se treba nalaziti na putanji D:\Podaci za svaki učitani ID linka potrebno zapisati attribute linka u formatu:

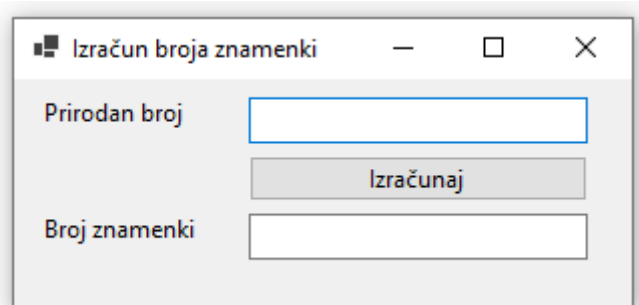
ID;NAZIV;DULJINA_KM;SATNI_INDEKS;VRIJEME_PUTOVANJA_U_SATNOM_INDEKSU

Potrebno je koristiti try-catch blok te nije potrebno dodati imenski prostor System.IO.

```
class LINK
{
    public int ID;
    public string NAZIV;
    public int DULJINA_KM;
    public double[] BRZINE_U_SATU;
}
static void Main(string[] args)
{
    List<LINK> linkovi = inicijalizirana lista linkova;
}
```



5. Nadopunite C# kod Windows forme na slici koja izračunava koliko znamenaka ima uneseni prirodni broj.

The image shows a Windows application window titled "Izračun broja znamenki". Inside the window, there are two text input fields. The first field is labeled "Prirodan broj" and is empty. Below it is a button labeled "Izračunaj". Below the button is a second text input field labeled "Broj znamenki", which is also empty. The window has standard Windows window controls (minimize, maximize, close) in the top right corner.

Kontrole koje se koriste su: txtBroj (TextBox), txtBrojZnamenki (TextBox), labBroj (Label), labBrojZnamenki (Label) i btnIzracunaj (Button). Potrebno je napisati samo dijelove koda ispod komentara „Nadopuni“.

```
public Form1()
{
    InitializeComponent();
}
private void btnIzracunaj_Click(object sender, EventArgs e)
{
    //Nadopuni

}
```



Ispit #51 (17.04.21. - grupa A)

Napomena: ispit je u potpunosti isti kao i kolokvij koji je pisan 13.04.2021. (grupa A).

1. Što će se ispisati u konzolu nakon izvršavanja sljedećeg programskog odsječka, ukoliko se s konzole učitava tekst ALGPRO. Potrebno je naznačiti redoslijed ispisivanja. Kao točno rješenje priznaje se samo ono u kojem je vidljiv postupak i sadržaj varijabli u svakoj iteraciji.

```
static void Main(string[] args)
{
    string unos = Console.ReadLine();
    Console.WriteLine(metoda(unos));
}
static int metoda(string x)
{
    for (int i = x.Length; i >= 0; i--)
    {
        if (i % 2 == 1)
        {
            Console.Write(x[i]);
        }
    }
    Console.WriteLine();
    return x.Length;
}
```



2. Što će se ispisati u konzolu nakon izvršavanja sljedećeg programskog odsječka, ukoliko su s konzole učitane vrijednosti 7 i 3. Potrebno je naznačiti redoslijed ispisivanja. Kao točno rješenje priznaje se samo ono u kojem je vidljiv postupak i sadržaj varijabli u svakom pozivu metoda.

```
static void Main(string[] args)
{
    int x = Convert.ToInt32(Console.ReadLine());
    int y = Convert.ToInt32(Console.ReadLine());
    if (x / y > 2)
    {
        metoda(x, y);
    }
    else
    {
        metoda(y, x);
    }
    Console.WriteLine(x + " , " + y);
}
static void metoda(double x, int y)
{
    double z = x / y;
    switch (z)
    {
        case 2.0:
            Console.Write("-", " ");
            break;
        case 2.5:
            Console.Write("X, ");
            break;
        default:
            Console.Write("=", " ");
            break;
    }
    x = y++;
    Console.Write((x + y) + ",");
}
```

3. Napišite C# kod za izračun udjela osobnih i teretnih vozila. S konzole je potrebno učitati broj N koji predstavlja ukupan broj vozila, te potom učitati N cjelobrojnih vrijednosti s konzole. Ukoliko je unesena vrijednost 1 radi se o osobnom automobilu, a ako je unesena vrijednost 2 radi se o teretnom vozilu (nije potrebna provjera da li je unesena ispravna vrijednost).

Ispis udjela osobnih i teretnih vozila potrebno je napraviti u zasebnoj metodi Ispis koja kao argumente prima ukupan broj osobnih i ukupan broj teretnih vozila.

Ispis mora biti u formatu kao u primjeru: „Udio osobnih vozila iznosi: 42.7123645%, a udio teretnih vozila iznosi 57.2876355%!“.



4. Napišite C# kod za izračun udaljenosti između 100 telekomunikacijskih baznih stanica, čije su koordinate upisane u datoteci bazneStanice.txt. U svakom retku datoteke upisane su x i y koordinate (realni brojevi) bazne stanice u formatu x;y (ukupno ima 100 redaka).

Podatke je potrebno spremati u zasebno polje.

Nakon učitavanja podataka potrebno je kreirati i ispisati 100 x 100 matricu udaljenosti između baznih stanica. Format ispisa je proizvoljan.

Za izračun udaljenosti potrebno je koristiti euklidsku udaljenost zadanu jednačbom:

$$d(T_1(x_1, y_1), T_2(x_2, y_2)) = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

5. Napišite C# kod za učitavanje podataka o vozilu na tehničkom pregledu. Podatke o vozilima potrebno je učitavati s konzole u instance priložene klase VOZILO.

```
class VOZILO
{
    public int ID;
    public string REGISTRACIJA;
    public int GODINA_PROIZVODNJE;
    public string MODEL;
    public bool PROLAZAK_NA_PREGLEDU;
}
```

Vozila je potrebno učitavati sve dok se za ID vozila ne unese vrijednost 0. Sva unesena vozila potrebno je spremiti u listu vozila.

Nakon unesenih vozila potrebno je u proizvoljnom formatu ispisati registarske tablice svih vozila koja nisu prošla tehnički pregled.



Napomena: ispit je u potpunosti isti kao i kolokvij koji je pisan 13.04.2021. (grupa B).

1. Što će se ispisati u konzolu nakon izvršavanja sljedećeg programskog odsječka. Kao točno rješenje priznaje se samo ono u kojem je vidljiv postupak i sadržaj varijabli u svakoj iteraciji. Potrebno je naznačiti redoslijed ispisivanja.

```
static void Main(string[] args)
{
    int[,] matrica = { { 1, 2, 3 }, { 3, 4, 2 }, { 5, 6, 7 } };

    for (int i = -2; i <= matrica.GetLength(0) - 3; i++)
    {
        for (int j = matrica.GetLength(1) + 1; j > 0; j--)
        {
            if (i + 2 == j - 1)
            {
                Console.WriteLine(matrica[i + 2, j - 1] + " ");
            }
        }
        Console.WriteLine(matrica.Length);
    }
}
```

2. Što će se ispisati u konzolu nakon izvršavanja sljedećeg programskog odsječka. Kao točno rješenje priznaje se samo ono u kojem je vidljiv postupak i sadržaj varijabli u svakoj iteraciji.

```
static void Main(string[] args)
{
    int a = 7 % 7;
    int b = 5 + a;
    int c = 0, d = 1;
    while (a + b != 1)
    {
        if (a >= 2 || (a != 3 ^ d == 1))
        {
            c = a++;
            d = 2 * 10;
            b -= b;
        }
        else
        {
            c += a;
            d = a * 3 + 2;
        }
    }
    Console.WriteLine(a + " , " + b + " , " + c + " , " + d);
}
```



3. Napišite C# kod koji učitava realne brojeve za parametre šesterostrane piramide: h – visina i a – duljina brida baze. Parametre je potrebno učitavati s konzole sve dok su oni valjani, te u svakom koraku izračunati i ispisati volumen šesterostrane piramide prema sljedećem izrazu:

$$V = \frac{a^2\sqrt{3}}{2}h$$

Format ispisa mora biti sljedeći: „Volumen pravilne šesterostrane piramide visine h i duljine brida baze a iznosi V “, pri čemu h , a i V treba zamijeniti odgovarajućim vrijednostima.

Volumen V potrebno je zaokružiti na dva decimalna mjesta.

Provjeru da li su učitane vrijednosti valjane potrebno je napraviti u zasebnoj metodi Provjera koja prima dva argumenta h i a . Vrijednosti su valjane ukoliko su ova argumenta veća od nule.

4. Napišite C# kod programa koji učitava 1000 nizova znakova koji predstavljaju brojeve i provjerava je li upisani niz znakova OIB ili JMBAG (pretpostavka je da se učitavaju samo nizovi znakova koji predstavljaju brojeve).

Kada se program pokrene u konzoli treba ispisati poruku „Upišite broj za provjeru“. Ukoliko korisnik upiše OIB, program treba ispisati poruku „Upisali ste OIB!“, ukoliko upiše JMBAG treba ispisati poruku „Upisali ste JMBAG!“, a ako upisani broj nije niti JMBAG niti OIB onda program treba korisniku ispisati poruku „Upisani broj nije niti OIB niti JMBAG!“.

Provjeru broja treba izvršiti pomoću metode ProvjeraZnamenki koja vraća broj znamenki upisanog broja (niz znakova).

Dodatno, u listu osobnibrojevi u svakom koraku je potrebno spremati valjane OIB-e i JMBAG-ove. Na kraju je potrebno ispisati sve valjane OIB-e i JMBAG-ove, te njihov broj (koliko ih ima) u proizvoljnom formatu.

Napomena: JMBAG ima 10 znamenki, a OIB ima 11 znamenki (broj 0 može biti na početku).

5. Napišite C# kod programa koji učitava podatke o N vozila i njihovim atributima: marka, boja i starost vozila (cijeli broj). Broj vozila N potrebno je učitati s konzole.

Potrebno je napisati klasu Vozilo koja ima javne varijable marka, boja i starost, te konstruktor klase koji kao argument prima sve podatke (marka, boja i starost) za kreiranje instance klase.

U glavnom dijelu programa trebaju se učitavati podaci o N vozila s konzole i spremati u jednodimenzionalno polje naziva vozniPark.



Napomena: ispit je u potpunosti isti kao i kolokvij koji je pisan 13.04.2021. (grupa C).

1. Što će se ispisati u konzolu nakon izvršavanja sljedećeg programskog odsječka. Kao točno rješenje priznaje se samo ono u kojem je vidljiv postupak i sadržaj varijabli u svakom koraku.

```
static void Main(string[] args)
{
    int[] a = new int[3];
    a[0] = 15;
    a[a.Length - 1] = 314;
    a[a.Length / 2] = 42;
    int i;
    for (i = -1; i < a.Length - 1; i++)
    {
        Metoda(a);
        Console.WriteLine(a[i + 1]);
        if (a[i + 1] > 50)
        {
            break;
        }
    }
}
static void Metoda(int[] b)
{
    b[0] = b[0] + b[1];
    b[1] = 73;
    b[2] = 314;
}
```

2. Što će se ispisati u konzolu nakon izvršavanja sljedećeg programskog odsječka. Potrebno je naznačiti redoslijed ispisivanja. Kao točno rješenje priznaje se samo ono u kojem je vidljiv postupak i sadržaj varijabli u svakoj iteraciji.

```
static void Main(string[] args)
{
    int i, j;
    double x;
    i = 15 % 3 + 2;
    j = i++ * 3 / 4;
    x = j * 5.0 / 2;
    if (i == j + 1)
    {
        Console.WriteLine(i + "." + j + "." + x + "." + (i + 1));
    }
    else if (i - 2 == j)
    {
        Console.WriteLine(i + "." + j + "." + x + "." + i + 1);
    }
    else
    {
        Console.WriteLine(i + "." + j + "." + x + "." + (i + j));
    }
}
```



3. Napišite C# kod za izračun prosječne ocjene i udjela (postotka) ocjena na kolegiju Algoritmi i programiranje. S konzole je potrebno učitavati ime, prezime i ocjenu studenata sve dok se za ocjenu ne unese vrijednost 0.

Ispis prosječne ocjene na kolegiju i udjela (postotka) svake ocjene potrebno je napraviti u zasebnoj metodi Ispis koja prima 5 argumenata: za svaku ocjenu od 1 do 5 broj pojavljivanja te ocjene.

Ispis mora biti u formatu kao u primjeru: „Prosječna ocjena na kolegiju iznosi 2.692307692! Udio ocjena 1, 2, 3, 4 i 5 redom iznosi: 19.23%, 23.07%, 38.46%, 7.69% i 11.55%!“.

4. Napišite C# kod za izračun težinskog prosjeka ocjena na temelju ocjene predmeta (cijeli broj) i ECTS bodova (cijeli broj) koliko taj predmet nosi.

S konzole je potrebno učitati broj N koji predstavlja ukupan broj zapisa parova ocjena i ECTS bodova.

Potom je s konzole potrebno učitavati ocijene i ECTS bodove te ih spremiti u dvodimenzionalno polje naziva vrijednosti.

Ispit težinskog prosjeka ocjena potrebno je napraviti u zasebnoj metodi IzracunProsjeka kojoj će se proslijediti polje vrijednosti, te koja će izračunati težinski prosjek ocjena koristeći sljedeću jednadžbu:

$$\text{težinski prosjek} = \frac{\sum_i^N (ECTS_i * ocjena_i)}{\sum ECTS}$$

Format ispisat je proizvoljan.

5. Napišite C# kod za učitavanje podataka o stavkama računa (na jednom računu nalazi se više stavaka). Podatke o stavkama računa potrebno je učitavati u instance priložene klase STAVKA.

Stavke je potrebno učitavati sve dok se za ID stavke ne unese vrijednost 0.

Učitane stavke potrebno je spremiti u listu naziva stavke.

Nakon učitavanja i unošenja stavki potrebno je u proizvoljnom formatu ispisati ukupnu cijenu računa za koju su učitavane stavke.

```
class STAVKA
{
    public int ID_STAVKE;
    public string PROIZVID;
    public double CIJENA;
    public double POPUST;
}
static void Main(string[] args)
{
    //Nadopuni
}
```



Ispit #54 (10.06.21. - grupa A)

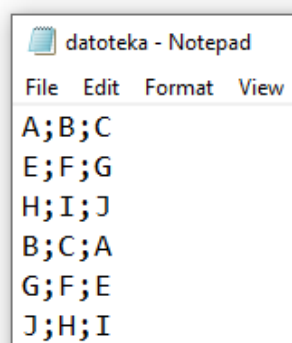
1. Što će se ispisati u konzolu nakon izvršavanja sljedećeg programskog odsječka. Potrebno je naznačiti redoslijed ispisivanja. Kao točno rješenje priznaje se samo ono u kojem je vidljiv postupak i sadržaj varijabli u svakom pozivu metoda.

Napomena: ne treba biti predetalan oko opisivanja načina na koji se ispisuju podaci, dovoljno je to napraviti jednom.

```
static void Main(string[] args)
{
    int[] p = new int[6] { 1, -1, 2, -2, -3, 3 };
    ispisPolja(0, p);
    izracunaj(p);
    ispisPolja(2, p);
    p = new int[6];
    ispisPolja(0, p);
}
private static void ispisPolja(int a, int[] p)
{
    for (int i = a; i < p.Length; i++)
    {
        Console.Write(p[i] + " ");
    }
    Console.WriteLine();
}
private static void izracunaj(int[] p)
{
    for (int i = 1; i < p.Length; i++)
    {
        if (i == p.Length)
            break;
        p[i - 1] = p[i - 1] * p[i];
    }
    ispisPolja(3, p);
}
```



2. Što će se ispisati u konzolu nakon izvršavanja sljedećeg programskog odsječka, ukoliko se u datoteci *datoteka.txt* smještenoj na relativnoj putanji nalaze vrijednosti prikazane na slici.



Potrebno je naznačiti redoslijed ispisivanja. Kao točno rješenje priznaje se samo ono u kojem je vidljiv postupak i sadržaj varijabli u svakom koraku.

```
static void Main(string[] args)
{
    try
    {
        StreamReader citanje = new StreamReader("datoteka.txt");
        int i = 2;
        while (citanje.EndOfStream == false)
        {
            string var = citanje.ReadLine();
            string[] dio = var.Split(';');
            var = citanje.ReadLine();
            dio = var.Split(';');
            Console.WriteLine(dio[i]);
            i--;
        }
        citanje.Close();
        string var2 = citanje.ReadLine();
        string[] dio2 = var2.Split(';');
        Console.WriteLine(dio2[0]);
    }
    catch (Exception ex)
    {
        Console.WriteLine("M");
    }
}
```



3. Napišite C# kod dvaju klasa *Lokacija* i *Korisnik*.

Klasa *Lokacija* treba imati dvije javne varijable koje predstavljaju x i y koordinate u koordinatnom sustavu (brojevi s decimalnim zarezom), te klasa ne smije imati poseban konstruktor.

Klasa *Korisnik* treba imati tri varijable: javnu varijablu ID korisnika (cijeli broj), javnu varijablu za instancu klase *Lokacija* koja će predstavljati mjesto gdje se nalazi korisnik i privatnu varijablu za spremanje pozicije korisnika u listi svih korisnika, odnosno na kojemu mjestu u listi svih korisnika se nalazi pojedina instanca klase *Korisnik*.

Konstruktor klase *Korisnik* treba primiti i spremiti vrijednost sve tri varijable.

Dodatno, klasa *Korisnik* treba imati dvije metode: metodu *dohvatiPozicijuUListi* koja će vratiti vrijednost pozicije korisnika u listi, te metodu *postaviVrijednostPozicije* koja postavlja vrijednost varijable pozicije u listi na temelju vrijednosti koja se predaje kao argument metodi.

4. Napišite C# kod metode *pronadiNajblizeg* koja prima dva argumenta: instancu klase *Lokacija* i listu svih korisnika (lista instanci klase *Korisnik*).

Metoda treba vratiti onog korisnika (njegovu instancu) koji se među listom korisnika nalazi najbliže predanom argumentu lokacije.

Za izračun udaljenosti koristiti euklidsku udaljenost danu jednadžbom:

$$d(T_1(x_1, y_1), T_2(x_2, y_2)) = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

Klase *Korisnik* i *Lokacija* odnose se na klase napravljene u 3. zadatku.

5. Napišite C# kod za učitavanje podataka o korisnicima u listu korisnika (lista instanci klase *Korisnik*) sve dok se za ID korisnika ne unese vrijednost manja od 0.

Prilikom učitavanja potrebno je unijeti sve podatke o korisniku, stvoriti instancu klase *Korisnik* iz zadatka 3 i dodati ga u listu.

Nakon učitavanja potrebno je simulirati rješavanje problema trgovačkog putnika na način da se u konzolu redom ispisuju ID-evi korisnika koji bi se dodali u vozilo. Početna lokacija od kuda počinje vozilo ima koordinate $(x, y) = (0, 0)$, te je u svakom koraku sve dok se ne obrade svi korisnici (dok lista korisnika nije prazna) potrebno:

- korištenjem metode *pronadiNajblizeg* (zadatak 4) pronaći korisnika u listi korisnika koji je najbliži trenutnoj lokaciji (u početku je trenutna lokacija zadana početnim vrijednostima $(x, y) = (0, 0)$);
- ispisati u konzolu poruku u formatu *U vozilo se dodaje korisnika po ID-em x!*, pri čemu x predstavlja vrijednost ID-a korisnika;
- ažurirati vrijednosti trenutne lokacije s vrijednostima lokacije dohvaćenog korisnika;
- na temelju vrijednosti varijable pozicije u listi korisnika, izbaciti iz liste svih korisnika prethodno dohvaćenog korisnika.



Ispit #55 (10.06.21. - grupa B)

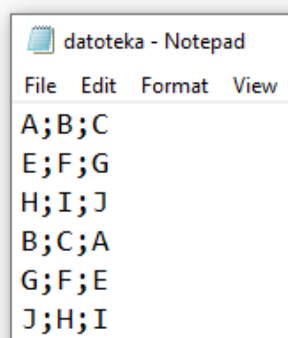
1. Što će se ispisati u konzolu nakon izvršavanja sljedećeg programskog odsječka. Potrebno je naznačiti redoslijed ispisivanja. Kao točno rješenje priznaje se samo ono u kojem je vidljiv postupak i sadržaj varijabli u svakom pozivu metoda.

Napomena: ne treba biti predetalan oko opisivanja načina na koji se ispisuju podaci, dovoljno je to napraviti jednom.

```
static void Main(string[] args)
{
    int[] p = new int[6] { -3, 3, -2, 2, -1, 1 };
    ispisPolja(0, p);
    izracunaj(p);
    ispisPolja(1, p);
    p = new int[5];
    ispisPolja(0, p);
}
private static void ispisPolja(int a, int[] p)
{
    for (int i = p.Length - 1; i >= a; i--)
    {
        Console.Write(p[i] + " ");
    }
    Console.WriteLine();
}
private static void izracunaj(int[] p)
{
    for (int i = p.Length - 1; i > 1; i--)
    {
        if (i == 0)
            break;
        p[i - 1] = p[i - 1] * p[i];
    }
    ispisPolja(2, p);
}
```



2. Što će se ispisati u konzolu nakon izvršavanja sljedećeg programskog odsječka, ukoliko se u datoteci *datoteka.txt* smještenoj na relativnoj putanji nalaze vrijednosti prikazane na slici.



Potrebno je naznačiti redoslijed ispisivanja. Kao točno rješenje priznaje se samo ono u kojem je vidljiv postupak i sadržaj varijabli u svakom koraku.

```
static void Main(string[] args)
{
    try
    {
        StreamReader citanje = new StreamReader("datoteka.txt");
        int i = 2;
        string var = "";
        string[] dio;
        while (citanje.EndOfStream == false)
        {
            var = citanje.ReadLine();
            var = citanje.ReadLine();
            dio = var.Split(';');
            Console.WriteLine(dio[i]);
            i--;
        }
        dio = var.Split(';');
        Console.WriteLine(dio[dio.Length - 1]);
        citanje.Close();
    }
    catch (Exception ex)
    {
        Console.WriteLine("M");
    }
}
```



3. Napišite C# kod dvaju klasa *Lokacija* i *Korisnik*.

Klasa *Lokacija* treba imati dvije privatne varijable koje predstavljaju x i y koordinate u koordinatnom sustavu (brojevi s decimalnim zarezom), te klasa treba imati konstruktor koji treba primiti i spremiti vrijednost obje varijable.

Dodatno, klasa *Korisnik* treba imati četiri metode: metodu *dohvatiX* koja će vratiti vrijednost x varijable klase, metodu *dohvatiY* koja će vratiti vrijednost y varijable klase, metodu *postaviX* koja će postaviti vrijednost x varijable klase na temelju vrijednosti koja se predaje kao argument metodi, te metodu *postaviY* koja će postaviti vrijednost y varijable klase na temelju vrijednosti koja se predaje kao argument metodi.

Klasa *Korisnik* treba imati tri varijable: javnu varijablu ID korisnika (cijeli broj), javnu varijablu za instancu klase *Lokacija* koja će predstavljati mjesto gdje se nalazi korisnik i javnu varijablu za spremanje pozicije korisnika u listi svih korisnika, odnosno na kojemu mjestu u listi svih korisnika se nalazi pojedina instanca klase *Korisnik*.

4. Napišite C# kod metode *pronadiNajblizeg* koja prima dva argumenta: instancu klase *Lokacija* i listu svih korisnika (lista instanci klase *Korisnik*).

Metoda treba vratiti onog korisnika (njegovu instancu) koji se među listom korisnika nalazi najbliže predanom argumentu lokacije.

Za izračun udaljenosti koristiti euklidsku udaljenost danu jednadžbom:

$$d(T_1(x_1, y_1), T_2(x_2, y_2)) = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

Klase *Korisnik* i *Lokacija* odnose se na klase napravljene u 3. zadatku.

5. Napišite C# kod za učitavanje podataka o korisnicima u listu korisnika (lista instanci klase *Korisnik*) sve dok se za ID korisnika ne unese vrijednost manja od 0.

Prilikom učitavanja potrebno je unijeti sve podatke o korisniku, stvoriti instancu klase *Korisnik* iz zadatka 3 i dodati ga u listu.

Nakon učitavanja potrebno je simulirati rješavanje problema trgovačkog putnika na način da se u konzolu redom ispisuju ID-evi korisnika koji bi se dodali u vozilo. Početna lokacija od kuda počinje vozilo ima koordinate $(x, y) = (0, 0)$, te je u svakom koraku sve dok se ne obrade svi korisnici (dok lista korisnika nije prazna) potrebno:

- korištenjem metode *pronadiNajblizeg* (zadatak 4) pronaći korisnika u listi korisnika koji je najbliži trenutnoj lokaciji (u početku je trenutna lokacija zadana početnim vrijednostima $(x, y) = (0, 0)$);
- ispisati u konzolu poruku u formatu *U vozilo se dodaje korisnika po ID-em x!*, pri čemu x predstavlja vrijednost ID-a korisnika;
- ažurirati vrijednosti trenutne lokacije s vrijednostima lokacije dohvaćenog korisnika;
- na temelju vrijednosti varijable pozicije u listi korisnika, izbaciti iz liste svih korisnika prethodno dohvaćenog korisnika.



Ispit #56 (10.06.21. - grupa C)

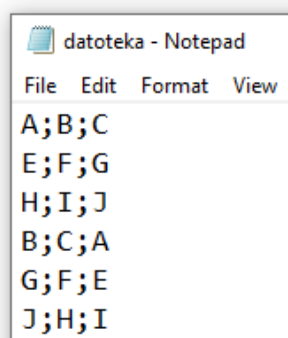
1. Što će se ispisati u konzolu nakon izvršavanja sljedećeg programskog odsječka. Potrebno je naznačiti redoslijed ispisivanja. Kao točno rješenje priznaje se samo ono u kojem je vidljiv postupak i sadržaj varijabli u svakom pozivu metoda.

Napomena: ne treba biti predetalan oko opisivanja načina na koji se ispisuju podaci, dovoljno je to napraviti jednom.

```
static void Main(string[] args)
{
    int[] p = new int[6] { 3, -1, 2, -3, 1, -2 };
    ispisPolja(0, p);
    izracunaj(p);
    ispisPolja(1, p);
    p = new int[4];
    ispisPolja(0, p);
}
private static void ispisPolja(int a, int[] p)
{
    for (int i = a; i < p.Length; i++)
    {
        Console.Write(p[i] + " ");
    }
    Console.WriteLine();
}
private static void izracunaj(int[] p)
{
    for (int i = p.Length - 1; i > 1; i--)
    {
        if (i == 0)
            break;
        p[i] = p[i - 1] * p[i];
    }
    ispisPolja(2, p);
}
```



2. Što će se ispisati u konzolu nakon izvršavanja sljedećeg programskog odsječka, ukoliko se u datoteci *datoteka.txt* smještenoj na relativnoj putanji nalaze vrijednosti prikazane na slici.



Potrebno je naznačiti redoslijed ispisivanja. Kao točno rješenje priznaje se samo ono u kojem je vidljiv postupak i sadržaj varijabli u svakom koraku.

```
static void Main(string[] args)
{
    try
    {
        StreamReader citanje = new StreamReader("datoteka.txt");
        int i = 0;
        while (citanje.EndOfStream == false)
        {
            string var = citanje.ReadLine();
            var = citanje.ReadLine();
            string[] dio = var.Split(';');
            dio = var.Split(';');
            Console.WriteLine(dio[i]);
            i++;
        }
        citanje.Close();
        string var2 = citanje.ReadLine();
        string[] dio2 = var2.Split(';');
        Console.WriteLine(dio2[2]);
    }
    catch (Exception ex)
    {
        Console.WriteLine("M");
    }
}
```



3. Napišite C# kod dvaju klasa *Lokacija* i *Korisnik*.

Klasa *Lokacija* treba imati dvije javne varijable koje predstavljaju x i y koordinate u koordinatnom sustavu (brojevi s decimalnim zarezom), te klasa treba imati konstruktor koji treba primiti i spremiti vrijednosti obje varijable.

Klasa *Korisnik* treba imati tri varijable: privatnu varijablu ID korisnika (cijeli broj), javnu varijablu za instancu klase *Lokacija* koja će predstavljati mjesto gdje se nalazi korisnik i javnu varijablu za spremanje pozicije korisnika u listi svih korisnika, odnosno na kojemu mjestu u listi svih korisnika se nalazi pojedina instanca klase *Korisnik*.

Klasa *Korisnik* ne treba imati konstruktor.

Dodatno, klasa *Korisnik* treba imati dvije metode: metodu *dohvatiID* koja će vratiti vrijednost ID-a korisnika u listi, te metodu *postaviID* koja postavlja vrijednost ID-a korisnika na temelju vrijednosti koje se predaju kao argument metodi.

4. Napišite C# kod metode *pronadiNajblizeg* koja prima dva argumenta: instancu klase *Lokacija* i listu svih korisnika (lista instanci klase *Korisnik*).

Metoda treba vratiti onog korisnika (njegovu instancu) koji se među listom korisnika nalazi najbliže predanom argumentu lokacije.

Za izračun udaljenosti koristiti euklidsku udaljenost danu jednadžbom:

$$d(T_1(x_1, y_1), T_2(x_2, y_2)) = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

Klase *Korisnik* i *Lokacija* odnose se na klase napravljene u 3. zadatku.

5. Napišite C# kod za učitavanje podataka o korisnicima u listu korisnika (lista instanci klase *Korisnik*) sve dok se za ID korisnika ne unese vrijednost manja od 0.

Prilikom učitavanja potrebno je unijeti sve podatke o korisniku, stvoriti instancu klase *Korisnik* iz zadatka 3 i dodati ga u listu.

Nakon učitavanja potrebno je simulirati rješavanje problema trgovačkog putnika na način da se u konzolu redom ispisuju ID-evi korisnika koji bi se dodali u vozilo. Početna lokacija od kuda počinje vozilo ima koordinate $(x, y) = (0, 0)$, te je u svakom koraku sve dok se ne obrade svi korisnici (dok lista korisnika nije prazna) potrebno:

- korištenjem metode *pronadiNajblizeg* (zadatak 4) pronaći korisnika u listi korisnika koji je najbliži trenutnoj lokaciji (u početku je trenutna lokacija zadana početnim vrijednostima $(x, y) = (0, 0)$);
- ispisati u konzolu poruku u formatu *U vozilo se dodaje korisnika po ID-em x!*, pri čemu x predstavlja vrijednost ID-a korisnika;
- ažurirati vrijednosti trenutne lokacije s vrijednostima lokacije dohvaćenog korisnika;
- na temelju vrijednosti varijable pozicije u listi korisnika, izbaciti iz liste svih korisnika prethodno dohvaćenog korisnika.



Ispit #57 (28.06.21. - grupa A)

1. Napišite C# kod metode *ZbrojiElemente* koja kao argument prima 3x3 matricu cijelih brojeva.

Metoda u konzolu treba ispisati zbroj elemenata trećeg stupca predane matrice u proizvoljnom formatu.

U *Main* metodi potrebno je kreirati matricu, popuniti ju s vrijednostima učitanim s konzole i pozvati metodu *ZbrojiElemente*.

2. Napišite C# kod konzolne aplikacije koja omogućuje korisniku zapisivanje n proizvoljnih korisničkih imena i pripadajućih zaporki u datoteku pod nazivom *login.txt*, s tim da je zaporku potrebno enkriptirati *Rot13* algoritmom čija je deklaracija metode dana u nastavku:

```
private static string Rot13(string var)
```

Vrijednost n unosi se s konzole, a datoteku *login.txt* potrebno je kreirati u direktoriju D:\Prijava.

Korisničko ime i pripadajuća zaporka trebaju u datoteci biti zapisani u zasebnom retku u formatu „korisničkoIme;lozinka“.

Napomena: nije potrebno implementirati algoritam *Rot13*, već samo pozvati i iskoristiti odgovarajuću metodu. Potrebno je koristiti try-catch blok, te nije potrebno dodati imenski prostor System.IO.

3. Napišite C# kod programa koji u polje sprema 100 slučajnih cijelih brojeva unutar intervala [*DonjaGranica*, *GornjaGranica*], pri čemu se *DonjaGranica* i *GornjaGranica* unose s konzole.

Ukoliko korisnik upiše vrijednost *DonjaGranica* veću od *GornjaGranica* program treba zamijeniti vrijednost varijabla *DonjaGranica* i *GornjaGranica*.

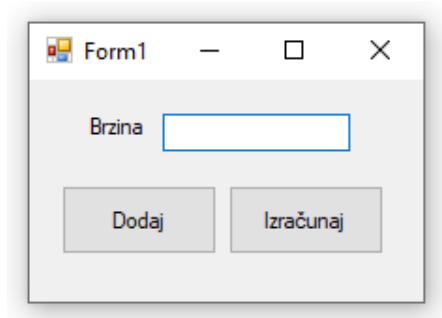
Na kraju program treba izračunati prosječnu vrijednost elemenata u polju, uzimajući u obzir svaki drugi element u polju, a ne sve elemente polja. Format ispisa prosječne vrijednosti je proizvoljan.



4. Nadopišite C# kod na način da se klikom na gumb *Dodaj* (*btnDodaj*) u listu brzina doda brzina automobila (broj s plivajućim zarezom) upisana u *TextBox txtBrzina*, a klikom na gumb *Izračunaj* (*btnIzracunaj*) izračunava se i ispisuje vrijednost najveće brzine u listi brzina u skočnom prozoru (*MessageBox*) u formatu *Najveća brzina iznosi x km/h!*, pri čemu varijabla *x* predstavlja vrijednost najveće brzine.

Vrijednost najveće brzine mora biti zaokružena na dvije decimale.

Sučelje je prikazano slikom:



Potrebno je napisati samo dijelove koda unutar *Nadopuni*.

```
public partial class Form1 : Form
{
    //Nadopuni - 1

    public Form1()
    {
        InitializeComponent();
    }

    private void btnDodaj_Click(object sender, EventArgs e)
    {
        //Nadopuni - 2
    }

    private void btnIzracunaj_Click(object sender, EventArgs e)
    {
        //Nadopuni - 3
    }
}
```

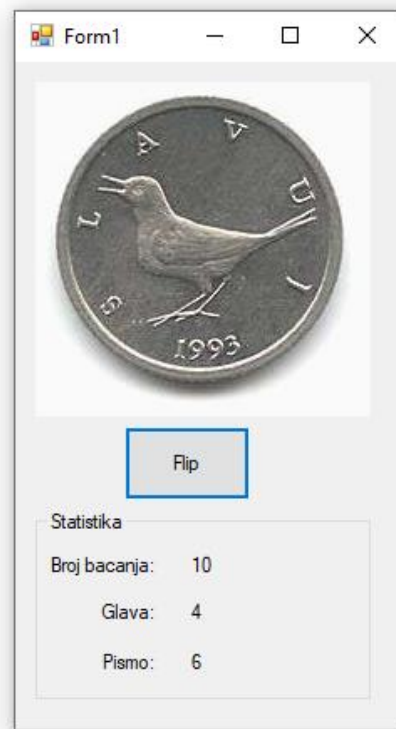
5. Napišite C# kod konzolne aplikacije koja simulira *n* bacanja kovanice, pri čemu se *n* učitava s tipkovnice.

U konzolu je potrebno ispisati vrijednost pojave glave i vjerojatnost pojave pisma u proizvoljnom formatu



Ispit #58 (28.06.21. - grupa B)

1. Nadopunite C# kod unutar odgovarajućih metoda tako da se pritiskom na gumb *Flip* simulira bacanje kovanice, čiji je rezultat ili pismo ili glava. Izgled sučelja prikazan je na slici.



Pritiskom na gumb *Flip* se u okviru kontrole *PictureBox* naziva *pictureBox1* slučajnim izborom prikazuje slika glave ili pisma kovanice.

Također, potrebno je u kontrolu tipa *Label* naziva *lblBrojBacanja* ispisati koliko je ukupno bilo bacanja, u *Label* naziva *lblPismo* potrebno je ispisati koliko puta se pojavilo pismo, te u *Label* naziva *lblGlava* ispisati koliko puta se pojavila glava.

Napomena: za prikaz slike glava ili pismo potrebno je koristiti naredbu `pictureBox1.Load(„Glava.jpg“)`, odnosno `pictureBox1.Load(„Pismo.jpg“)`. U exam.netu napisati samo dijelove koda unutar *Nadopuni*.

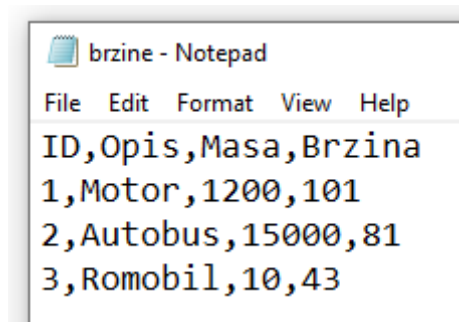
```
public partial class Form1 : Form
{
    //Nadopuni - 1

    public Form1()
    {
        InitializeComponent();
    }

    private void btnFlip_Click(object sender, EventArgs e)
    {
        //Nadopuni - 2
    }
}
```



2. U datoteci naziva *brzine.txt* smještenoj u direktoriju D:\Podaci nalaze se podaci o vozilima i njihovim ostvarenim brzinama (prirodan broj). Primjer podataka prikazan je na slici.



```
brzine - Notepad
File Edit Format View Help
ID,Opis,Masa,Brzina
1,Motor,1200,101
2,Autobus,15000,81
3,Romobil,10,43
```

Potrebno je napisati C# kod konzolne aplikacije koja će iz datoteke *brzine.txt* čitati brzine vozila i spremati ih u listu podaci. Program treba izračunati prosječnu brzinu vozila te u konzolu treba ispisati poruku u formatu: „Prosječna brzina iznosi: x!“, pri čemu x predstavlja vrijednost prosječne brzine.

3. Napišite C# kod metode *Ispis* koja kao argument prima znakovni niz koji predstavlja rečenicu, pri čemu su riječi u rečenici odvojene jednim razmakom.

Metoda treba podijeliti rečenicu na riječi (niz znakova) te ispisati riječi obrnutim redoslijedom. Svaka riječ se treba ispisati u novi redak.

Primjerice, ukoliko se kao argument metode preda niz *mi volimo algpro!*, program treba ispisati:

```
algpor!
volimo
mi
```

4. Napišite C# kod konzolne aplikacije koja stvara dvodimenzionalno polje veličine 3 x 3 slučajnih cijelih brojeva u intervalu [*donja*, *gornja*] kojeg određuje korisnik upisom vrijednosti u konzolu.

Program treba ispisati najveće vrijednosti u svakom pojedinom stupcu u proizvoljnom formatu.

5. Zadano je polje cijelih brojeva [5, 21, 54, -2, 7]. Kako će izgledati polje nakon svih iteracija *selection sort* algoritma (sortirano od najmanjeg do najvećeg elementa).

Potrebno je naznačiti sve zamjene elemenata polja u svakoj iteraciji.



Ispit #59 (28.06.21. - grupa C)

1. Napišite C# kod klase *Dostava* koja sadrži javne atribute *DostavljaclD*, *BrojPaketa*, *Masa* i privatni atribut *Vrijednost*.

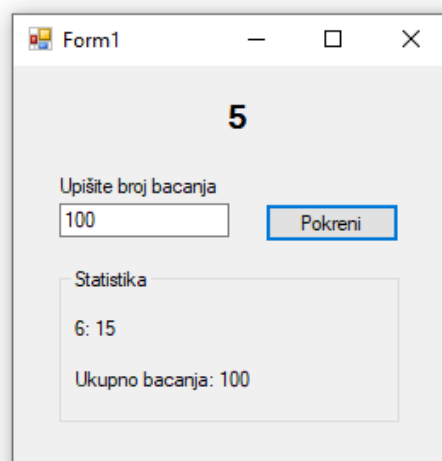
Konstruktor klase prima argumente *DostavljaclD*, *BrojPaketa* i *Masa*, a privatni atribut *Vrijednost* postavlja na vrijednost 0.

Također, potrebno je napisati i javnu metodu *upisVrijednosti* koja kao argument prima vrijednost paketa te postavlja paketu zadanu vrijednost.

2. Nadopunite C# kod aplikacije koja simulira bacanje kockice. Korisnik treba u *TextBox* naziva *txtBrBacanja* upisati koliko bacanja želi simulirati, a aplikacija klikom na gumb *btnStart* s tekstom *Pokreni* simulira zadani broj bacanja i u *Label* pod nazivom *lblVrijednost* ispisuje koji broj se pojavio.

U statistici bacanja treba u *Label* pod nazivom *lblSestica* ispisati koliko je puta pao broj 6, a u *Label* pod nazivom *lblUkupno* treba ispisati koliko je bilo ukupno bacanja.

Potrebno je napisati samo dijelove koda unutar *Nadopuni*.



```
public partial class Form1 : Form
{
    //Nadopuni - 2

    public Form1()
    {
        InitializeComponent();
    }

    private void btnStart_Click(object sender, EventArgs e)
    {
        //Nadopuni - 2
    }
}
```



3. Napišite C# kod konzolne aplikacije koja učitava 250 zapisa imena, prezimena i količine potrošene električne energije u kWh (cijeli broj) s konzole.

Aplikacija za svakog učitano korisnika treba izračunati trošak električne energije u kunama, uz uvjet da cijena jednog kWh iznosi 70 lipa.

Korisnike i njihove podatke potrebno je zapisivati u datoteku *korisnici.txt* preko relativne putanje, svaki u zasebnom retku.

Format zapisa podataka je: Ime,Prezime,potrošnja_u_kWh,cijena_u_kunama.

Potrebno je koristiti try-catch blok, te nije potrebno dodati imenski prostor System.IO.

4. Što će biti zapisano u datoteci *ispis.txt* nakon izvršenja sljedećeg programskog odsječka? Kao točno rješenje priznaje se samo ono u kojem je vidljiv postupak i sadržaj varijabli u svakoj iteraciji. Također je potrebno naznačiti format zapisa sadržaja u datoteku.

```
static void Main(string[] args)
{
    StreamWriter izlaz = new StreamWriter("ispis.txt");
    int n = 10;
    for (int i = 0; i < n; i++)
    {
        izlaz.Write(Fibonacci(i) + " ");
    }
    izlaz.Flush();
    izlaz.Close();
}

static int Fibonacci(int n)
{
    if (n == 0 || n == 1)
        return n;
    else
        return Fibonacci(n - 1) + Fibonacci(n - 2);
}
```

5. Napišite C# kod konzolne aplikacije koja simulira 1000 slučajnih cijelih brojeva unutar intervala [20, 150] km/h koji predstavljaju brzine.

Brzine vozila čija je vrijednost veća od ograničenja koje iznosi 70 km/h potrebno je spremiti u zasebnu listu.

Na kraju je u proizvoljnom formatu potrebno ispisati koliko ima brzina u listi, čija je vrijednost veća od prosječne vrijednosti brzina u listi.



Ispit #60 (08.07.21. - grupa A)

1. Napišite C# kod metode *IzracunStraniceTrokuta* koja prima argumente stranice **a**, **b** i kuta **α** (sve brojevi s plivajućim zarezom).

Metoda *IzracunStraniceTrokuta* u konzolu treba ispisati duljinu stranice **c** zaokruženu na 3 decimale u proizvoljnom formatu.

U *Main* metodi potrebno je unositi stranice i ispisati duljinu stranice **c** sve dok korisnik unosi ispravne vrijednosti, odnosno stranice **a** i **b**, te kut **α** moraju biti veći od nula.

Formula za izračun stranice: $c = \sqrt{a^2 + b^2 - 2ab * \cos(\alpha)}$

2. Napišite C# kod koji ispisuje i računa cijenu najjeftinijeg prijevoznika tereta na osnovu mase tereta i duljine puta. Prijevoznici *CveleTrans*, *PRIJEvoz* i *DataTrans* imaju vlastiti način izračuna cijene prijevoza zadan sljedećom tablicom:

Prijevoznik	Cijena
CveleTrans	$\sqrt{Masa^{\frac{3}{2}} * Duljina}$
PRIJEvoz	$\sqrt{Masa * Duljina^2}$
DataTrans	$Masa * 2 + Duljina$

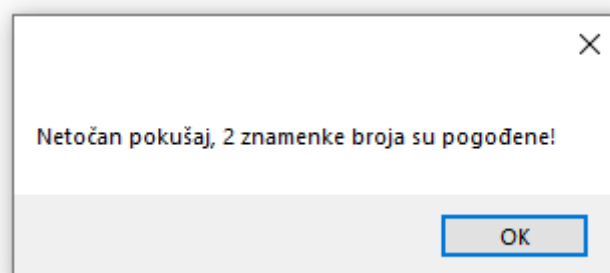
U *Main* dijelu programa potrebo je učitati masu tereta i duljinu puta te u konzolu ispisati poruku u formatu: *Najjeftinija cijena prijevoznika (ime prijevoznika) iznosi x kn!* pri čemu umjesto (*ime prijevoznika*) piše ime, a varijabla *x* predstavlja izračunatu cijenu prijevoza.

3. Napišite C# kod aplikacije koja traži od korisnika da pogodi nasumično generiran prirodan troznamenkasti broj. Korisnik treba pritisnuti gumb *Započni* (*btnGenerirajBroj*) kako bi se generirao slučajni troznamenkasti broj koji treba pogoditi.

Potom, korisnik treba unositi brojeve u *TextBox* naziva *txtPokusaj* te pritiskati gumb *Pogodi* (*btnPogodi*) sve dok ne pogodi nasumično generiran troznamenkasti broj. Povijest pogađanja potrebno je bilježiti u *RichTextBox*-u naziva *txtPovijestPogađanja*.

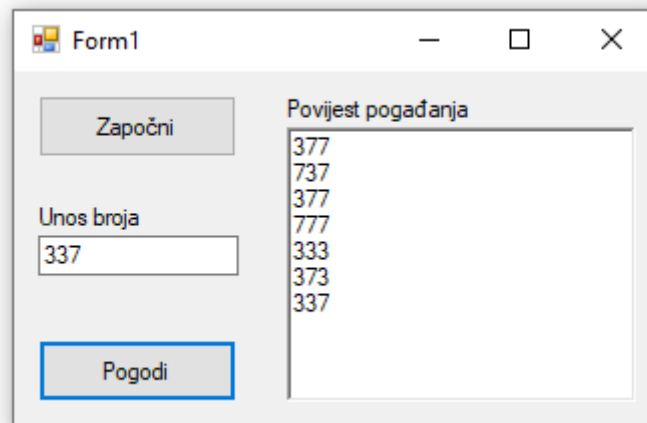
Nakon svakog pokušaja, u *MessageBox* treba se ispisati je li korisnik pogodio broj ili nije. Ukoliko nije pogodio broj, u *MessageBox*-u je potrebno ispisati koliko znamenki (računajući redoslijed) je korisnik pogodio. Za dohvat znamenki cijelog broja potrebno je koristiti već napisanu metodu *dohvatiZnamenke* (nije ju potrebno pisati), koja vraća polje cijelih brojeva, pri čemu je prvi element polja prva znamenka predanog broja, a zadnji element polja zadnja znamenka predanog broja.

Izgled poruke prikazan je na slici:



U exam.netu napisati samo dijelove koda unutar *Nadopuni*.

Izgled sučelja prikazan je na slici:



```
public partial class Form1 : Form
{
    //Nadopuni - 1

    public Form1()
    {
        InitializeComponent();
    }

    private void btnGenerirajBroj_Click(object sender, EventArgs e)
    {
        //Nadopuni - 2
    }

    private void btnPogodi_Click(object sender, EventArgs e)
    {
        //Nadopuni - 3
    }

    private static int[] dohvatiZnamenke(int broj)
    {
        ...
    }
}
```



4. Napišite C# kod konzolne aplikacije koja računa cijenu autobusnog prijevoza.

Potrebno je prvo napraviti klasu naziva *BusPrijevoznici* koja sadrži javne atribute *ImePrijevoznika*, *BrojSjedala* (cijeli broj), *CijenaKarte* (broj s plivajućim zarezom).

U *Main* dijelu programa potrebno je kreirati 2 autobusna prijevoznika po želji preko konstruktora klase koji treba primiti sve atribute (nije potrebno učitavanje iz konzole).

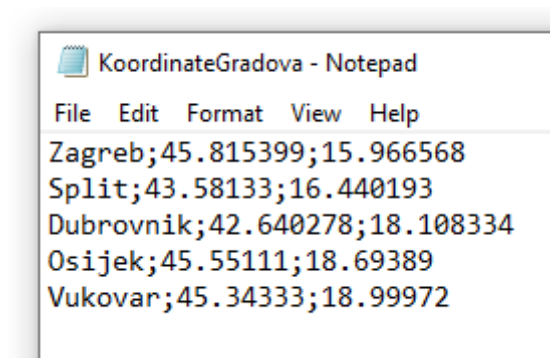
Potom je preko konzole potrebno unijeti broj putnika i udaljenost koju treba prevesti, te na temelju unesenog, izračunati i ispisati ukupnu dobit za svakog prijevoznika po formuli:

$$\text{Dobit} = \text{BrojPutnika} * \text{CijenaKarte} - \text{Udaljenost} * 4.5$$

Ispis u konzolu treba sadržavati ime prijevoznika i njegovu dobit.

5. Napišite C# kod programa koji pritiskom na gumb *Učitaj* (*btnUcitaj*) iz tekstualne datoteke *KoordinateGradova.txt*, smještenej na relativnoj putanji, učitava zapise *ImeGrada*; *širina*; *dužina*, pri čemu je *širina* geografska širina (y koordinata), a *dužina* geografska dužina (x koordinata).

Izgled datoteke *KoordinateGradova.txt* prikazan je na slici:



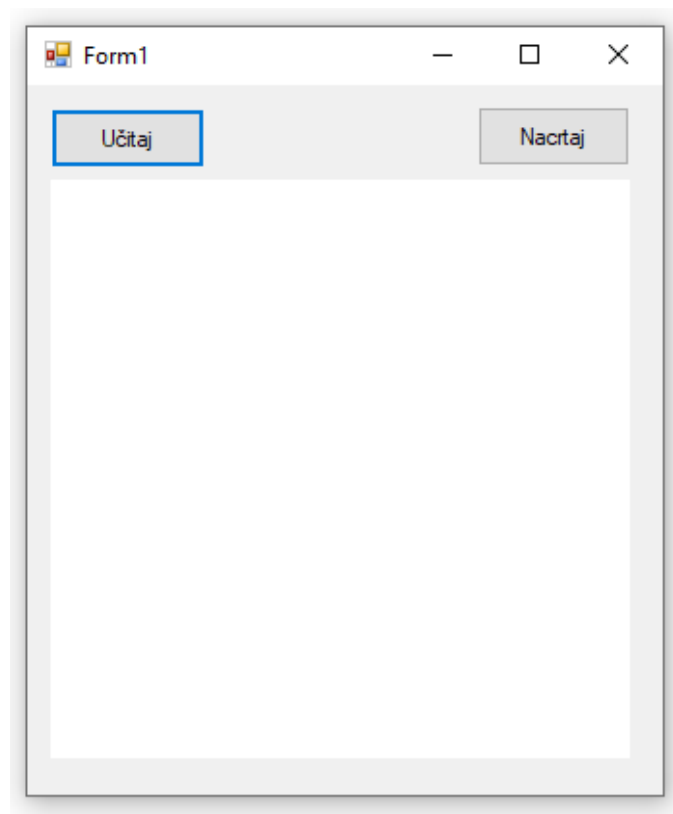
Pritiskom na gumb *Nacrtaj* (*btnNacrtaj*) potrebno je u *PictureBox* naziva *slika* iscrtati liniju između točke (0,0) i grada najudaljenijeg točki (0,0). Dio C# koda za iscrtavanje linije već je napisan.

U exam.netu napisati samo dijelove koda unutar *Nadopuni*.

Za izračun udaljenosti između dvije točke koristiti Euklidsku udaljenost.



Izgled sučelja prikazan je na slici:



```
public partial class Form1 : Form
{
    //Nadopuni - 1

    public Form1()
    {
        InitializeComponent();
    }

    private void btnUcitaj_Click(object sender, EventArgs e)
    {
        //Nadopuni - 2
    }

    private void btnNacrtaj_Click(object sender, EventArgs e)
    {
        //Nadopuni - 3

        slika.Refresh();
    }

    private void slika_Paint(object sender, PaintEventArgs e)
    {
        e.Graphics.DrawLine(Pens.Black, 0, 0, (float)xND, (float)yND);
    }
}
```



Ispit #61 (08.07.21. - grupa B)

1. Napišite C# kod metode *OpsegMnogokuta* koja kao argument prima listu duljina stranica mnogokuta. Metoda u konzolu treba ispisati opseg mnogokuta u formatu: „Mnogokut sa n stranica ima opseg O “, pri čemu vrijednost varijabli n i O treba zamijeniti odgovarajućim vrijednostima.

U *Main* metodi je potrebno preko konzole unijeti broj n koji predstavlja broj stranica mnogokuta, popuniti listu s duljinama stranica (broj s plivajućim zarezom) također učitanih s konzole i pozvati metodu *OpsegMnogokuta*.

2. Napišite C# kod konzolne aplikacije koja traži od korisnika da pogodi slučajno generiran prirodan troznamenkasti broj. Korisnik mora unositi brojeve sve dok ne pogodi nasumično generiran troznamenkasti broj.

Također, aplikacija nakon svakog pokušaja mora ispisati je li korisnik pogodio znamenke broja kojeg pogađa, te ako nije treba ispisati koliko znamenki (prema redoslijedu) je pogodio.

Pomoć: za dohvat pojedinih znamenki troznamenkastog broja može se koristiti cjelobrojno dijeljenje i operator za izračun ostatka dijeljenja.

Primjer:

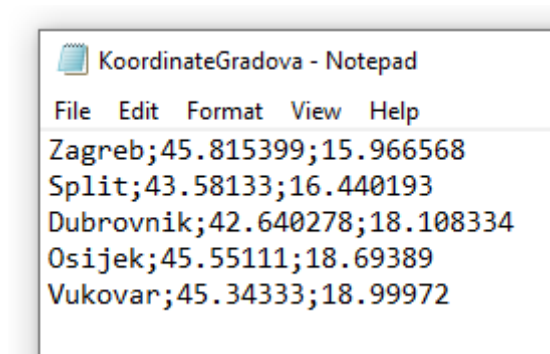
Nasumično generiran broj: 231

Korisnik unio broj: 831

Primjer ispisa: „Netočan pokušaj, 2 znamenke broja su pogođene!“



3. Napišite C# kod programa koji učitava tekstualnu datoteku *KoordinateGradova.txt* smještenu na relativnoj putanji, s 10 redova zapisa u formatu *ImeGrada;širina;dužina*, pri čemu je *širina* geografska širina (y koordinata), a *dužina* geografska dužina (x koordinata). Izgled datoteke *KoordinateGradova.txt* prikazan je na slici:



U *Main* dijelu programa potrebno je pročitati sadržaj datoteke i spremiti ju na proizvoljan način.

Potom, napišite i pozovite metodu *NajvecaUdaljenost* koja kao argument prima učitane podatke (prethodno odabran način spremanja podataka) te ispisuje ime i udaljenost grada koji je najudaljeniji od točke (0, 0).

Primjer ispisa: „*Najudaljeniji grad je grad D, s udaljenosti X km!*“, pri čemu vrijednost *G* i *X* treba zamijeniti s nazivom grada, odnosno udaljenošću. Za izračun udaljenosti potrebno je koristiti Euklidsku udaljenost pomnoženu s vrijednošću 111, jer je $1^\circ \approx 111\text{km}$.

4. Napišite C# kod konzolne aplikacije koja računa cijenu taksi prijevoza. Potrebno je prvo napisati klasu *TaxiPrijevoznici* koja sadrži privatni atribut *CijenaStart* i javne attribute *CijenaKmDan*, *CijenaKmNoc* (sve brojevi s plivajućim zarezom) i u *Main* dijelu kreirati 2 taxi prijevoznika po želji preko konstruktora koji prima sve attribute klase (nije potrebno učitavanje iz konzole).

Nakon toga, preko konzole korisnik treba unijeti doba dana i duljinu puta na temelju čega je potrebno ispisati dobit svakog od kreiranih taxi prijevoznika u proizvoljnom formatu.

Za dohvat odgovarajućih privatnih atributa potrebno je napisati odgovarajuće metode u klasi.

Dobit se računa sljedećim izrazom:

$$D = \text{CijenaStart} + \text{CijenaKm} * \text{DuljinaPut}$$



5. Nadopišite C# kod koji računa i ispisuje cijenu najjeftinijeg prijevoznika tereta na osnovu mase tereta i duljine puta. Prijevoznici *CveleTrans*, *PRIJEvoz* i *DataTrans* imaju vlastiti način izračuna cijene prijevoza zadan tablicom:

Prijevoznik	Cijena
CveleTrans	$\sqrt{Masa^{\frac{2}{3}} * Duljina}$
PRIJEvoz	$\sqrt{Masa * Duljina^2}$
DataTrans	$Masa * 2 + Duljina$

Pritiskom na gumb *Izračunaj* (*btnIzracunaj*) potrebno je izračunati cijenu prijevoza za svakog prijevoznika na temelju upisanih vrijednosti mase tereta u *TextBox* (*txtMasa*) i duljine puta u *TextBox* (*txtDuljina*) (brojevi s plivajućim zarezom).

Varijable cijene i imena najjeftinijeg prijevoznika potrebno je deklarirati na razini klase kako bi varijable bile vidljive svim metodama unutar klase.

Ukupnu cijenu i ime najjeftinijeg prijevoznika potrebno je ispisati u skočnom prozoru (*MessageBox*) u formatu *Najjeftinija cijena prijevoznika (ime prijevoznika) iznosi: x kn!* pri čemu umjesto (*ime prijevoznika*) treba pisati ime najjeftinijeg prijevoznika, a varijabla *x* predstavlja izračunatu cijenu prijevoza.

Sučelje je prikazano sljedećom slikom:

U exam.netu potrebno je napisati samo dijelove koda unutar *Nadopuni*.

```
public partial class Form1 : Form
{
    //Nadopuni - 1

    public Form1()
    {
        InitializeComponent();
    }

    private void btnIzracunaj_Click(object sender, EventArgs e)
    {
        //Nadopuni - 2
    }
}
```



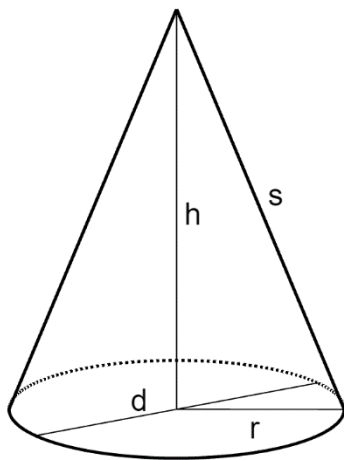
Ispit #62 (08.07.21. - grupa C)

Napomena: rok nije kompletan (nedostaje zadatak 5)

1. Napišite C# kod koji temeljem unosa korisnika računa oplošje ili volumen stošca.

Potrebno je napraviti metodu *IzracunZaStozac* koja kao argument prima izbor unesen od korisnika (oplošje ili volumen) i dimenzije stošca (polumjer i visinu – realni brojevi), a kao rezultat vraća oplošje ili volumen stošca ovisno o izboru.

U *Main* metodi potrebno je unijeti izbor unesen od strane korisnika i dimenzije stošca (polumjer i visinu), pozvati metodu *IzracunZaStozac* i ispisati rezultat u konzolu u proizvoljnom formatu.



$$V = \frac{1}{3}\pi r^2 h$$

$$O = \pi r(r + s)$$

$$s = \sqrt{r^2 + h^2}$$

V – obujam

O – oplošje

r – polumjer

s – polumjer plašta

h – visina

2. Napišite C# kod konzolne aplikacije koja izračunava zbroj znamenki dva peteroznamenkasta broja. Učitavanje dva broja i ispis zbroja njihovih znamenki u proizvoljnom formatu potrebno je vršiti sve dok se unose ispravni peteroznamenkasti brojevi u rasponu [10000, 99999].

3. Napišite C# kod konzolne aplikacije koja računa cijenu avionskih karata.

Potrebno je prvo napraviti klasu *AvioKompanije* koja sadrži javne attribute *ImePrijevoznika*, *BrojSjedala* (cijeli broj), *CijenaKarte* (cijeli broj) te u *Main* dijelu kreirati 2 proizvoljna avio prijevoznika bez konstruktora (instance klase *AvioKompanije*, nije potrebno učitavanje vrijednosti iz konzole)).

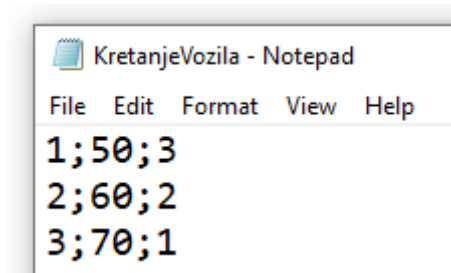
Nakon toga, preko konzole korisnik treba unijeti broj putnika (cijeli broj) koji će se prevesti. Na temelju unesenog, izračunajte i ispišite neostvarenu dobit za svakog prijevoznika po formuli:

$$\text{NeostvarenaDobit} = \text{BrojPraznihSjedala} * \text{CijenaKarte}$$

Format ispisa je proizvoljan, ali mora sadržavati ime prijevoznika i neostvarenu dobit.



4. Napišite C# kod programa koji iz tekstualne datoteke *KretanjeVozila.txt* Smještene na D: disku, s 1000 redova zapisa, učitava zapise u formatu *VoziloID;brzina_km/h;VrijemePutovanja_h*. Izgled datoteke *KretanjeVozila.txt* prikazan je na slici:



U *Main* dijelu programa potrebno je pročitati i spremiti sadržaj datoteke u dvodimenzionalno polje.

Napišite metodu *DuljinaPutovanja* koja kao argument prima učitano polje i ID vozila za koji se želi izračunati prijeđena udaljenost na temelju brzine vozila i vremena putovanja. Metoda treba ispisati prijeđenu udaljenost vozila.

Format ispisa: „*Prijeđena udaljenost vozila (IDVozila) je X km*“. Varijabla *X* kod ispisa predstavlja izračunatu prijeđenu udaljenost vozila, dok (*IDVozila*) treba zamijeniti ID-em vozila za kojeg se računa udaljenost.



Ispit #63 (02.09.21. - grupa A)

1. Kako će izgledati polje 45, -8, -44, 6, 99, 7 nakon svih iteracija selection sort algoritma (sortiranje od najmanjeg prema najvećem).

Potrebno je naznačiti svaku zamjenu elemenata polja zasebno u svim koracima svake iteracije.

2. Što će biti zapisano u datoteci *data.txt* nakon što se izvrši naredba *Flush* na kraju sljedećeg programskog odsječka.

Kao točno rješenje priznaje se samo ono u kojem je vidljiv postupak i sadržaj varijabli u svakoj iteraciji i sadržaj upisan u datoteku nakon izvršavanja.

```
static void Main(string[] args)
{
    StreamWriter izlaz = new StreamWriter("data.txt");
    for (int i = 2; i <= 13; i++)
    {
        if (Provjera(i))
        {
            izlaz.WriteLine("PAR");
        }
        else
        {
            izlaz.WriteLine(i);
        }
        if (i % 8 == 1) break;
    }
    izlaz.Flush();
    izlaz.Close();
}
public static bool Provjera(int i)
{
    return (i % 2 == 0);
}
```



3. Napišite C# kod metode *IzracunVolumenaStosca* koja prima argumente polumjera baze *r* i visine *h* (brojevi s plivajućim zarezom).

Metoda *IzracunVolumenaStosca* u konzolu treba ispisati volumen stošca zaokružen na 2 decimale u proizvoljnom formatu.

U Main metodi potrebno je osigurati unos duljine radijusa i visine stošca, te napraviti provjeru upisanih vrijednosti (*r* i *h* moraju biti veći od 0).

Formula za izračun volumena $V = \frac{1}{3}r^2\pi h$

4. Napišite C# kod konzolne aplikacije za upravljanje parkiralištem.

Potrebno je napisati klasu *Vozilo* koja sadrži javne atribute *Registracija (tekst)*, *VrijemeUlaska (DateTime)*, *VrijemeIzlaska (DateTime)*.

U aplikaciji je prvo potrebno odabrati radi li se o ulasku novog vozila ili o izlasku vozila s parkirališta. Ukoliko se radi o ulasku potrebno je zabilježiti vrijeme dolaska vozila, a ukoliko se radi o izlasku potrebno je ispisati vrijeme provedeno na parkiralištu.

Sva vozila unutar parkirališta trebaju imati zaseban objekt unutar liste *ListaVozilaParking* koja je već kreirana unutar aplikacije i nije ju potrebno inicijalizirati.

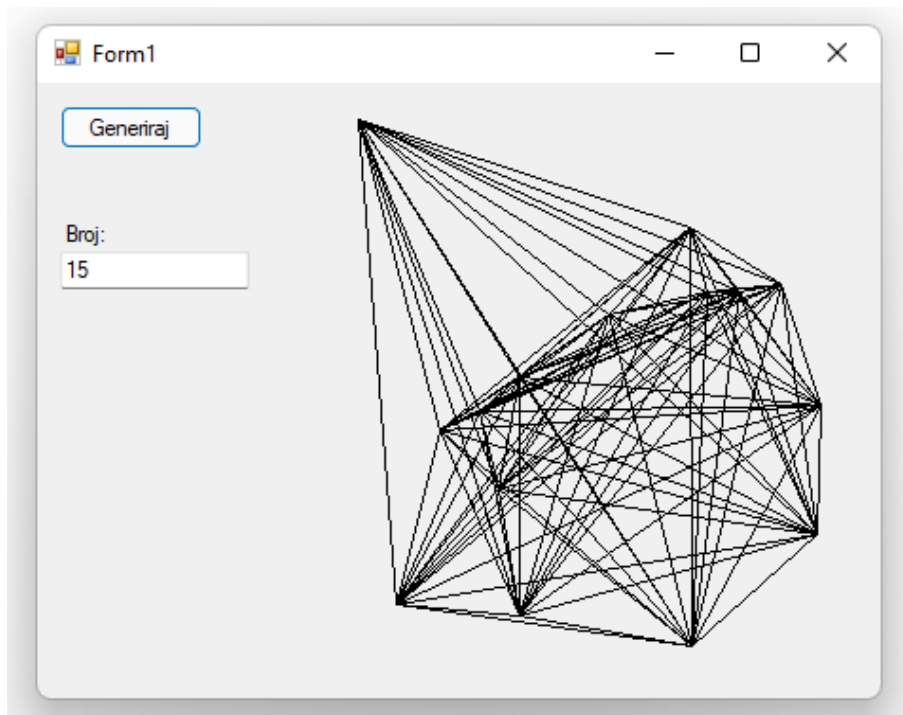


5. Napišite C# kod programa koji pritiskom na gumb *Generiraj* stvara dvodimenzionalno polje s dva stupca i brojem redaka jednakih vrijednosti upisanoj u TextBoxu *txtBoxBroj*.

U svaki element polja potrebno je upisati nasumični cijeli broj u rasponu od 0 do 300. Prvi stupac predstavlja x-koordinatu, dok drugi predstavlja y koordinatu točke.

Nakon generiranja brojeva, potrebno je u PictureBox (postavljen na dimenzije 300 x 300) iscrtati sve moguće linije koje povezuju generirane točke.

U exam.netu napisati samo dijelove koda unutar Nadopuni.



```
public partial class Form1 : Form
{
    //Nadopuni 1

    public Form1()
    {
        InitializeComponent();
    }

    private void btnGeneriraj_Click(object sender, EventArgs e)
    {
        //Nadopuni 2

        pictureBox1.Refresh();
    }

    private void pictureBox1_Paint(object sender, PaintEventArgs e)
    {
        //Nadopuni 3
    }
}
```



Ispit #64 (02.09.21. - grupa B)

1. Kako će izgledati polje 45, -8, -44, 6, 99, 7 nakon svih iteracija insertion sort algoritma (**sortiranje najvećeg prema najmanjem**). Potrebno je naznačiti svaku zamjenu elemenata polja zasebno u svim koracima svake iteracije.

2. Što će biti zapisano u datoteci *data.txt* nakon što se izvrši naredba *Flush* na kraju sljedećeg programskog odsječka.

Kao točno rješenje priznaje se samo ono u kojem je vidljiv postupak i sadržaj varijabli u svakoj iteraciji i sadržaj upisan u datoteku nakon izvršavanja naredbe.

```
static void Main(string[] args)
{
    StreamWriter izlaz = new StreamWriter("data.txt");
    bool[] A = new bool[14];
    for (int i = 2; i <= 13; i++) A[i] = true;
    for (int i = 2; i <= 13; i++)
    {
        if (A[i])
        {
            izlaz.WriteLine(i);
            for (int j = i * 2; j <= 13; j = j + i)
            {
                A[j] = false;
            }
        }
    }
    izlaz.Flush();
    izlaz.Close();
}
```

3. Napišite C# kod metode *IzracunVolumenaPiramide* koja prima argumente površine baze *B* i visine *h* (sve brojevi s plivajućim zarezmom).

Metoda *IzracunVolumenaPiramide* u konzolu treba ispisati volumen piramide zaokružen na 3 decimale u proizvoljnom formatu.

U Main metodi potrebno je osigurati unos površine baze i visine piramide, te napraviti provjeru upisanih vrijednosti (*B* i *h* moraju biti veći od 0).

Formula za izračun volumena $V = \frac{1}{3} B * h$



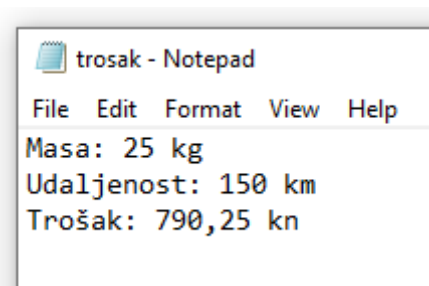
4. Napišite C# kod aplikacije za izračun troška prijevoza tereta prema unesenim parametrima *Udaljenost* i *Masa* (brojevi s plivajućim zarezom).

Klikom na gumb *Izracunaj* (*btnIzracunaj*) računa se ukupni trošak koji se ispisuje u obliku MessageBox-a. Trošak je potrebno izračunati prema izrazu:

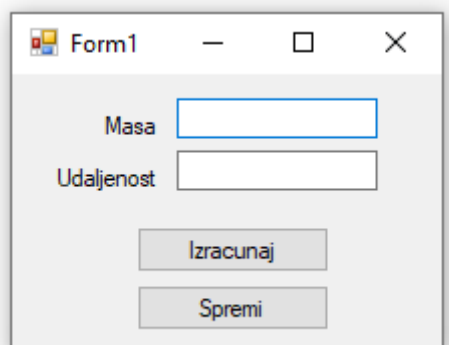
$$T = 100 + Udaljenost * 4,23 + Masa * 2,23, \quad \text{ako je } Masa < 100kg$$

$$T = 100 + Udaljenost * 4,23 + Masa * 1,98, \quad \text{ako je } Masa \geq 100kg$$

Nakon izračuna klikom na gumb *Spremi* (*btnSpremi*) se posljednji izračun sprema u tekstualnu datoteku prema primjeru na slici:



U exam.netu napisati samo dijelove koda unutar Nadopuni.



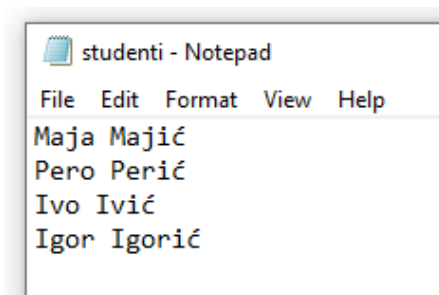
```
public partial class Form1 : Form
{
    //Nadopuni 1
    public Form1()
    {
        InitializeComponent();
    }

    private void btnIzracunaj_Click(object sender, EventArgs e)
    {
        //Nadopuni 2
    }

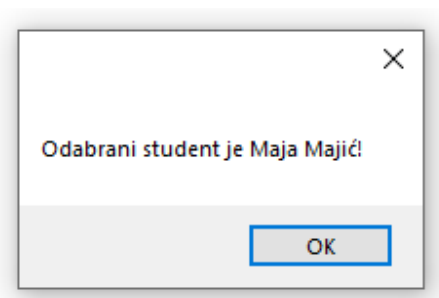
    private void btnSpremi_Click(object sender, EventArgs e)
    {
        //Nadopuni 3
    }
}
```



5. Napišite C# kod aplikacije koja može učitati tekstualnu datoteku s popisom studenata koja je prikazana na slici:



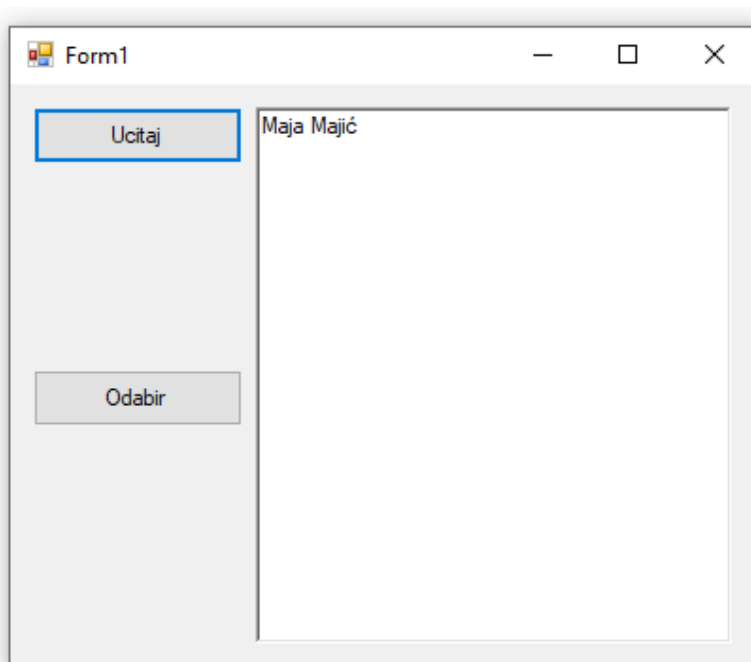
Korisnik treba pritisnuti gumb *Odabir* (*btnOdabir*) kako bi se s liste studenata odabrao nasumičan student, te se u MessageBox-u trebaju ispisati njegovi podaci kao u primjeru:



Ponovnim klikom na gumb *Odabir* bira se novi nasumični student, ali jednom odabrani student više ne može biti odabran do ponovnog učitavanja datoteke.

Podaci o svakom odabranom studentu se bilježe u RichTextBox-u *richTextBoxOdabrani*.

U exam.netu napisati samo dijelove koda unutar Nadopuni.



```
public partial class Form1 : Form
{
    //Nadopuni 1
    public Form1()
    {
        InitializeComponent();
    }

    private void btnUcitaj_Click(object sender, EventArgs e)
    {
        //Nadopuni 2
    }

    private void btnOdabir_Click(object sender, EventArgs e)
    {
        //Nadopuni 3
    }
}
```



Ispit #65 (02.09.21. - grupa C)

1. Kako će izgledati polje 45, 6, 99, 7 nakon svih iteracija bubble sort algoritma (sortiranje od najvećeg prema najmanjem). Potrebno je naznačiti svaku zamjenu elemenata polja zasebno u svim koracima svake iteracije.

2. Što će biti zapisano u datoteci *data.txt* nakon što se izvrši naredba *Flush* na kraju sljedećeg programskog odsječka.

Kao točno rješenje priznaje se samo ono u kojem je vidljiv postupak i sadržaj varijabli u svakoj iteraciji i sadržaj upisan u datoteku nakon izvršavanja naredbe.

```
static void Main(string[] args)
{
    StreamWriter izlaz = new StreamWriter("data.txt");
    int[] niz = { 3, 4, 2, 1, 5, 6 };
    int temp = 0;
    for (int i = 0; i < niz.GetLength(0); i++)
    {
        if (niz[i] > temp)
        {
            izlaz.WriteLine(niz[i]);
            temp = niz[i];
        }
        else
        {
            izlaz.Write(0);
            temp = niz[i];
        }
    }
    izlaz.Flush();
    izlaz.Close();
}
```

3. Napišite C# kod koji ispisuje i računa cijenu najjeftinijeg pozivnog vozila (TAXI) na osnovu broja putnika i duljine puta. Prijevoznici TAXI-M, ZooM i FPZTAXI imaju vlastiti način izračuna cijene prijevoza zadan sljedećom tablicom:

Prijevoznik	Cijena
TAXI-M	$10 * BrojPutnika + Duljina * 5,43$
ZooM	$4 * BrojPutnika + Duljina * 7,32$
FPZTAXI	$20 + BrojPutnika * Duljina * 3,45$

U *Main* dijelu programa potrebno je učitati broj putnika (cijeli broj) i duljinu puta (broj s plivajućim zarezom) te u konzolu ispisati poruku u formatu: *Najjeftinija cijena (ime prijevoznika) iznosi: x kn!* pri čemu umjesto *(ime prijevoznika)* treba napisati ime najjeftinijeg prijevoznika, a varijabla *x* predstavlja izračunatu cijenu prijevoza.



4. Napišite C# kod aplikacije za evidentiranje ocjena studenata na ispitu.

Potrebno je napisati klasu Student s javnim atributima: ime, prezime, JMBAG, ocjena.

U glavnoj metodi programa potrebno je učitati n studenata, pri čemu se n unosi s konzole. Nakon unosa svih podataka potrebno je ispisati prosječnu ocjenu svih studenata na ispitu, te ispisati ime i prezime studenata s najvišim brojem bodova.

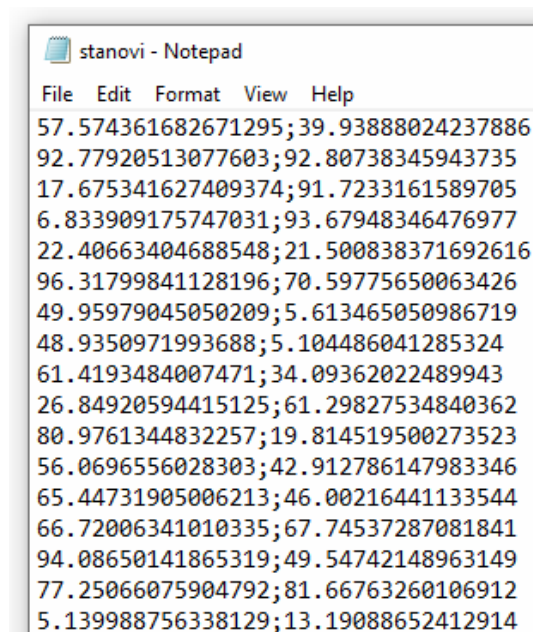
Napomena: ovaj rok je prepun grešaka i zadataka koji nisu dovoljno precizno definirani (posebice ova grupa). Greška u ovom zadatku je što se u ovom zadatku traži unos ocjene, a za ispis se traži ispis studenata s najvišim brojem bodova. Pretpostavka je da je bila greška i da je i za ispis trebalo napraviti samo ispis studenata s najvećom ocjenom.

Ovakve greške sam u prošlim rokovima ispravio bez da sam napomenuo kakve greške su bile na ispitima, ali sam ovdje htio skrenuti pozornost na učestalost ovakvih pogrešaka.

Još jedna od stvari koje nisu dovoljno definirane je korištenje klase, jer se ovaj zadatak može riješiti bez upotrebe iste, ali je pretpostavka da su željeli da se svaki student kreira kao instanca klase Student.

Inače su ispiti i na kolegijima „Računalstvo“ i „Algoritmi i programiranje“ tražili samo ispis jednog studenta s najvećom ocjenom, ali kako je ovdje navedeno traži se ispis studenata (u množini) pa sam odlučio riješiti zadatak na taj način (koji je ujedno i teži).

5. Napišite C# kod koji učitava koordinate stambenih objekata iz tekstualne datoteke stanovi.txt.



```
File Edit Format View Help
57.574361682671295;39.93888024237886
92.77920513077603;92.80738345943735
17.675341627409374;91.7233161589705
6.833909175747031;93.67948346476977
22.40663404688548;21.500838371692616
96.31799841128196;70.59775650063426
49.95979045050209;5.613465050986719
48.9350971993688;5.104486041285324
61.4193484007471;34.09362022489943
26.84920594415125;61.29827534840362
80.9761344832257;19.814519500273523
56.0696556028303;42.912786147983346
65.44731905006213;46.00216441133544
66.72006341010335;67.74537287081841
94.08650141865319;49.54742148963149
77.25066075904792;81.66763260106912
5.139988756338129;13.19088652412914
```

Nakon učitavanja koordinata potrebno je odrediti optimalnu poziciju za izgradnju telekomunikacijske bazne stanice (središte učitanih koordinata) te ispisati izračunate koordinate u konzolu.



Ispit #66 (06.11.21. - grupa A)

1. Dva vozila V_1 i V_2 voze se po istoj prometnici. Kretanje vozila po prometnici može se opisati linearnom jednačom u obliku $y = kx + l$, gdje y predstavlja prijeđenu udaljenost (km), x vrijeme (h), k brzinu (km/h) i l početnu prijeđenu udaljenost (km).

Potrebno je napisati C# kod programa koji za svako od vozila V_1 i V_2 s konzole učitava **brzinu vozila (m/s) i početnu prijeđenu udaljenost vozila u km** (brojevi s plivajućim zarezom).

Nakon učitavanja, ukoliko su učitane brzine različite, potrebno je u konzolu u formatu kao u primjeru ispisati vrijeme (**zaokruženo na jednu decimalu**) i prijeđenu udaljenost (**bez decimala**) u kojima će se vozila susresti (sjecište dvaju pravaca), inače je u konzolu potrebno ispisati *Vozila imaju istu brzinu v m/s!*, gdje je umjesto v potrebno ispisati odgovarajuću vrijednost brzine.

Primjer ispisa za slučaj različitih brzina: *Vozilo V1 s brzinom 76.5 km/h i početnom udaljenosti 20.35 km će se susresti s vozilom V2 brzine 84.096 km/h i početnom udaljenosti 10.1678 km u trenutku 1.4 h na 129 km!*. Pomoć: $1 \text{ km/h} = \frac{1}{3.6} \text{ m/s}$.

2. Napišite C# kod metode *UmnozakMatrica* koja preko argumenata prima dvije cjelobrojne matrice A i B (nisu nužno iste veličine), te kao rezultat vraća novu matricu C .

Ukoliko je broj stupaca matrice A različit od broja redaka matrice B metoda treba vratiti *nul* matricu (matrica sa samo jednim elementom, koji je jednak 0). Inače, metoda treba vratiti matricu C kao umnožak matrica A i B .

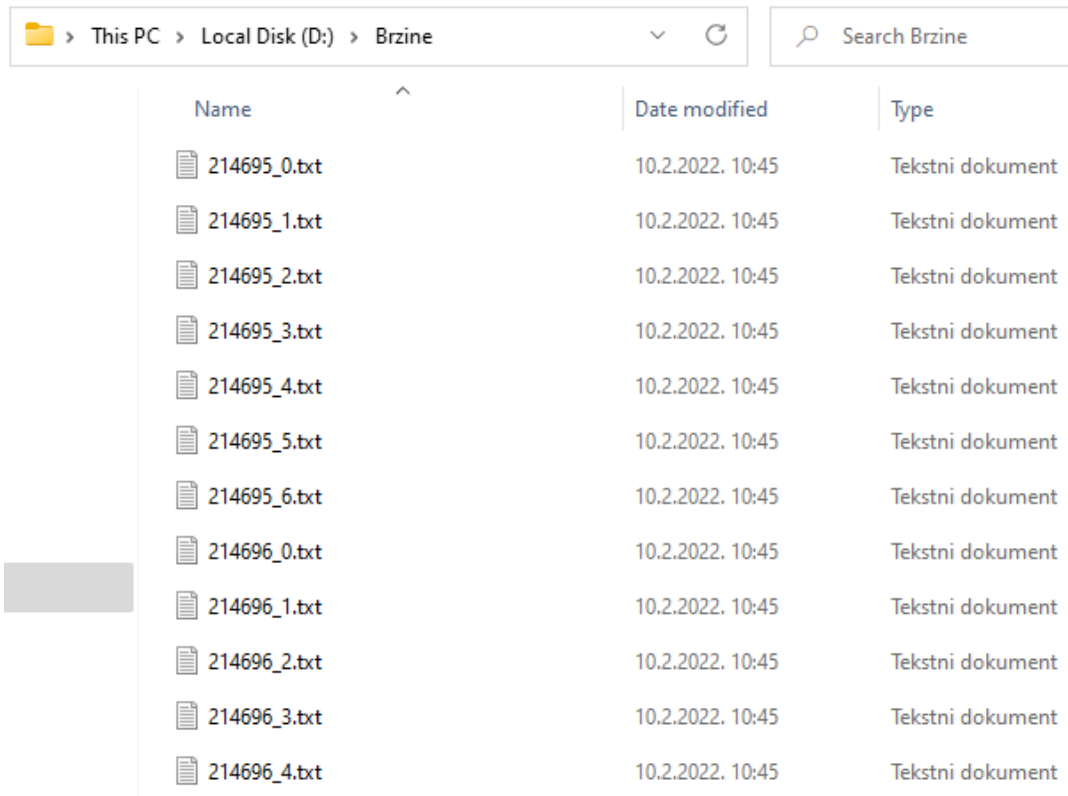
Pomoć: primjer izračuna umnoška matrica $A^{2 \times 3}$ i $B^{3 \times 2}$ dan je sljedećom jednačom:

$$\begin{aligned} C = A * B &= \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \end{bmatrix} * \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \\ b_{31} & b_{32} \end{bmatrix} \\ &= \begin{bmatrix} a_{11}b_{11} + a_{12}b_{21} + a_{13}b_{31} & a_{11}b_{12} + a_{12}b_{22} + a_{13}b_{32} \\ a_{21}b_{11} + a_{22}b_{21} + a_{23}b_{31} & a_{21}b_{12} + a_{22}b_{22} + a_{23}b_{32} \end{bmatrix} \end{aligned}$$



3. Napišite C# kod konzolne aplikacije za izračun akceleracija. Potrebno je napisati program koji s konzole učitava ID prometnice (cijeli broj) i indeks dana u tjednu (cijeli broj: 0 označava ponedjeljak, a 6 nedjelju).

Potom je potrebno učitati peto-minutne brzine u danu iz odgovarajuće tekstualne datoteke koja se nalazi u direktoriju *D:\Brzine*. Datoteka koja sadrži peto-minutne brzine ima naziv u formatu *ID_indeksDana.txt*, kao što je to prikazano na sljedećoj slici:



Name	Date modified	Type
214695_0.txt	10.2.2022. 10:45	Tekstni dokument
214695_1.txt	10.2.2022. 10:45	Tekstni dokument
214695_2.txt	10.2.2022. 10:45	Tekstni dokument
214695_3.txt	10.2.2022. 10:45	Tekstni dokument
214695_4.txt	10.2.2022. 10:45	Tekstni dokument
214696_0.txt	10.2.2022. 10:45	Tekstni dokument
214696_1.txt	10.2.2022. 10:45	Tekstni dokument
214696_2.txt	10.2.2022. 10:45	Tekstni dokument
214696_3.txt	10.2.2022. 10:45	Tekstni dokument
214696_4.txt	10.2.2022. 10:45	Tekstni dokument

Svaka brzina (za pojedini peto-minutni interval) zapisana je u zasebnom retku. Na temelju učitanih brzina potrebno je izračunati akceleracije u peto-minutnim intervalima pomoću sljedećeg izraza:

$$a = \frac{v_2 - v_1}{\Delta t}$$

Izračunajte akceleracije za prethodno odabrani ID prometnice i dan, potrebno je zapisati u direktorij *D:\Akceleracije* u datoteku s istim nazivom kao i odgovarajuća datoteka brzina. Peto-minutne akceleracije potrebno je zapisati u jedan redak s oznakom ; između susjednih akceleracija. Potrebno je paziti da se akceleracija za prvi peto-minutni interval u danu ne može izračunati jer ne postoji zapis brzine prije.

Potrebno je koristiti *try-catch* blok nad čitavim kodom, te nije potrebno pisati *Using System.IO*.



4. Što će se ispisati u konzolu nakon izvršavanja sljedećeg programskog odsječka. Potrebno je naznačiti redoslijed ispisivanja. Kao točno rješenje priznaje se samo ono u kojem je vidljiv postupak.

```
int suma = 0, i = 0, j = 4;
for (i = 1; i < j / 2; i += 1)
{
    for (j = 4; j <= 6; j++)
    {
        if (j % 2 == 1)
        {
            for (int k = 3; k < j; k += i)
            {
                suma += i * k;
            }
        }
        else
        {
            suma += i + j;
        }
        Console.WriteLine(suma);
    }
}
```

5. Što će se ispisati u konzolu nakon izvršavanja sljedećeg programskog odsječka. Potrebno je naznačiti redoslijed ispisivanja. Kao točno rješenje priznaje se samo ono u kojem je vidljiv postupak.

```
List<int> lista = new List<int>();

for (int i = 81; i > 0; i /= 3)
{
    lista.Add(i++);
}
lista[1] = lista[4] / lista[2];
lista.Remove(0);
lista.InsertRange(1, new int[] { lista[1] / 2, lista[lista.Count - 1] * 4 });
lista.RemoveRange(lista.Count % 4, 2);
for (int i = lista.Count - 1; i >= 0; i--)
{
    Console.WriteLine(lista[i]);
}
```



Ispit #67 (03.02.22. - grupa A)

1. Napišite C# kod konzolne aplikacije koja učitava matricu udaljenosti između n gradova. Potrebno je prvo preko tipkovnice unijeti vrijednost n , potom kreirati matricu $M = n \times n$, te ju popuniti brojevima s plivajućim zarezom koji predstavljaju udaljenosti između gradova.

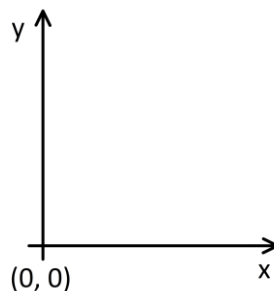
Na kraju je potrebno ispisati prosječnu vrijednost svakog retka u formatu *Prosječna vrijednost retka pod indeksom i iznosi x!*, pri čemu i predstavlja indeks retka, a x iznos prosječne vrijednosti u tom retku.

2. Napišite C# kod konzolne aplikacije koja simulira pomicanje točke na ekranu. S tipkovnice je potrebno kontinuirano učitavati znak po znak, pri čemu se očekuje da se mogu unositi samo znakovi U, D, L i R .

Početna pozicija točke je $(x, y) = (0, 0)$ (cijeli brojevi), te se u svakoj iteraciji točka pomiče za 1 (cijeli broj).

U predstavlja pomicanje gore, D pomicanje dolje, L pomicanje u lijevo, a R pomicanje u desno. Učitavanje znakova traje dok god je točka unutar ekrana, odnosno dok su zadovoljeni sljedeći uvjeti $x \in [-20, 20]$ i $y \in [-20, 20]$.

Na kraju je potrebno u konzolu ispisati konačnu poziciju točke, koliko je ukupno pomaka točke bilo, te također pojedinačno broj pomaka u svakom od smjerova: gore, dolje, lijevo i desno. Razmatra se klasično postavljen koordinatni sustav prikazan ispod.



3. Napišite C# kod konzolne aplikacije koja učitava sadržaj datoteke *brzineNaPrometnici.txt* u kojoj su zapisane brzine (brojevi s plivajućim zarezom), svaka u zasebnom retku, ali s praznim retkom između dva zapisa kako je prikazano na sljedećoj slici:



Sve vrijednosti datoteke potrebno je učitati u odgovarajuću listu.

Na kraju je na slučajan način potrebno dohvatiti 10 brzina i liste i ispisati ih u formatu: *Brzina iznosa x nalazi se na indeksu i u listi!*, pri čemu *x* predstavlja vrijednost brzine, a *i* indeks u listi (nije potrebno paziti da se dohvaćaju brzine na različitim indeksima).

Potrebno je koristiti *try-catch* blok nad čitavim kodom, te nije potrebno pisati *Using System.IO*.

4. Što će se ispisati u konzolu nakon izvršavanja sljedećeg programskog odsječka. Potrebno je naznačiti redoslijed ispisivanja. Kao točno rješenje priznaje se samo ono u kojem je vidljiv postupak.

```
double[] p = new double[7] { 1.2346, 3.257, 8.3678, 4.233345, 9.7865, 5.68974, 11.4678 };
```

```
int i = 1;
for (i = 0; i < p.Length - 1; i += 2)
{
    Console.WriteLine(Math.Round(p[(i + 1) / 2], 2));
}
Console.WriteLine(p[p.Length - 2]);
```



5. Što će se ispisati u konzolu nakon izvršavanja sljedećeg programskog odsječka. Potrebno je naznačiti redoslijed ispisivanja. Kao točno rješenje priznaje se samo ono u kojem je vidljiv postupak.

```
static void Main(string[] args)
{
    int a = 25 % 4;
    int b = a + 3;
    M1(b, a);
    Console.WriteLine(a + "," + b);
}
static void M1(int a, int b)
{
    a = b;
    Console.WriteLine(a + "," + b);
    M3(a, b);
}
static void M2(int a, int b)
{
    a = 3;
    b = a;
    Console.WriteLine(a + "," + b);
}
static void M3(int a, int b)
{
    Console.WriteLine(b + "," + a);
    Console.WriteLine(b + "," + a++);
}
```



Ispit #68 (17.02.22. - grupa A)

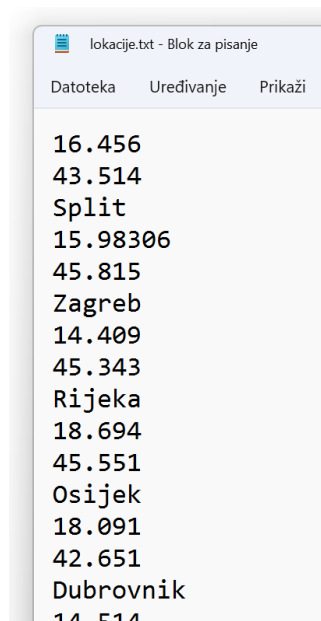
1. Napišite C# kod klase *Lokacija* koja mora imati dva privatna atributa *x* i *y* (brojevi s plivajućim zarezom) koji predstavljaju geografsku lokaciju (dužinu i širinu), te javni atribut *naziv* (niz znakova) koji predstavlja naziv lokacije.

Klasa treba imati konstruktor koji prima sve attribute klase i postavlja ih, te dvije javne metode: *Ispis* i *Udaljenost*.

Metoda *Ispis* ne vraća rezultat i ne prima argumente, a treba u konzolu ispisati podatke instance klase *Lokacija* u formatu: *Lokacija: naziv=NAZIV, x=X i y=Y!* (umjesto znakova *NAZIV*, *X* i *Y* u ispisanom nizu znakova trebaju pisati odgovarajuće vrijednosti brojeva).

Metoda *Udaljenosti* preko argumenta prima (drugu) instancu klase *Lokacija*, te računa i vraća euklidsku udaljenost između te dvije lokacije (instance klase *Lokacija* u kojoj se poziva metoda *Udaljenost* i druge instance klase *Lokacija* koja se predaje kao argument).

2. Napišite C# kod konzolne aplikacije koja učitava sadržaj datoteke *lokacije.txt* (smještene na relativnoj putanji) u kojoj su redoslijedno zapisani podaci *x*, *y* i *naziv*, svaki u zasebnom retku (kako je prikazano na sljedećoj slici) potrebni za instanciranje klase *Lokacija* iz 1. zadatka.



```
lokacije.txt - Blok za pisanje
Datoteka  Uređivanje  Prikaži

16.456
43.514
Split
15.98306
45.815
Zagreb
14.409
45.343
Rijeka
18.694
45.551
Osijek
18.091
42.651
Dubrovnik
14.514
```

U datoteci se nalazi ukupno 27 redaka, odnosno podaci o 9 lokacija. Podatke iz datoteke potrebno je učitati u matricu instanci klase *Lokacija*, koja ima 3 retka i 3 stupca (način i redoslijed učitavanja je proizvoljan).

Neposredno nakon što se učita lokacija potrebno je ispisati podatke koristeći metodu za ispis napisanu u 1. zadatku.

Potrebno je koristiti *try-catch* blok nad čitavim kodom, te nije potrebno pisati *Using System.IO*.



3. Napišite C# kod konzolne aplikacije koja očekuje unos niza znakova: *trapez* ili *sesterokut*.

Ukoliko se za niz znakova unese *trapez*, potrebno je od korisnika zatražiti unos vrijednosti osnovica *a* i *c* i kraka *b* jednakokračnog trapeza (sve brojevi s plivajućim zarezom), te je potom u konzolu potrebno ispisati opseg trapeza u formatu *Opseg jednakokračnog trapeza zadanog stranicama a=A, b=B i c=C iznosi O=o!* (umjesto znakova *A*, *B*, *C* i *o* u ispisanom nizu znakova, trebaju pisati odgovarajuće vrijednosti brojeva).

Ukoliko se za niz znakova unese *sesterokut*, potrebno je od korisnika zatražiti unos stranice *a* pravilnog šesterokuta (cijeli broj), te potom u konzolu ispisati površinu šesterokuta u formatu *Površina pravilnog šesterokuta zadanog stranicom a=A iznosi P=p!* (umjesto znakova *A* i *p* u ispisanom nizu znakova trebaju pisati odgovarajuće vrijednosti brojeva). Površina JEDNOG trokuta unutar pravilnog šesterokuta iznosi $P = \frac{a^2\sqrt{3}}{4}$.

Postupak učitavanja niza znakova treba ponavljati sve dok se ne unese neka vrijednost koja nije niti *trapez* niti *sesterokut*, te je u tom slučaju potrebno ispisati *Nije unesen ispravan niz znakova! Program završava!*.

4. Što će se ispisati u konzolu nakon izvršavanja sljedećeg programskog odsječka. Potrebno je naznačiti redoslijed ispisivanja. Kao točno rješenje priznaje se samo ono u kojem je vidljiv postupak.

```
static void Main(string[] args)
{
    Console.WriteLine(F(3) - 1);
}
private static int F(int n)
{
    if (n < 0)
    {
        return -1;
    }
    else
    {
        int v = F(n - 2) + F(n - 3);
        Console.WriteLine(v);
        return v;
    }
}
```

5. Detaljno opišite liniju po liniju što radi sljedeći programski odsječak.

```
Random rnd = new Random();
int[] p = new int[10] { 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 };
for (int i = 0; i < p.Length; i++)
{
    int v = rnd.Next(0, p.Length);
    Console.WriteLine(p[v]);
}
```



Ispit #69 (26.03.22. - grupa A)

1. Napišite C# kod metode *provjeraID* koja kao argument prima ID vozila (niz znakova) i vraća istinu ili laž ovisno o tome da li je ID ispravan ili ne.

ID je ispravan ako je njegova suma znamenki **IZVAN** intervala [10, 20].

Očekuje se da će se za ID predati niz znamenki kao niz znakova, te to nije potrebno provjeravati.

2. Napišite C# kod konzolne aplikacije koja učitava n i m vrijednosti s konzole koji određuju matricu $A = n \times m$.

S konzole potrebno je popuniti matricu A s ID-evima vozila (niz znakova), na način da se na pojedino mjesto u matrici sprema samo ispravna vrijednost ID-a. Za provjeru ispravnosti ID-a potrebno je koristiti metodu *provjeraID* napisanu u 1. zadatku.

Na kraju je potrebno ispisati sadržaj matrice unazad, od zadnjeg elementa z matrici prema prvom elementu redak po redak, kao u primjer.

Ako se primjerice učitava matrica:

1	3	4	15
18	23	6	999
36	90	81	7
877	123459	44	11

Potrebno je ispisati:

11	44	123459	877
7	81	90	36
999	6	23	18
15	4	3	1

3. Napišite C# kod windows forme koja izračunava broj artikala na polici.

U skladištu ima x polica (cijeli broj) koje se unose *TextBoxom* naziva *txtBrojPolica* i na tim policama ima ukupno n artikala (cijeli broj) koji se unose *TextBoxom* naziva *txtBrojArtikala*.

Potom se svaki dan ukupno sa svih polica potroši y artikala (cijeli broj) koji se unosi *TextBoxom* naziva *txtBrojPotrosenih*, te se na kraju dana ukupno doda z artikala (cijeli broj) koji se unosi *TextBoxom* naziva *txtBrojDodanih*.

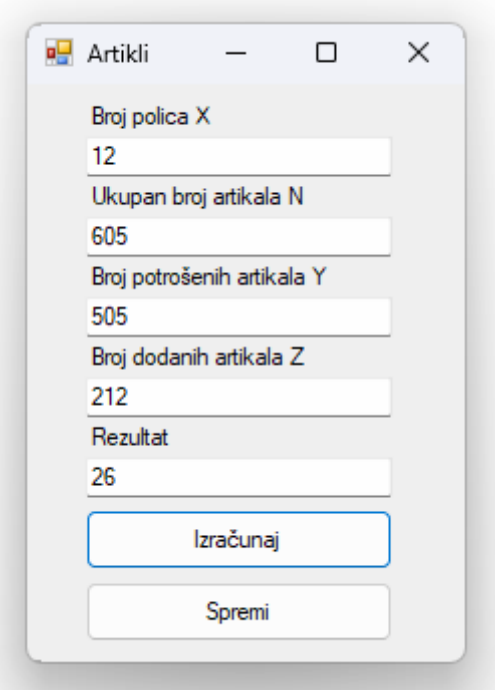
Na pritisak gumba naziva *btnIzracunaj* i teksta *Izracunaj* potrebno je izračunati koliko na kraju dana ima artikala na jednoj polici ako na **svakoj polici ima jednak broj artikala**. Ukoliko rezultat nije cijeli broj (pomoć: upotrijebiti ostatak cjelobrojnog dijeljenja) potrebno je u *TextBox* naziva *txtRezultat* ispisati *Nisu uneseni ispravni brojevi!*, inače ako je rezultat cijeli broj potrebno je u *TextBox* naziva *txtRezultat* ispisati rezultat, te ga dodati u listu rezultata (listu je potrebno definirati na razini klase).

Na kraju je, pritiskom na gumb naziva *btnSpremi* i teksta *Spremi*, potrebno zapisati sve rezultate u listi u tekstualnu datoteku *rezultat.txt* smještenu na relativnoj putanji, redak po redak.

Potrebno je koristiti *try-catch* blok nad kodom za pisanje u datoteku, te nije potrebno pisati *Using System.IO*.



Sučelje je prikazano na sljedećoj slici, s primjerom unesenih vrijednosti koje kao rezultat daju vrijednost 26.



```
public partial class Form1 : Form
{
    //Nadopuni 1

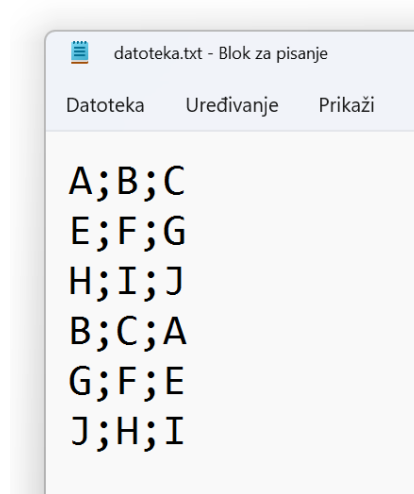
    public Form1()
    {
        InitializeComponent();
    }

    private void btnIzracunaj_Click(object sender, EventArgs e)
    {
        //Nadopuni 2
    }

    private void btnSpremi_Click(object sender, EventArgs e)
    {
        //Nadopuni 3
    }
}
```



4. Što će se ispisati u konzolu nakon izvršavanja sljedećeg programskog odsječka, ukoliko se u datoteci datoteka.txt smještene na relativnoj putanji nalaze vrijednosti prikazane na sljedećoj slici.



Potrebno je naznačiti redoslijed ispisivanja. Kao točno rješenje priznaje se samo ono u kojem je vidljiv postupak.

```
try
{
    StreamReader citanje = new StreamReader("datoteka.txt");
    string linija1 = "";
    int i = -1;
    while (citanje.EndOfStream == false)
    {
        string linija2 = citanje.ReadLine();
        linija1 = citanje.ReadLine();
        string[] dio = linija2.Split(';');
        Console.WriteLine(dio[i + 1]);
        i++;
    }
    citanje.Close();
    string[] dio2 = linija1.Split(';');
    Console.WriteLine(dio2[dio2.Length / 2]);
}
catch (Exception ex)
{
    Console.WriteLine("M");
}
```



5. Što će se ispisati u konzolu nakon izvršavanja sljedećeg programskog odsječka. Potrebno je naznačiti redoslijed ispisivanja. Kao točno rješenje priznaje se samo ono u kojem je vidljiv postupak.

```
static void Main(string[] args)
{
    PravokutanTrokut p = new PravokutanTrokut(10, 12);
    Console.WriteLine(p.izracunajPovrsinu(8, 4));
    p.izracunajPovrsinu();
    p.ispis();
}
class PravokutanTrokut
{
    private double A;
    private double B;
    private double povrsina;
    public PravokutanTrokut(double a, double b)
    {
        A = a;
        B = b;
        povrsina = -1;
        ispis();
    }
    public double izracunajPovrsinu(int a, int b)
    {
        double POVRSINA = (b * a) / 2;
        ispis();
        return POVRSINA;
    }
    public void izracunajPovrsinu()
    {
        povrsina = (A * B) / 2;
        ispis();
    }
    public void ispis()
    {
        Console.WriteLine(povrsina);
    }
}
```



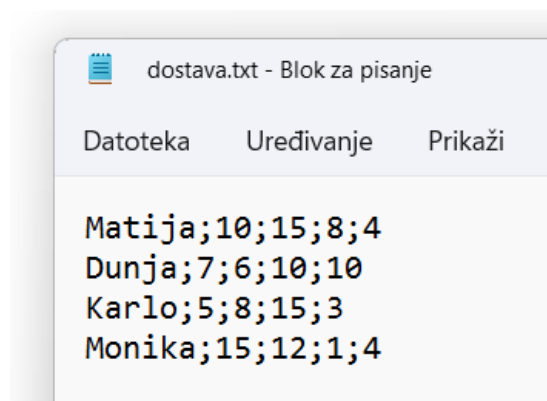
Ispit #70 (23.06.22. - grupa A)

1. Napišite C# kod metode *Zbroj* koja zbraja sve parne prirodne brojeve unutar intervala $[n - m]$.

Metoda prima parametre n i m te vraća rezultat zbrajanja parnih brojeva.

U glavnom dijelu programa učitavaju se granice intervala n i m s tipkovnice te je potrebno napraviti odgovarajući ispis: „Zbroj parnih brojeva unutar intervala n - m je zbroj!“ pri čemu su n i m granice razmatranog intervala, a *zbroj* rezultat zbrajanja metode *Zbroj*.

2. Napišite C# kod konzolne aplikacije koja učitava sadržaj tekstualne datoteke prikazane na sljedećoj slici.



U datoteci su navedeni sljedeći podaci: *ime dostavljača;broj paketa kategorije 1;broj paketa kategorije 2;broj paketa kategorije 3;broj paketa kategorije 4*.

Potrebno je podatke iz tekstualne datoteke učitati u dvodimenzionalno polje te ispisati prosječan broj dostavljenih paketa za svaku kategoriju u formatu „*Prosječan broj dostavljenih paketa kategorije x je prosjek*“, pri čemu je x broj kategorije, a *prosjek* prosječan broj dostavljenih paketa te kategorije.

Potrebno je koristiti *try-catch* blok nad kodom za čitanje iz datoteke, te nije potrebno pisati *Using System.IO*.

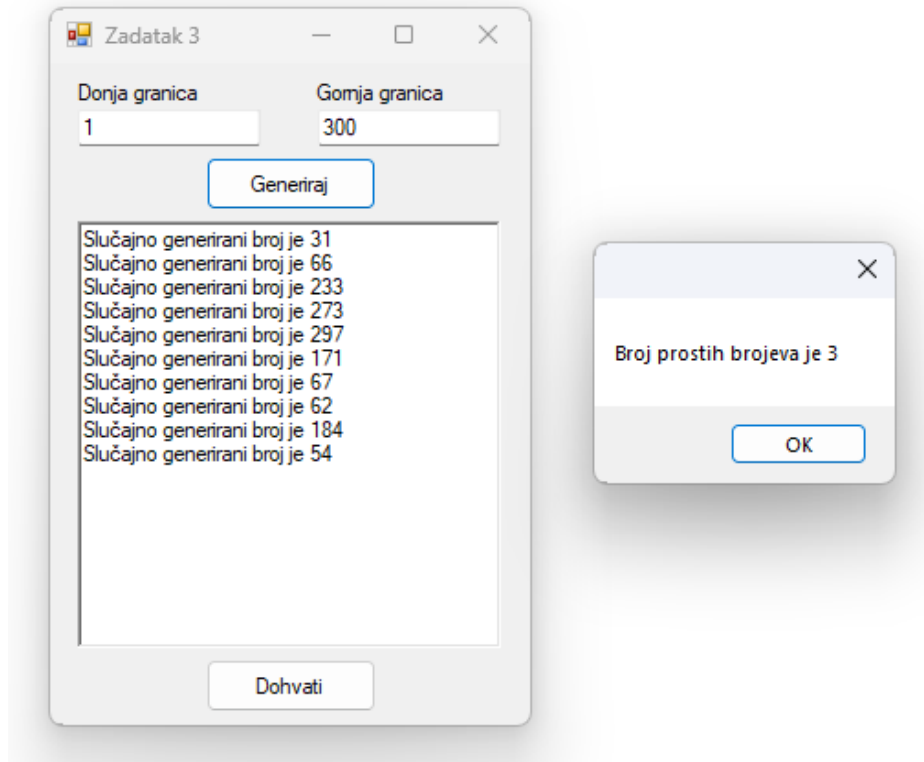
3. Napišite C# kod Windows forme koja generira slučaje brojeve u zadanom intervalu. Slučajan broj generira se u rasponu između donje granice (cijeli broj) koja se unosi *TextBoxom* naziva *txtDG* i gornje granice (cijeli broj) koji se unosi *TextBoxom* naziva *txtGG*.

Na pritisak gumba naziva *btnGeneriraj* i teksta *Generiraj* potrebno je generirati slučajni broj između zadane donje i gornje granice te ga ispisati u *RichTextBoxu* naziva *rtxtIspis* u formatu „*Slučajno generirani broj je x*“ pri čemu je x slučajno generirani broj. Svaki novi generirani slučajni broj (novi pritisak gumba *Generiraj*) prikazuje se u zasebnom retku *RichTextBoxa*.

Dodatno, pritiskom gumba naziva *btnDohvati* i teksta *Dohvati* potrebno je provjeriti koliko slučajnih brojeva (od svih do tada generiranih) je prostih. Format ispisa je proizvoljan. Za provjeru pripada li broj skupu prostih brojeva koriste metodu *ProvjeraProstihBrojeva* koja je napisana i nije ju potrebno pisati.

Sučelje, s primjerom unesenih vrijednosti, je prikazano na sljedećoj slici.





```
public partial class Form1 : Form
{
    //Nadopuni 1

    public Form1()
    {
        InitializeComponent();
    }

    private void btnGeneriraj_Click(object sender, EventArgs e)
    {
        //Nadopuni 2

    }

    private void btnDohvati_Click(object sender, EventArgs e)
    {
        //Nadopuni 3

    }

    static bool ProvjeraProstihBrojeva(int n)
    {
        for (int i = 2; i < n / 2; i++)
        {
            if (n % i == 0)
            {
                return false;
            }
            return true;
        }
    }
}
```



4. Što će se ispisati u konzolu nakon izvršavanja sljedećeg programskog odsječka?

```
static void Main(string[] args)
{
    Console.Write(R(5, 2));
}
static int R(int x, int y)
{
    if (x == 2)
        return y;
    else
    {
        Console.WriteLine(x);
        return R(x - 1, x + y);
    }
}
```

5. Što će se ispisati u konzolu nakon izvršavanja sljedećeg programskog odsječka? Potrebno je naznačiti redoslijed ispisivanja. Kao točno rješenje priznaje se samo ono u kojem je vidljiv postupak i opis zašto je ispisana vrijednost poprimila baš tu vrijednost.

```
static void Main(string[] args)
{
    RasporedVoznje p = new RasporedVoznje("222", new int[] { 10, 12, 14 },
    new int[] { 0, 30, 0, 0, 15, 30 });
    p.izracunajVoznje(p.minutePolaska);
    p.ispis();
}
class RasporedVoznje
{
    private string linija;
    private int[] satiPolaska;
    public int[] minutePolaska;
    public int brojVoznji;
    public RasporedVoznje(string L, int[] sati, int[] minute)
    {
        linija = L;
        satiPolaska = sati;
        minutePolaska = minute;
        ispis();
    }
    public double izracunajVoznje(int[] minute)
    {
        brojVoznji = minute.Length;
        return brojVoznji;
    }
    public void ispis()
    {
        Console.WriteLine(brojVoznji);
    }
}
```



Ispit #71 (23.06.22. - grupa B)

1. Napišite C# kod metode *ZamijeniNeparne* koja sve neparne elemente polja zamijeni s njegovim susjednim većim parnim elementom. Primjerice, ako je element polja 3 metoda će na to mjesto u polje pohraniti vrijednost 4.

Metoda kao parametar prima polje cijelih brojeva te vraća koliko neparnih elemenata je zamijenila s parnim.

U glavnom dijelu programa potrebno je: učitati s tipkovnice vrijednost n koja predstavlja veličinu polja, polje veličine n popuniti slučajnim brojevima u intervalu od $[0, 1000]$ te ispisati broj učinjenih zamjena u proizvoljnom formatu pri čemu je broj zamjena rezultat metode *ZamijeniNeparne*.

2. Napišite C# kod konzolne aplikacije koja popunjava matricu cijelih brojeva s radnim satima zaposlenika. Matrica sadrži 5 redaka i 7 stupaca. Redak se odnosi na pojedinog zaposlenika, a sedam stupaca za sedam dana u tjednu. Matricu je potrebno popuniti s konzole.

Nakon toga se za svaki dan u matrici (svaki stupac matrice) izračunava ukupan broj odrađenih sati za sve zaposlenike te se bilježi onaj dan na koji je odrađeno najviše sati. Popunjenu matricu te bilješku o danu s najvećim brojem sati u formatu: „Dan s najviše odrađenih sati je x “ pri čemu je x redni broj dana, potrebno je zapisati u datoteku *brojSati.txt* u formatu kako je prikazano je u primjeru.

Porebno je koristiti *try-catch* blok nad kodom za pisanje u datoteku, te nije potrebno pisati *Using System.IO*.

Primjerice, ako se učitava matrica

5	8	2	5	2	8	1
1	6	9	1	5	5	7
2	9	3	4	8	7	3
3	2	7	3	1	2	1
5	3	3	5	6	4	4

u datoteku je potrebno zapisati

```
brojDostava.txt - Blok za pisanje
Datoteka   Uređivanje   Prikaži
5 8 2 5 2 8 1
1 6 9 1 5 5 7
2 9 3 4 8 7 3
3 2 7 3 1 2 1
5 3 3 5 6 4 4
Dan s najviše odrađenih sati je 2
```

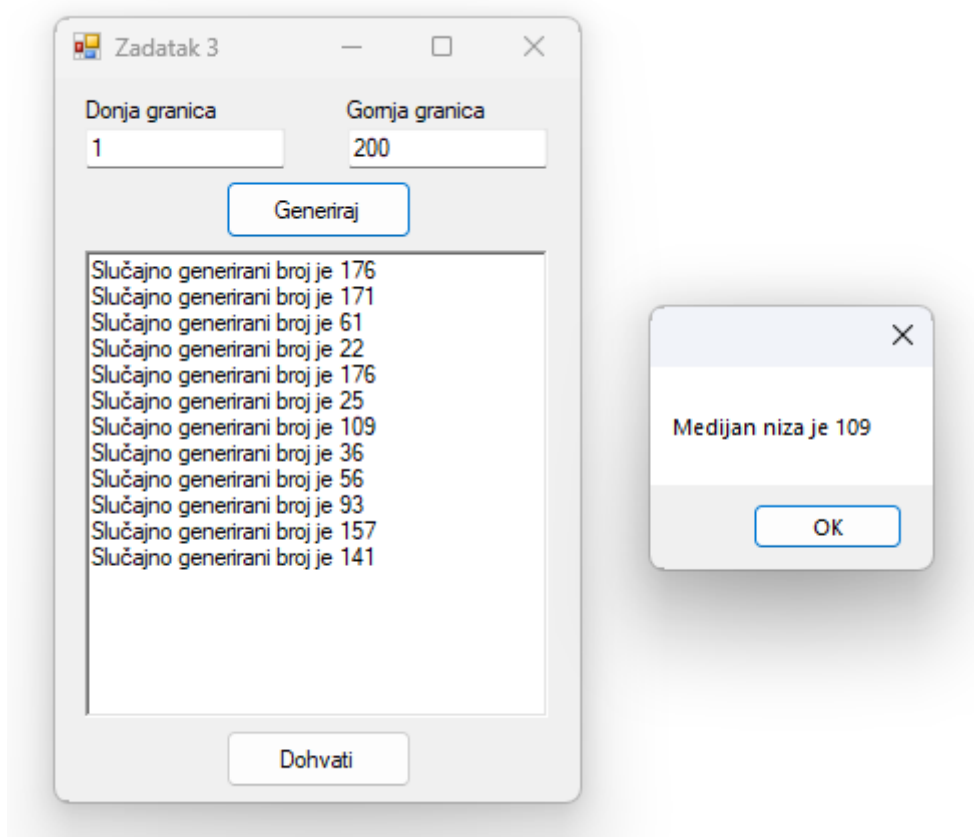


3. Napišite C# kod Windows forme koja generira slučajne brojeve u zadanom intervalu. Slučajan broj generira se u rasponu između donje granice (cijeli broj) koja se unosi *TextBoxom* naziva *txtDG* i gornje granice (cijeli broj) koja se unosi *TextBoxom* naziva *txtGG*.

Na pritisak gumba naziva *btnGeneriraj* i teksta *Generiraj* potrebno je generirati slučajni broj između zadane donje i gornje granice te ga ispisati u *RichTextBoxu* naziva *rtxtIspis* u formatu „Slučajno generirani broj je *x*“ pri čemu je *x* slučajno generirani broj. Svaki novi generirani slučajni broj (novi pritisak gumba *Generiraj*) prikazuje se u zasebnom retku *RichTextBoxa*.

Dodatno, pritiskom gumba naziva *btnDohvati* i teksta *Dohvati* svi do tada generirani slučajni brojevi se sortiraju od najmanjeg prema najvećem (za sortiranje koristite napisanu metodu *InsertionSort*) te se u skočnom prozoru ispisuje središnji element u sortiranom nizu brojeva.

Sučelje, s primjerom unesenih vrijednosti, prikazano je sljedećom slikom.



```

public partial class Form1 : Form
{
    //Nadopuni 1

    public Form1()
    {
        InitializeComponent();
    }

    private void btnGeneriraj_Click(object sender, EventArgs e)
    {
        //Nadopuni 2

    }

    private void btnDohvati_Click(object sender, EventArgs e)
    {
        //Nadopuni 3

    }

    private static void InsertionSort(List<int> lista)
    {
        int n = lista.Count;
        for (int i = 1; i < n; i++)
        {
            int temp = lista[i];
            int j;
            for (j = i; j >= 1 && lista[j - 1] > temp; j--)
            {
                lista[j] = lista[j - 1];
            }
            lista[j] = temp;
        }
    }
}

```



4. Što će se ispisati u konzolu nakon izvršavanja sljedećeg programskog odsječka?

```
static void Main(string[] args)
{
    Console.Write(R(7, 5));
}
static int R(int x, int y)
{
    if (x == 3)
        return y;
    else
    {
        Console.WriteLine(x);
        return R(x - 1, x + y);
    }
}
```

5. Što će se ispisati u konzolu nakon izvršavanja sljedećeg programskog odsječka? Potrebno je naznačiti redoslijed ispisivanja. Kao točno rješenje priznaje se samo ono u kojem je vidljiv postupak i opis zašto je ispisana vrijednost poprimila baš tu vrijednost.

```
static void Main(string[] args)
{
    Trajektorija p = new Trajektorija("1", "Automobil");
    p.PromijeniMod("Pješak");
    p.DodajBrzinu(5);
    p.ispis();
}
class Trajektorija
{
    private string trajektorija;
    private List<double> brzina;
    public List<string> modPrijevoza;
    public Trajektorija(string t, string mod)
    {
        trajektorija = t;
        brzina = new List<double>();
        modPrijevoza = new List<string>();
        modPrijevoza.Add(mod);
        DodajBrzinu(80);
    }
    public void DodajBrzinu(double v)
    {
        brzina.Add(v);
        for (int i = 0; i < brzina.Count; i++)
        {
            Console.WriteLine(brzina[i]);
        }
    }
    public void PromijeniMod(string mod)
    {
        modPrijevoza.Add(mod);
    }
    public void ispis()
    {
        Console.WriteLine("Broj zapisa u trajektoriji je " + brzina.Count);
    }
}
```



Ispit #72 (07.07.22. - grupa A)

1. Napišite C# kod klase *Automobil* koja mora imati tri privatna atributa: model, registracijsku oznaku (oba su niz znakova) i starost vozila (cijeli broj) te javi atribut *boja* (niz znakova).

Klasa treba imati i konstruktor koji prima sve attribute klase i postavlja im vrijednosti, te dvije javne metode: *ispis* i *azuriraj*.

Metoda *ispis* ne vraća rezultat i ne prima argumente, a treba u konzolu ispisati podatke instance klase *Automobil* u formatu: *Automobil modela=MODEL i registracijske oznake=REGISTRACIJSKA OZNAKA star je STAROST godina!* (umjesto znakova *MODEL*, *REGISTRACIJSKA OZNAKA* i *STAROST* trebaju pisati odgovarajuće vrijednosti atributa).

Metoda *azuriraj* preko argumenata prima cijeli broj, te povećava starost automobilu trenutne instance za primljenu cjelobrojnu vrijednost.

2. Napišite C# kod konzolne aplikacije koja učitava matricu slučajnih brojeva s plivajućim zarezom te jednodimenzionalno polje slučajnih cijelih brojeva.

Potrebno je preko tipkovnice unijeti vrijednost n i m , potom kreirati matricu veličine $M = n \times m$ te polje P veličine m . Matricu M potrebno je popuniti slučajnim brojevima s plivajućim zarezom u intervalu $[1,100]$, a polje P slučajnim cijelim brojevima u intervalu $[0, 50]$.

U polju P potrebno je pronaći indeks elementa čija je vrijednost najveća, te potom u matrici M na tom pronađenom indeksu stupca, svim elementima u stupcu promijeniti vrijednost na korijen trenutne vrijednosti pojedinog elementa u stupcu. Ukoliko polje P sadrži više elemenata iste najveće vrijednosti odabire se element čiji je indeks najveći.

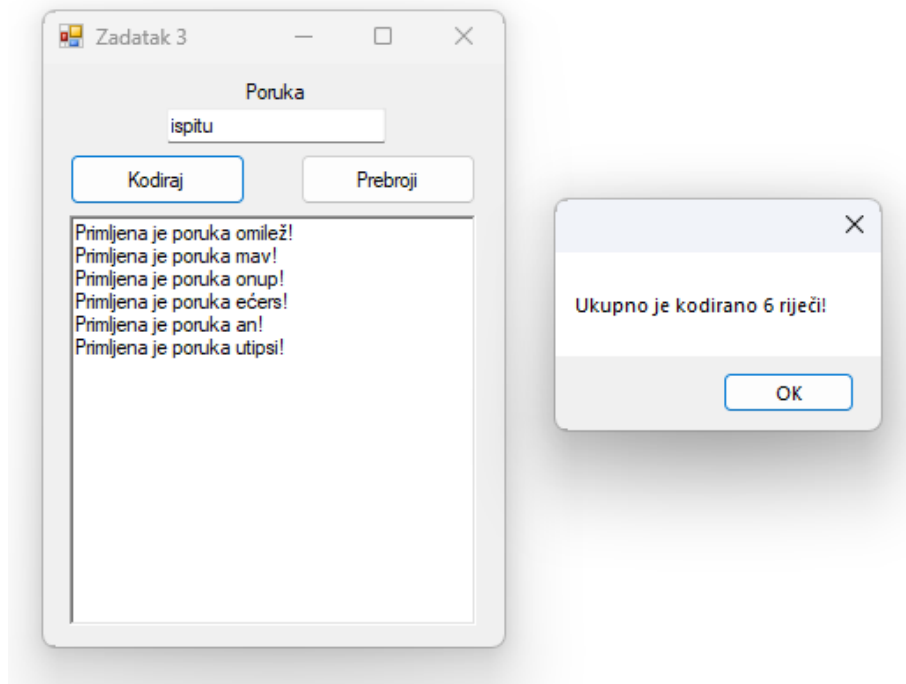
3. Napišite C# kod Windows forme koja prikazuje upisanu poruku (znakovni niz) obrnutim redoslijedom znakova.

Korisnik unosi poruku u *TextBox* naziva *txtPoruka*. Na pritisak gumba naziva *btnKodiraj* i teksta *Kodiraj* potrebno je ispisati u *RichTextBoxu* naziva *rtxtIspis* kodiranu poruku u formatu "*Primljena je poruka x!*" pri čemu je x upisana poruka s obrnutim redoslijedom znakova. Svaka nova poruka (novi pritisak gumba *Kodiraj*) prikazuje se u zasebnom retku *RichTextBoxa*.

Dodatno, pritiskom gumba naziva *btnPrebroji* i teksta *Prebroji* potrebno je u skočnom prozoru ispisati broj do tada kodiranih riječi. Format ispisa je proizvoljan.

Sučelje, s primjerom unesenih vrijednosti, prikazano je sljedećom slikom.





```
public partial class Form1 : Form
{
    //Nadopuni 1

    public Form1()
    {
        InitializeComponent();
    }

    private void btnKodiraj_Click(object sender, EventArgs e)
    {
        //Nadopuni 2

    }

    private void btnPrebroji_Click(object sender, EventArgs e)
    {
        //Nadopuni 3

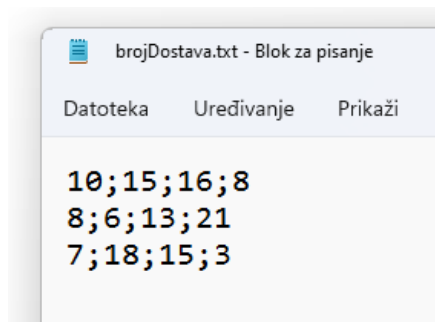
    }
}
```



4. Što će se ispisati u konzolu nakon izvršavanja sljedećeg programskog odsječka?

```
static void Main(string[] args)
{
    int[,] matrica = new int[4, 3] { { 1, 2, 3 }, { 3, 4, 2 }, { 5, 6, 7 },
    { 1, 2, 3 } };
    for (int j = matrica.GetLength(1) + 1; j > 2; j--)
    {
        for (int i = -1; i <= matrica.GetLength(0) - 2; i++)
        {
            if (i + 2 == j - 1)
            {
                Console.WriteLine(matrica[i + 1, j - 2]);
            }
        }
    }
    Console.WriteLine(matrica.Length);
}
```

5. Što će se ispisati u konzolu nakon izvršavanja sljedećeg programskog odsječka? Sadržaj datoteke koja se učitava prikazan je sljedećom slikom.



Potrebno je naznačiti i redoslijed ispisivanja. Kao točno rješenje priznaje se samo ono u kojem je vidljiv postupak i opis zašto je ispisana vrijednost poprimila baš tu vrijednost.

```
static void Main(string[] args)
{
    StreamReader citaj = new StreamReader("brojDostava.txt");
    while (!citaj.EndOfStream)
    {
        string linija = citaj.ReadLine();
        string[] dijeloviLinije = linija.Split(';');
        int max = int.MinValue;
        for (int i = 0; i < dijeloviLinije.Length; i++)
        {
            if (Convert.ToInt32(dijeloviLinije[i]) > max)
                max = Convert.ToInt32(dijeloviLinije[i]);
        }
        Console.WriteLine(max % 2);
    }
    Console.WriteLine(citaj.EndOfStream);
    citaj.Close();
}
```



Ispit #73 (07.07.22. - grupa B)

1. Napišite C# kod klase *Prometnica* koja mora imati pet privatnih atributa: niz znakova kojim se bilježi kategorija prometnice te četiri broja s plivajućim zarezom kojima se bilježe geografske koordinate početne i krajnje točke prometnice (*xPocetak*, *yPocetak*, *xKraj*, *yKraj*).

Klasa treba imati konstruktor koji prima sve attribute klase i postavlja im vrijednosti, te dvije javne metode: *ispis* i *izracunajDuzinu*.

Metoda *izracunajDuzinu* ne prima argumente, ali izračunava i vraća Euklidsku udaljenost između početne i krajnje točke trenutne instance prometnice, tj. izračunava dužinu prometnice prema sljedećem izrazu.

$$udaljenost(T_1, T_2) = \sqrt{(x_{T_2} - x_{T_1})^2 + (y_{T_2} - y_{T_1})^2}$$

Metoda *ispis* ne vraća rezultat i ne prima argumente, a treba u konzolu ispisati dužinu prometnice u proizvoljnom formatu.

2. Napišite C# kod konzolne aplikacije koja učitava matricu slučajnih brojeva s plivajućim zarezom te jednodimenzionalno polje slučajnih cijelih brojeva.

Potrebno je preko tipkovnice unijeti vrijednost *n* i *m*, potom kreirati matricu veličine $M = n \times m$ te polje *P* veličine *n*.

Matricu *M* potrebno je popuniti slučajnim brojevima s plivajućim zarezom u intervalu [100, 200], a polje *P* slučajnim cijelim brojevima u intervalu [150,200).

U polju *P* potrebno je pronaći indeks elementa čija je vrijednost najmanja, te potom u matrici *M* u retku koji se nalazi na pronađenom indeksu svim elementima promijeniti vrijednost na kvadrat trenutne vrijednosti pojedinog elementa u retku. Ukoliko polje *P* sadrži više elemenata iste najmanje vrijednosti odabire se element čiji je indeks najveći.

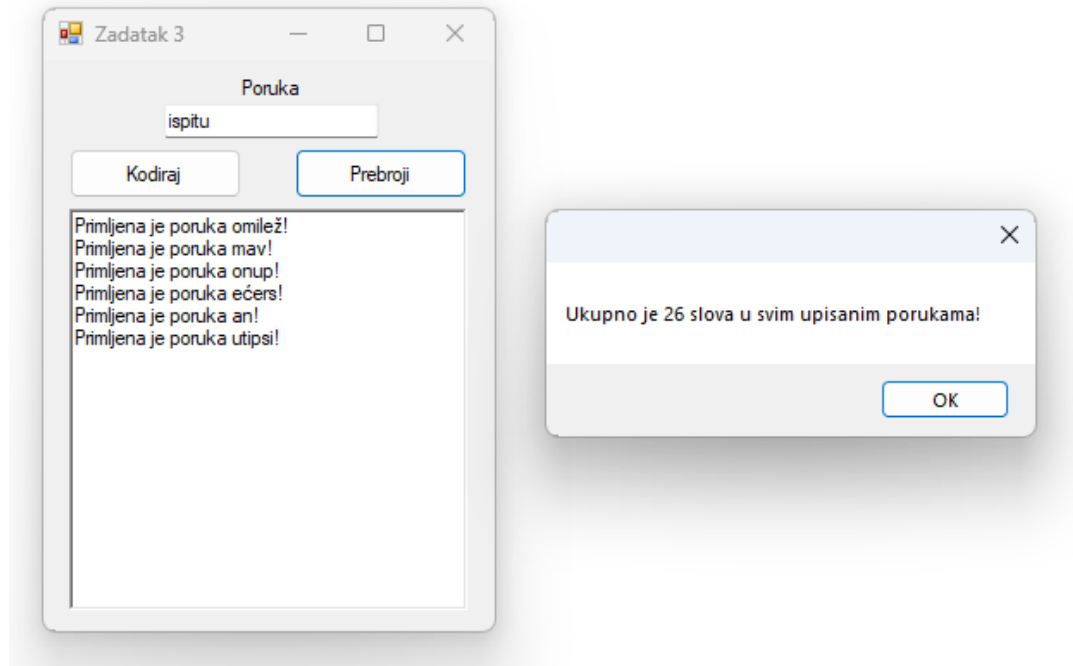
3. Napišite C# kod Windows forme koja prikazuje upisanu poruku (znakovni niz) obrnutim redoslijedom znakova.

Korisnik unosi poruku u *TextBox* naziva *txtPoruka*. Na pritisak gumba naziva *btnKodiraj* i teksta *Kodiraj* potrebno je ispisati u *RichTextBoxu* naziva *rtxtIspis* kodiranu poruku u formatu "*Primljena je poruka x!*" pri čemu je *x* upisana poruka s obrnutim redoslijedom znakova. Svaka nova poruka (novi pritisak gumba *Kodiraj*) prikazuje se u zasebnom retku *RichTextBoxa*.

Dodatno, pritiskom gumba naziva *btnPrebroji* i teksta *Prebroji* potrebno je u skočnom prozoru ispisati broj ukupnog broja slova u svim riječima nad kojima je izvršena zamjena. Format ispisa je proizvoljan.

Sučelje, s primjerom unesenih vrijednosti, prikazano je na sljedećoj slici.





```
public partial class Form1 : Form
{
    //Nadopuni 1

    public Form1()
    {
        InitializeComponent();
    }

    private void btnKodiraj_Click(object sender, EventArgs e)
    {
        //Nadopuni 2
    }

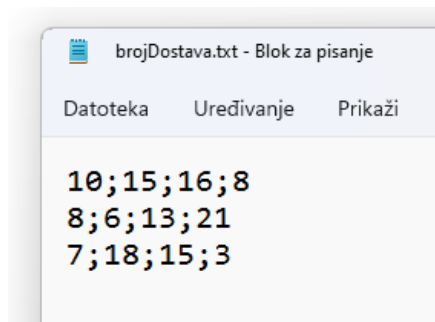
    private void btnPrebroji_Click(object sender, EventArgs e)
    {
        //Nadopuni 3
    }
}
```



4. Što će se ispisati u konzolu nakon izvršavanja sljedećeg programskog odsječka?

```
static void Main(string[] args)
{
    int[,] matrica = new int[4, 3] { { 7, 8, 9 }, { 10, 6, 12 }, { 3, 6, 8 },
    { 1, 2, 3 } };
    for (int j = matrica.GetLength(1) + 1; j > 2; j--)
    {
        for (int i = 0; i <= matrica.GetLength(0) - 2; i++)
        {
            if (i + 1 == j - 2)
            {
                Console.WriteLine(matrica[i + 2, j - 2]);
            }
        }
    }
    Console.WriteLine(matrica.Length + 3);
}
```

5. Što će se ispisati u konzolu nakon izvršavanja sljedećeg programskog odsječka? Sadržaj datoteke koja se učitava prikazan je sljedećom slikom.



Potrebno je naznačiti i redoslijed ispisivanja. Kao točno rješenje priznaje se samo ono u kojem je vidljiv postupak i opis zašto je ispisana vrijednost poprimila baš tu vrijednost.

```
static void Main(string[] args)
{
    StreamReader citaj = new StreamReader("brojDostava.txt");
    while (!citaj.EndOfStream)
    {
        string linija = citaj.ReadLine();
        string[] dijeloviLinije = linija.Split(';');
        int min = int.MaxValue;
        for (int i = 0; i < dijeloviLinije.Length; i++)
        {
            if (Convert.ToInt32(dijeloviLinije[i]) < min)
                min = Convert.ToInt32(dijeloviLinije[i]);
        }
        Console.WriteLine(min % 2);
    }
    Console.WriteLine(!citaj.EndOfStream);
    citaj.Close();
}
```



Ispit #74 (01.09.22. - grupa A)

1. Napišite C# kod metode *Spajanje* koja kao argument prima dva polja niza znakova riječi i zabrane.

Program treba spojiti sve validne "riječi" (nizove znakova) iz polja riječi u jednu rečenicu s razmacima između pojedinih riječi. Riječ je validna za spajanje ako se ne pojavljuje u polju zabrane.

Na kraju metoda treba u konzolu ispisati cijelu spojenu rečenicu (format ispisa treba biti kao u primjeru).

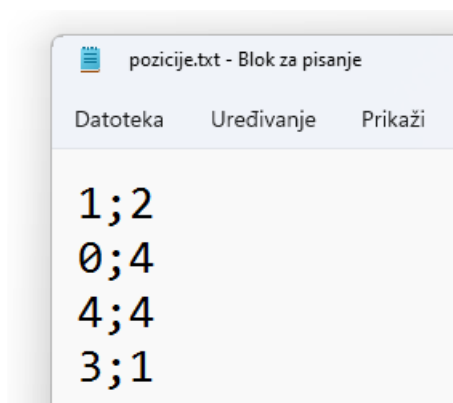
Dodatno metoda treba i vratiti spojenu rečenicu (niz znakova).

Primjerice neka polje riječi ima vrijednosti *Marko, ne, voli, :, programiranje i !*, a polje zabrane vrijednosti *ne i ;*, u konzolu se treba ispisati *Spojena rečenica glasi: Marko voli programiranje!*.

2. Napišite C# kod konzolne aplikacije koja postavlja vrijednosti varijabli *n* i *m* na slučajne vrijednosti iz intervala [5, 8], te potom kreira matricu $M^{n \times m}$.

Matricu *M* potrebno je popuniti slučajnim cijelim brojevima iz intervala [15, 40].

Potom je potrebno ispisati sadržaj matrice na točno određenim pozicijama u matrici u formatu *Vrijednost matrice M na indeksu retka IR i indeksu stupca IS iznosi V!* (umjesto znakova *IR, IS* i *V* trebaju pisati odgovarajuće vrijednosti). Pozicije su zapisane u datoteci *pozicije.txt* smještene na relativnoj putanji u formatu *indeksRetka;indeksStupca*, svaka u zasebnom retku datoteke. Primjer pozicija u datoteci prikazan je na sljedećoj slici.



Potrebno je koristiti *try-catch* blok nad dijelom koda za rad s datotekom, te nije potrebno pisati *Using System.IO*.



3. Što će se ispisati u konzolu nakon izvršavanja sljedećeg programskog odsječka?

```
int[, ] X = new int[5, 5] { { 1, 4, 5, 6, 9 },
                             { 3, 8, 4, 6, 1 },
                             { -5, 11, 8, 18, 4 },
                             { 7, -8, 10, 15, -12 },
                             { 12, 21, 37, 89, 36 } };

int x = 4;
int y = x - 4;
for (int j = x; j >= 1; j--)
{
    int v = 0;
    for (int i = y; i <= 4; i++)
    {
        if (X[i, j] % 3 == 0 ^ X[i, j] % 5 == 0)
        {
            v++;
        }
    }
    Console.WriteLine(v);
}
```

4. Što će se ispisati u konzolu nakon izvršavanja sljedećeg programskog odsječka?

```
string s1 = " Marko ";
string s2 = " Mirko ";
string s = s1 + " " + s2;
int x1 = 0;
int x2 = 0;
int x3 = 0;
int x4 = 0;
try
{
    for (int i = 0; i <= s.Length; i++)
    {
        if (s[i] == ' ')
        {
            x1++;
        }
        else if (s[i] == 'O')
        {
            x2++;
        }
        if (s[i] == 'M')
        {
            x3++;
        }
        else
        {
            x4++;
        }
    }
}
catch (Exception ex)
{
    x1++;
}
Console.WriteLine(x1);
Console.WriteLine(x2);
Console.WriteLine(x3);
Console.WriteLine(x4);
```



5. Napišite C# kod konzolne aplikacije koja učitava cijeli broj n s tipkovnice, te potom u listu taksija dodaje n instanci klase *Taxi* pri čemu se vrijednosti atributa pojedine instance učitavaju s konzole.

Nakon učitavanja taksija u listu, potrebno je pronaći taksi koji je najudaljeniji poziciji (0, 0) i ispisati njegovu registraciju u konzolu.

Za izračun udaljenosti koristiti Euklidsku udaljenost danu sljedećim izrazom.

$$D(T_1, T_2) = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

```
class Taxi
{
    public string REGISTRACIJA;
    public string VOZAC;
    private double POZICIJA_X;
    private double POZICIJA_Y;
    public Taxi(string registracija, string vozac, double pozicija_x,
        double pozicija_y)
    {
        REGISTRACIJA = registracija;
        VOZAC = vozac;
        POZICIJA_X = pozicija_x;
        POZICIJA_Y = pozicija_y;
    }
    public void postaviX(double x)
    {
        POZICIJA_X = x;
    }
    public void postaviY(double y)
    {
        POZICIJA_Y = y;
    }
    public double dohvatiX()
    {
        return POZICIJA_X;
    }
    public double dohvatiY()
    {
        return POZICIJA_Y;
    }
}
```



Ispit #75 (01.09.22. - grupa B)

1. Napišite C# kod metode *Mnozenje* koja kao argument prima dva polja cijelih brojeva *brojevi* i *dozvoljeni*.

Program treba izračunati umnožak svih validnih brojeva iz polja *brojevi*. Broj je validan ako se pojavljuje u polju *dozvoljeni*.

Na kraju metoda treba u konzolu ispisati umnožak svih validnih brojeva (format ispisa treba biti kao u primjeru).

Dodatno metoda treba i vratiti iznos umnoška. Primjerice neka polje *brojevi* ima vrijednosti 4, 5, 4, 2, 8 i 4, a polje *dozvoljeni* vrijednosti 4, 9 i 2, u konzolu se treba ispisati *Umnožak validnih brojeva iznosi: 128!*.

2. Napišite C# kod konzolne aplikacije koja vrijednosti varijabli n i m učitava s tipkovnice, te potom kreira matricu $M^{n \times m}$.

Matricu M potrebno je popuniti slučajnim brojevima s plivajućim zarezom iz intervala $[0, 50>$.

Potom je potrebno ispisati sadržaj matrice na točno određenim stupcima u matrici u formatu *Vrijednosti matrice M na indeksu stupca IS su: v1, v2,..., vn!* (umjesto znakova *IS* i $v1, v2, \dots, vn$ trebaju pisati odgovarajuće vrijednosti). Indeksi stupaca su zapisani u datoteci *stupci.txt* smještene na relativnoj putanji, svaki u indeksu u zasebnom retku datoteke. Primjer indeksa stupaca u datoteci prikazan je na sljedećoj slici.



Potrebno je koristiti *try-catch* blok nad dijelom koda za rad s datotekom, te nije potrebno pisati *Using System.IO*.



3. Što će se ispisati u konzolu nakon izvršavanja sljedećeg programskog odsječka?

```
int[, ] X = new int[5, 5] { { 1, 4, 5, 6, 9 },
                             { 3, 8, 4, 6, 1 },
                             { -4, 12, 15, 17, 0 },
                             { 0, -7, 11, 16, -15 },
                             { 12, 21, 37, 89, 36 } };

int x = 4;
int y = x - 4;
for (int j = x; j >= 1; j--)
{
    int v = 0;
    for (int i = y; i <= 4; i++)
    {
        if (X[i, j] % 3 == 0 ^ X[i, j] % 5 == 0)
        {
            v++;
        }
    }
    Console.WriteLine(v);
}
```

4. Što će se ispisati u konzolu nakon izvršavanja sljedećeg programskog odsječka?

```
string s1 = " Tino ";
string s2 = " Tomo ";
string s = s1 + " " + s2;
int x1 = 0;
int x2 = 0;
int x3 = 0;
int x4 = 0;
try
{
    for (int i = 0; i <= s.Length; i++)
    {
        if (s[i] == ' ')
        {
            x1++;
        }
        else if (s[i] == 't')
        {
            x2++;
        }
        if (s[i] == 'M')
        {
            x3++;
        }
        else
        {
            x4++;
        }
    }
}
catch (Exception ex)
{
    x2++;
}
Console.WriteLine(x1);
Console.WriteLine(x2);
Console.WriteLine(x3);
Console.WriteLine(x4);
```



5. Napišite C# kod konzolne aplikacije koja učitava cijeli broj n s tipkovnice, te potom u listu taksija dodaje n instanci klase *Taxi* pri čemu se vrijednosti atributa pojedine instance učitavaju s konzole.

Nakon učitavanja taksija u listu, potrebno je pronaći taksi koji je najbliži poziciji (100, 100) i ispisati njegovog vozača u konzolu.

Za izračun udaljenosti koristiti Euklidsku udaljenost danu sljedećim izrazom.

$$D(T_1, T_2) = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

```
class Taxi
{
    public string REGISTRACIJA;
    public string VOZAC;
    private double POZICIJA_X;
    private double POZICIJA_Y;
    public Taxi(string registracija, string vozac, double pozicija_x,
double pozicija_y)
    {
        REGISTRACIJA = registracija;
        VOZAC = vozac;
        POZICIJA_X = pozicija_x;
        POZICIJA_Y = pozicija_y;
    }
    public void postaviX(double x)
    {
        POZICIJA_X = x;
    }
    public void postaviY(double y)
    {
        POZICIJA_Y = y;
    }
    public double dohvatiX()
    {
        return POZICIJA_X;
    }
    public double dohvatiY()
    {
        return POZICIJA_Y;
    }
}
```



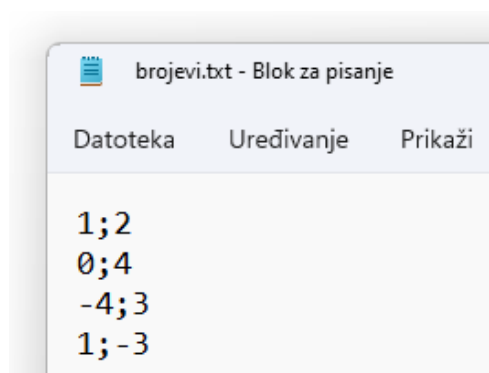
Ispit #76 (08.09.22. - grupa A)

Napomena: rok nije kompletan (nedostaju zadaci 1, 4 i 5)

2. Napišite C# kod konzolne aplikacije koja učitava vrijednosti iz datoteke *brojevi.txt* u listu cijelih brojeva.

Potom se izračunava i ispisuje na konzolu prosječna vrijednost svih pozitivnih elemenata liste. Ispis treba biti u formatu *Prosječna vrijednost x pozitivnih elemenata liste je prosjek*, pri čemu je x broj pronađenih pozitivnih vrijednosti u listi, a *prosjek* pronađena izračunata prosječna vrijednost.

Datoteka *brojevi.txt* smještena je na relativnoj putanji. Format zapisa u datoteci je *prvaVrijednost;drugaVrijednost*, svaka u zasebnom retku. Primjer sadržaja datoteke prikazan je na sljedećoj slici.



Potrebno je koristiti *try-catch* blok nad dijelom koda za rad s datotekom, te nije potrebno pisati *Using System.IO*.

3. Što će se ispisati u konzolu nakon izvršavanja sljedećeg programskog odsječka?

```
int a = 7 % 4;
int b = 5 + a;
int c = 0;
int d = 1;

do
{
    if (a >= 2 && (b != 0 ^ d == 1))
    {
        c -= a;
        d = 2 * 10;
        b -= b;
    }
    else
    {
        c += a;
        d = a * 3 + 2;
        a--;
    }
    Console.WriteLine(a + "," + b + "," + c + "," + d);
} while (a + b != 10);
```



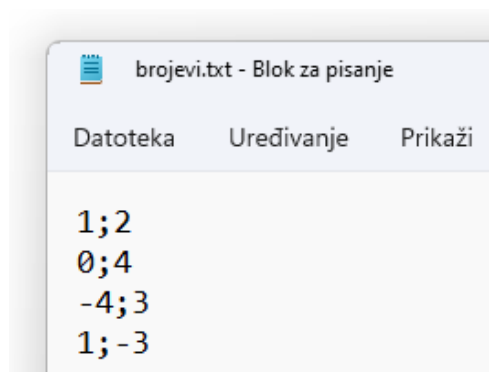
Ispit #77 (08.09.22. - grupa B)

1. Napišite C# kod metode *IspisiRetke* koja kao argument prima dvodimenzionalnu matricu cijelih brojeva *Z* i jednodimenzionalno polje cijelih brojeva *redci*. Metoda ispisuje elemente matrice *Z* na indeksu retka čija je vrijednost jednaka elementima polja *redci*.

2. Napišite C# kod konzolne aplikacije koja učitava vrijednosti iz datoteke *brojevi.txt* u listu cijelih brojeva.

Potom se izračunava i ispisuje na konzolu najmanja vrijednost svih parnih elemenata liste. Ispis treba biti u formatu *Najmanja vrijednost x parnih elemenata liste je najmanjaVrijednost*, pri čemu je *x* broj pronađenih parnih vrijednosti u listi, a *najmanjaVrijednost* pronađena najmanja vrijednost.

Datoteka *brojevi.txt* smještena je na relativnoj putanji. Format zapisa u datoteci je *prvaVrijednost;drugaVrijednost*, svaka u zasebnom retku. Primjer sadržaja datoteke prikazan je na sljedećoj slici.



Potrebno je koristiti *try-catch* blok nad dijelom koda za rad s datotekom, te nije potrebno pisati *Using System.IO*.



3. Što će se ispisati u konzolu nakon izvršavanja sljedećeg programskog odsječka?

```
int a = 10 % 2;
int b = 3 + a;
int c = 1;
int d = 0;

do
{
    if (a >= 2 && (b != 0 ^ d == 1))
    {
        c -= a;
        d = 2 * 10;
        b -= b;
    }
    else
    {
        c += a;
        d = a * 3 + 2;
        a--;
    }
    Console.WriteLine(a + "," + b + "," + c + "," + d);
} while (c + d == 1);
```



4. U zadatku je potrebno naznačiti redoslijed i vrijednosti koje se ispisuju. Kao točno rješenje priznaje se samo ono u kojem je vidljiv postupak i opis zašto je ispisana vrijednost poprimila baš tu vrijednost.

```
static void Main(string[] args)
{
    int n = 2;
    string vozac = "";
    double x = 0;
    double y = 0;
    for (int i = 0; i < n; i++)
    {
        vozac = "Z";
        x = n - i;
        y = n + i;
    }
    Taxi taksi = new Taxi(vozac, x, y);
    taksi.povecajX();
    Console.WriteLine("Pozicija x taksi je " + taksi.POZICIJA_X + "!");
    Console.WriteLine("Pozicija y taksi je " + taksi.POZICIJA_Y + "!");
}

class Taxi
{
    public string VOZAC;
    public double POZICIJA_X;
    public double POZICIJA_Y;
    public Taxi(string vozac, double pozicija_x, double pozicija_y)
    {
        VOZAC = vozac;
        POZICIJA_X = pozicija_x;
        POZICIJA_Y = pozicija_y;
        postaviX(pozicija_x + 3);
        postaviY(pozicija_y + 2);
    }
    public void postaviX(double x)
    {
        Console.WriteLine(POZICIJA_X);
        POZICIJA_X = x;
    }
    public void postaviY(double y)
    {
        POZICIJA_Y = y;
        Console.WriteLine(POZICIJA_Y);
    }
    public void povecajX()
    {
        POZICIJA_X = POZICIJA_X + 2;
    }
    public void povecajY()
    {
        POZICIJA_Y = POZICIJA_Y + 1;
    }
}
```

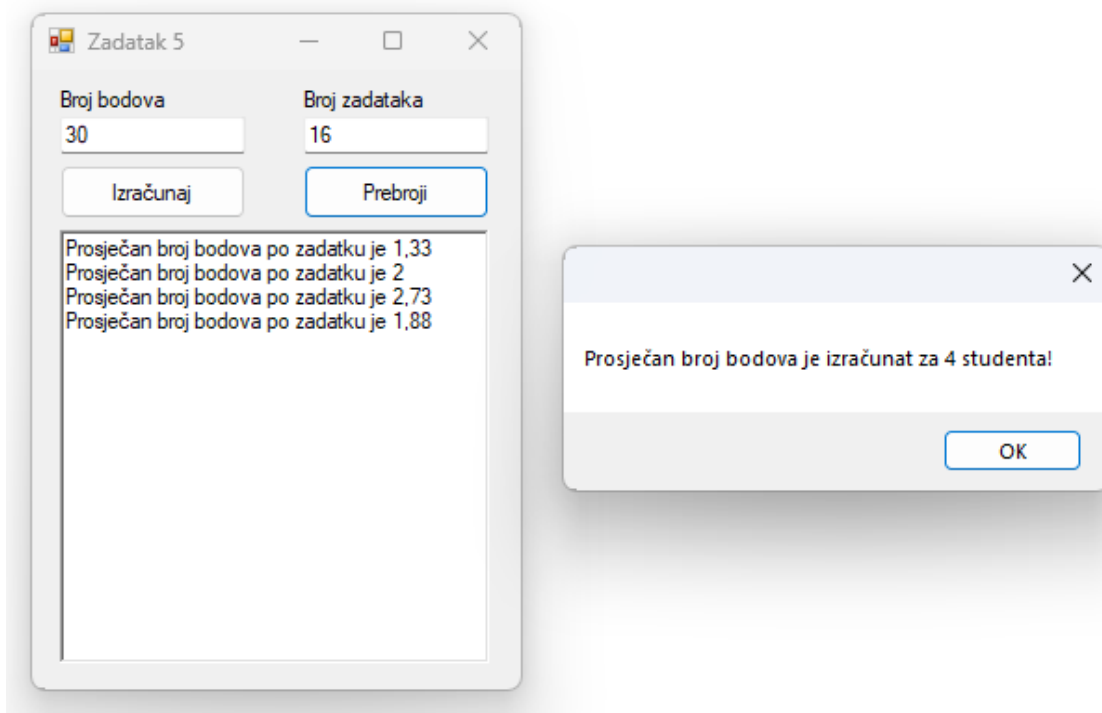


5. Napišite C# kod Windows forme koja izračunava prosječan broj bodova po zadatku na ispitu.

Korisnik unosi ukupan broj bodova studenta na ispitu u *TextBox* naziva *txtBodovi* te broj zadataka na ispitu u *TextBox* naziva *txtZadatci*. Na pritisak gumba naziva *btnIzracunaj* i teksta *Izračunaj* potrebno je izračunati prosječan broj bodova po zadatku te ga ispisati u *RichTextBox* naziva *rtxtIspis*. Svaki novi izračunati prosjek (novi pritisak gumba *Izračunaj*) prikazuje se u zasebnom retku *RichTextBoxa*.

Dodatno, pritiskom gumba naziva *btnPrebroji* i teksta *Prebroji* potrebno je u skočnom prozoru ispisati ukupan broj studenata za koje je prosječan broj bodova po zadatku izračunat. Format ispisa je proizvoljan.

Sučelje, s primjerom unesenih vrijednosti, prikazano je na sljedećoj slici.



```
public partial class Form1 : Form
{

    public Form1()
    {
        InitializeComponent();
    }

    private void btnIzracunaj_Click(object sender, EventArgs e)
    {

    }

    private void btnPrebroji_Click(object sender, EventArgs e)
    {

    }

}
```



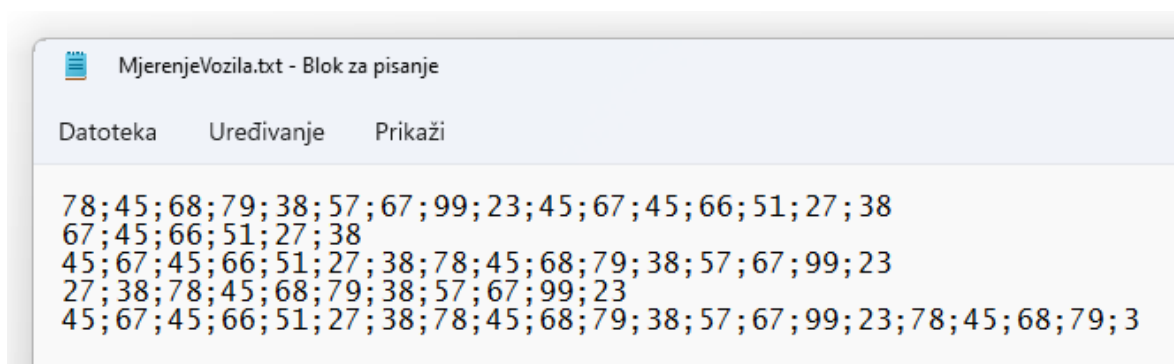
Ispit #78 (02.02.23. - grupa B)

1. Napišite C# kod metode **obradaMatrice** koja kao argument prima dvodimenzionalnu matricu brojeva s plivajućim zarezom **M**. Metoda treba vratiti jednodimenzionalno polje brojeva koji predstavljaju operaciju zbrajanja ili oduzimanja po redcima matrice **M**.

Ako je matrica **M** kvadratna izvršava se operacija zbrajanja vrijednosti u recima, inače ako matrica **M** nije kvadratna izvršava se operacija oduzimanja vrijednosti u redcima.

Prilikom operacija zbrajanja, odnosno oduzimanja, u obzir se uzimaju samo vrijednosti matrice koje se nalaze **ili** u intervalu $[-\infty, -10]$ ili u intervalu $[100, 200]$.

2. Napišite C# kod programa za učitavanje datoteke **MjerenjeVozila.txt** prikazane na sljedećoj slici smještene u direktoriju **D:**.

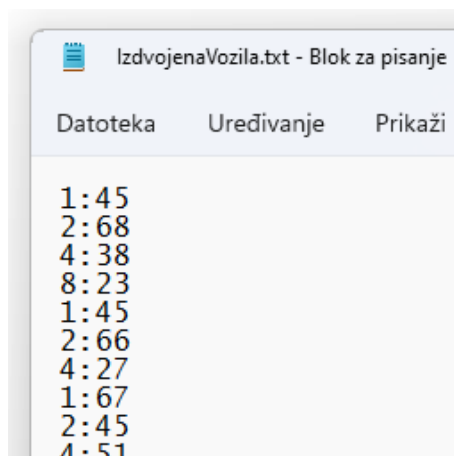


```
MjerenjeVozila.txt - Blok za pisanje
Datoteka   Uređivanje   Prikaži
78;45;68;79;38;57;67;99;23;45;67;45;66;51;27;38
67;45;66;51;27;38
45;67;45;66;51;27;38;78;45;68;79;38;57;67;99;23
27;38;78;45;68;79;38;57;67;99;23
45;67;45;66;51;27;38;78;45;68;79;38;57;67;99;23;78;45;68;79;3
```

U datoteci **MjerenjeVozila.txt** u pojedinom retku su izmjerene brzine vozila (cijeli brojevi) međusobno odvojeni točka-zarezom (;) na određenom dijelu prometnice u određeno vrijeme. Broj zabilježenih brzina po retku je varijabilan.

Program treba u novu datoteku **IzdvojenaVozila.txt**, smještenu u istom direktoriju **D:**, zapisati pojedine brzine iz svakog retka u formatu **X:Y** (u zasebnom retku), gdje znak **X** predstavlja indeks brzine u retku, a znak **Y** vrijednost te brzine. Iz svakog retka datoteke **MjerenjeVozila.txt** uzimaju se brzine na sljedećim indeksima 1, 2, 4, 8, 16, 32, ..., ∞ , odnosno dok god ima elemenata u retku na pojedinom indeksu (pomoć: eksponencijalna funkcija).

Potrebno je koristiti **try-catch** blok. Primjer zapisa u datoteci prikazan je na sljedećoj slici.



```
IzdvojenaVozila.txt - Blok za pisanje
Datoteka   Uređivanje   Prikaži
1:45
2:68
4:38
8:23
1:45
2:66
4:27
1:67
2:45
4:51
```



3. Napišite C# kod klase **Nesreca** koja mora imati tri javna atributa: *duzina* (broj s plivajućim zarezom), *sirina* (broj s plivajućim zarezom) i *policajska uprava* (niz znakova).

Klasa dodatno treba imati javnu metodi **ispis** koja ispisuje sadržaj klase u formatu *Duzina: X, sirina: Y i policajska uprava: Z*, pri čemu znakove *X*, *Y* i *Z* treba zamijeniti odgovarajućim vrijednostima.

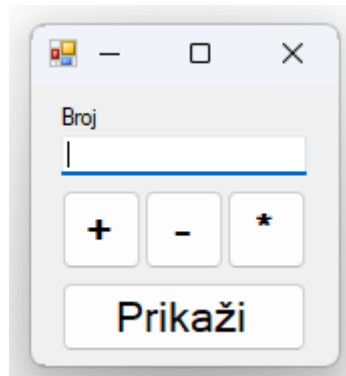
U glavnom dijelu programa učitajte **20** proizvoljnih instanci klase **Nesreca** koristeći petlju po vlastitom izboru, dodajte ih u listu proizvoljnog naziva i ispišite sadržaj pojedine instance klase u konzolu koristeći napisanu metodu **ispis**.

4. Što će se ispisati u konzolu nakon izvršavanja sljedećeg programskog odsječka? U zadatku je potrebno naznačiti redoslijed i vrijednosti koje se ispisuju.

```
static void Main(string[] args)
{
    int a = 2 + 7 / 2;
    int b = a + a % 5;
    Console.WriteLine(a + "," + b);
    M2(a, b);
}
static void M1(int a, int b)
{
    b = a % b;
    a--;
    Console.WriteLine(a + "," + b);
    a = 3;
}
static void M2(int a, int b)
{
    a = a * b;
    Console.WriteLine(a + "," + b);
    M1(a, b);
    Console.WriteLine(a + "," + b);
}
```



5. Nadopunite C# kod Windows forme koja radi operacije zbrajanja, oduzimanja i množenja cijelih brojeva. Forma je prikazana na sljedećoj slici.



Početna vrijednost rezultata je **0** (nula). Vrijednost cijelog broja unosi se u *TextBox* naziva *txtBroj*.

Pritiskom na gumb teksta + naziva *btnZbroj* povećava se vrijednost rezultata za uneseni broj.

Pritiskom na gumb teksta – naziva *btnRazlika* smanjuje se vrijednost rezultata za uneseni broj.

Pritiskom na gumb teksta * naziva *btnUmnozak* rezultat se množi s vrijednošću unesenog broja.

Pritiskom na gumb teksta **Prikaži** naziva *btnPrikazi* u skočnom prozoru ispisuje se poruka u formatu *Trenutni rezultat iznosi: X!*, pri čemu znak *X* predstavlja trenutnu vrijednost rezultata. Primjerice, ukoliko se obavi sljedeći redoslijed operacija: unos 10, zbroj, unos 5, razlika, unos 3 i umnožak - u skočnom prozoru treba se ispisati poruka *Trenutni rezultat iznosi: 15!*.

```
public partial class Form1 : Form
{
    public Form1()
    {
        InitializeComponent();
    }

    private void btnZbroj_Click(object sender, EventArgs e)
    {

    }

    private void btnRazlika_Click(object sender, EventArgs e)
    {

    }

    private void btnUmnozak_Click(object sender, EventArgs e)
    {

    }

    private void btnPrikazi_Click(object sender, EventArgs e)
    {

    }
}
```



Ispit #79 (16.02.23. - grupa A)

1. Napišite C# kod konzolne aplikacije koja zbraja realne brojeve unutar intervala $[n-m]$. Granice intervala potrebno je učitati s tipkovnice. Učitani brojevi se zbrajaju sve dok se ne unese broj izvan zadanog intervala.

Provjera da li je učitani broj unutar zadanog intervala vrši se unutar metode *provjeraUnosa* koja prima tri argumenta: učitani broj, donju granicu intervala i gornju granicu intervala. Metoda vraća *bool* vrijednost u ovisnosti da li je učitani broj unutar zadanog intervala ili izvan.

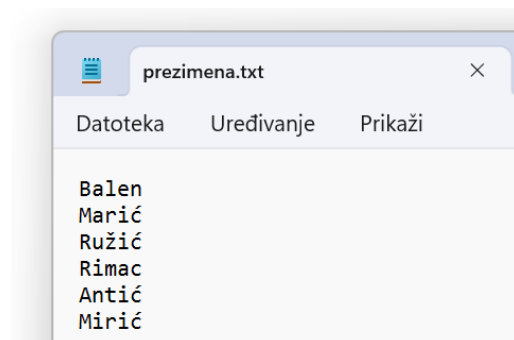
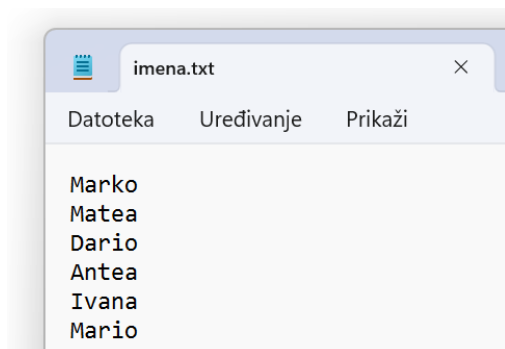
Nakon što je učitani broj izvan zadanog intervala, potrebno je ispisati sumu dosad učitanih brojeva u proizvoljnom formatu unutar glavnog dijela programa.

2. Napišite C# kod konzolne aplikacije koja učitava vrijednosti iz datoteka *imena.txt* i *prezimana.txt*. Imena se učitavaju u matricu *ime*, a prezimena u matricu *prezime*. Obje matrice su tipa podataka znaka (*char*). Sva imena i sva prezimena su jednake dužine te je u datotekama točno 6 imena i prezimena.

Nakon učitavanja matrica, inicijali studenata se pohranjuju u listu *inicijali* te ispisuju na konzolu u sljedećem formatu: „Inicijali studenata su: M.B.,M.M.,D.R.,A.R.,I.A.,M.M.“

Potrebno je koristiti *try-catch* blok nad dijelom koda za rad s datotekom, te nije potrebno pisati *Using System.IO*.

Primjer sadržaja datoteka prikazan je na slikama:



3. Što će se ispisati u konzolu nakon izvršavanja sljedećeg programskog odsječka? U zadatku je potrebno naznačiti redoslijed i vrijednosti koje se ispisuju.

```
int a = 7 % 2;
int b = 3 + a;
int c = 1;
int d = 0;

do
{
    if (a >= 2 && (b != 0 ^ d == 1))
    {
        c -= a;
        d = 2 * 4;
        b -= b;
    }
    else
    {
        c += a;
        d = a * 4 + 5;
        a--;
    }
    Console.WriteLine(a + "," + b + "," + c + "," + d);
} while (c + d == 2);
```



4. U zadatku je potrebno naznačiti redoslijed i vrijednosti koje se ispisuju. Kao točno rješenje priznaje se samo ono u kojem je vidljiv postupak i opis zašto je ispisana vrijednost poprimila baš tu vrijednost.

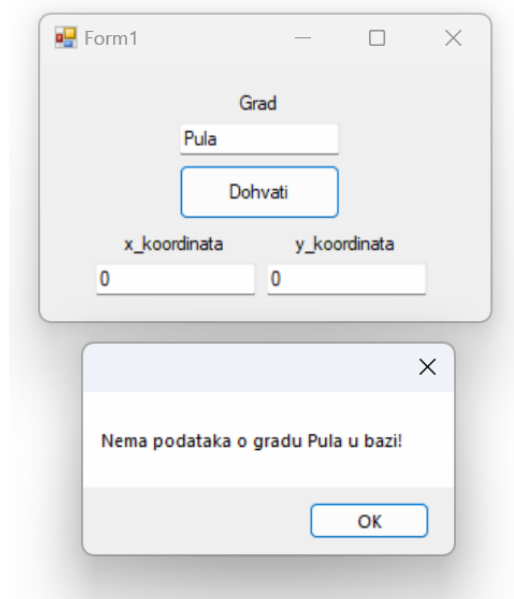
```
static void Main(string[] args)
{
    int n = 2;
    string vozac = "M.M.";
    string vozilo = "F.K.";
    VozniPark v = new VozniPark(vozilo, vozac, n, 0);
}
class VozniPark
{
    public int REDNIBROJ;
    public string VOZILO;
    public string VOZAC;
    public string POZICIJA;
    public VozniPark(string vozilo, string vozac, int RBprethodni,
double pozicija)
    {
        VOZILO = vozilo;
        postaviRB(RBprethodni);
        zamijeniVozaca(vozac);
        postaviPoziciju(pozicija + 2);
    }
    public void postaviRB(int x)
    {
        REDNIBROJ = x + 1;
        Console.WriteLine(REDNIBROJ);
    }
    public void zamijeniVozaca(string v)
    {
        VOZAC = v;
        Console.WriteLine(VOZAC);
    }
    public void postaviPoziciju(double p)
    {
        POZICIJA = POZICIJA + p;
        Console.WriteLine(POZICIJA);
        povecajPoziciju();
    }
    public void povecajPoziciju()
    {
        Console.WriteLine(POZICIJA);
        POZICIJA = POZICIJA + 1;
    }
}
```



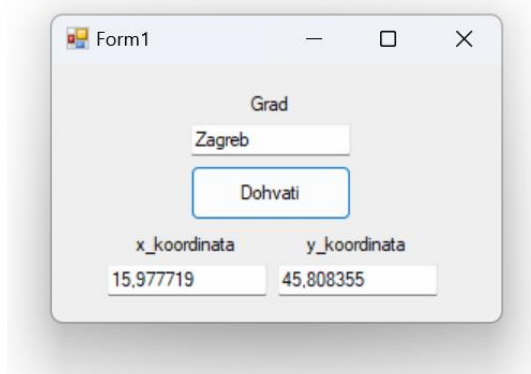
5. Napišite C# kod windows forme koja dohvaća x i y koordinatu grada na temelju unesenog naziva grada. Naziv grada se unosi *TextBoxom* naziva *txtGrad*.

Na pritisak gumba naziva *btnDohvati* i teksta *Dohvati* provjerava se postoji li uneseni naziv grada u listi *gradovi* te ako postoji u *TextBoxove* naziva *txtXkoord* i *txtYkoord* se ispisuju x i y koordinate grada. Koordinate se nalaze u listama *x_koord* i *y_koord* na istom indeksu na kojem se nalazi naziv odgovarajućeg grada u listi *gradovi*.

Ukoliko naziv grada nije pronađen, prikazuje se skočni prozor s porukom kako je prikazano na sljedećoj slici, a obje koordinate poprimaju vrijednost 0 te se ispisuju u pripadajućim *TextBoxovima*.



Sučelje, s primjerom unesenih vrijednosti grada koji postoji u listi *gradovi*, prikazano je na sljedećoj slici.



```

public partial class Form1 : Form
{
    public Form1()
    {
        InitializeComponent();

        List<string> gradovi = new List<string>() { "Zagreb", "Osijek", "Đakovo",
        "Split", "Fažana", "Gospić" };
        List<double> x_koord = new List<double>() { 15.977719, 18.695038, 18.411567,
        16.440157, 13.806238, 15.380820 };
        List<double> y_koord = new List<double>() { 45.808355, 45.552578, 45.311880,
        53.512867, 44.926796, 44.548078 };

        private void btnDohvati_Click(object sender, EventArgs e)
        {

        }
    }
}

```



Ispit #80 (25.03.23. - grupa A)

1. Napišite C# kôd klase *PrometnaRuta* koja mora imati tri privatna atributa: ime i tip rute (oba su niz znakova) te listu *stajalista* u kojoj su navedena stajališta u ruti (lista niza znakova).

Klasa treba imati konstruktor koji prima attribute ime i tip rute, postavlja im vrijednost i inicijalizira novu listu *stajalista*, te dvije javne metode: *ispis* i *dodajStajanje*.

Metoda *ispis* ne vraća rezultat i ne prima argumente, a treba u konzolu ispisati podatke instance klase *PrometnaRuta* u formatu: prometna ruta imena=*IME* i tipa=*TIP* sadrži *BROJ* stajališta! (umjesto znakova *IME*, *TIP* i *BROJ* trebaju pisati odgovarajuće vrijednosti atributa).

Metoda *dodajStajanje* preko argumenata prima znakovni niz, te dodaje u listu *stajališta* trenutne instance primljeni znakovni niz.

2. Napišite C# kôd konzolne aplikacije koja učitava matricu slučajnih brojeva s plivajućim zarezom te jednodimenzionalno polje cijelih brojeva. Potrebno je kreirati matricu *udaljenosti* veličine 4 x 5 te polje *troskovi* veličine 2.

Matrica *udaljenosti* predstavlja udaljenosti između gradova, a polje *troskovi* trošak prijevoza po prijeđenom kilometru.

Matricu *udaljenosti* potrebno je popuniti slučajnim brojevima s plivajućim zarezom u intervalu [1, 100>, a polje *troskovi* cijelim brojevima 2 i 4.

Zatim je potrebno izračunati ukupan trošak prijevoza između svih gradova prema sljedećem izrazu pri čemu je trošak prijevoza između svih gradova 2 osim onog grada koji se nalazi u trećem retku matrice *udaljenosti* čiji je trošak prijevoza 4:

$$trosakPrijevoza = \sum_{i=1}^4 \sum_{j=1}^5 (udaljenost_{i,j} * trosakPoKilometru)$$



3. Napišite C# kôd windows forme koja izračunava protok vozila na nekoj dionici ceste. Korisnik unosi gustoću prometnog toka u *TextBox* naziva *txtGustoca* te brzinu prometnog toka u *TextBox* naziva *txtBrzina*. Protok vozila se računa prema sljedećem izrazu:

$$\text{protok} = \text{gustoća} * \text{brzina}$$

Na pritisak gumba naziva *btnIzracunaj* i teksta *Izračunaj* potrebno je ispisati u *RichTextBoxu* naziva *txtIspis* protok vozila u formatu *Protok vozila je x!*“ pri čemu je x izračunati protok vozila. Svaki novi protok vozila (novi pritisak gumba *Izračunaj*) prikazuje se u zasebnom retku *RichTextBoxa* naziva *txtIspis*.

Dodatno, pritiskom gumba naziva *btnProsjek* i teksta *Prosjek* potrebno je u skočnom prozoru ispisati prosječan protok vozila na svim upisanim dionicama ceste. Format ispisa je proizvoljan.

Sučelje programa prikazano je na sljedećoj slici:

```
public partial class Form1 : Form
{
    public Form1()
    {
        InitializeComponent();
    }

    private void btnIzracunaj_Click(object sender, EventArgs e)
    {
    }

    private void btnProsjek_Click(object sender, EventArgs e)
    {
    }
}
```

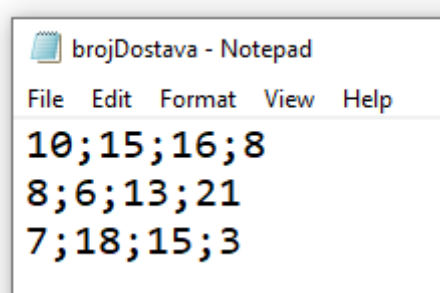
The screenshot shows a Windows application window titled "Form1". It has two text boxes at the top: "Gustoća prometnog toka" containing the number "10" and "Brzina" containing the number "20". Below these are two buttons: "Izračunaj" (highlighted with a blue border) and "Prosjek". Underneath the buttons is a RichText control displaying three lines of text: "Protok vozila je 1000", "Protok vozila je 500", and "Protok vozila je 200". Below the main window, there is a small modal dialog box with the text "Prosječan protok za unesene ceste je 566,67" and an "OK" button.



4. Što će se ispisati u konzolu nakon izvršavanja sljedećeg programskog odsječka?

```
static void Main(string[] args)
{
    int[,] matrica = new int[4, 3] { { 5, 7, 9 }, { 6, 8, 4 }, { 5, 6, 7 },
    { 1, 2, 3 } };
    for (int j = matrica.GetLength(1) + 1; j > 2; j--)
    {
        for (int i = -1; i <= matrica.GetLength(0) - 1; i++)
        {
            if (i + 2 == j - 1)
            {
                Console.WriteLine(matrica[i + 1, j - 2]);
            }
        }
    }
    Console.WriteLine(matrica.Length);
}
```

5. Što će se ispisati u konzolu nakon izvršavanja sljedećeg programskog odsječka. Sadržaj datoteke koja se učitava prikazan je sljedećom slikom:



Potrebno je naznačiti redoslijed ispisivanja. Kao točno rješenje priznaje se samo ono u kojem je vidljiv postupak i opis zašto je ispisana vrijednost poprimila baš tu vrijednost.

```
static void Main(string[] args)
{
    StreamReader citaj = new StreamReader("brojDostava.txt");
    while (!citaj.EndOfStream)
    {
        string linija = citaj.ReadLine();
        string[] dijeloviLinije = linija.Split(';');
        int odabir = 0;
        for (int i = 0; i < dijeloviLinije.Length; i++)
        {
            odabir = Convert.ToInt32(dijeloviLinije[i]);
        }
        if (odabir % 3 == 0)
            Console.WriteLine("3");
        else if (odabir % 5 == 0)
            Console.WriteLine("5");
        else
            Console.WriteLine(odabir);
    }
    Console.WriteLine(citaj.EndOfStream);
    citaj.Close();
}
```



Ispit #81 (15.06.23. - grupa A)

1. Što će se ispisati u konzolu nakon izvršavanja sljedećeg programskog odsječka? U zadatku je potrebno naznačiti redoslijed i vrijednosti koje se ispisuju.

```
List<string> list = new List<string>();
list.AddRange(new string[] { "Ana", "Marko", "Miro" });
list.InsertRange(list.Count - 2, new string[] { "Tomo", "Orogoro" });
list.Remove("Marko");
int duljina = list[0].Length;
list.RemoveAt(list.Count - duljina);
for (int i = 0; i < list.Count; i++)
{
    int brojac = 0;
    string element = list[i];
    for (int j = 0; j < element.Length; j++)
    {
        if (element[j] == 'o')
        {
            brojac++;
        }
    }
    Console.WriteLine(brojac);
}
```

2. Napišite C# kôd metode **provjeraRijeci** koja vraća **istinu ili laž** ovisno o tome da li je riječ predana kao argument ispravna ili ne.

Riječ je ispravna ako je broj znakova u riječi veći od 4 i ako riječ ne sadrži niti jedan od samoglasnika **a, e, i, o i u**.

U glavnom dijelu programa potrebno je kontinuirano učitavati proizvoljne riječi s tipkovnice sve dok se ne učita neispravna riječ. Za provjeru ispravnosti koristiti metodu **provjeraRijeci**.

Nakon učitavanja pojedine riječi potrebno je ispisati u konzolu da li je ispravna ili ne u formatu kao u primjeru. Primjerice, ako se učita riječ: **drvnk** program treba ispisati **Učitana riječ drvnk je ispravna!**.

3. Napišite C# kôd klase **Prometnica** koja mora imati pet privatnih atributa: niz znakova kojim se bilježi kategorija prometnice te četiri broja s plivajućim zarezom kojima se bilježe geografske koordinate početne i krajnje točke prometnice (**xPocetak, yPocetak, xKraj, yKraj**).

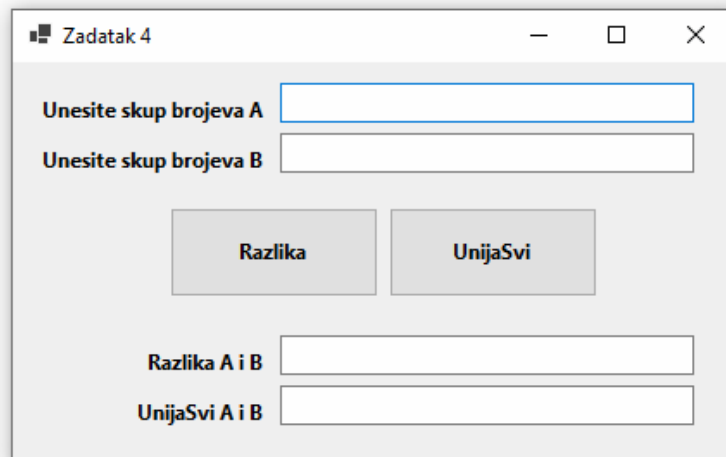
Klasa treba imati konstruktor koji prima sve attribute klase i postavlja im vrijednosti, te javnu metodu **izracunajDuzinu**. Metoda **izracunajDuzinu** **ne prima argument**, ali izračunava i vraća Euklidsku udaljenost između početne i krajnje točke trenutne instance prometnice, tj. izračunava duljinu prometnice prema sljedećem izrazu:

$$udaljenost(T_1, T_2) = \sqrt{(x_{T_2} - x_{T_1})^2 + (y_{T_2} - y_{T_1})^2}$$

Unutar glavnog dijela programa potrebno je preko tipkovnice učitati sve potrebne vrijednosti (proizvoljne) za stvaranje jedne instance klase, te potom u konzolu ispisati duljinu prometnice te instance koristeći metodu **izracunajDuzinu**. Format ispisa mora biti: **Duljina prometnice kategorije X iznosi Y!** (znakove X i Y potrebno je zamijeniti odgovarajućim vrijednostima).



4. Nadopunite C# kôd prikazan ispod za windows formu prikazanu na sljedećoj slici:



Forma sadrži 4 *Label*-a s tekstovima prikazanim na slici; 4 *TextBox*-a naziva *TxtSkupaA*, *TxtSkupB*, *TxtRazlika* i *TxtUnijaSvi*; i dva gumba (*Button*-a) naziva *BtnRazlika* i *BtnUnijaSvi* s tekstovima prikazanim na slici.

U *TextBox*-ove *TxtSkupaA* i *TxtSkupB* potrebno je unijeti brojeve skupova A i B (očekuje se da će se za jedan element skupa unositi samo cijeli brojevi od 0 do 9).

Pritiskom na gumb *BtnRazlika* potrebno je izračunati razliku skupa A i B, te ispisati razliku unutar *TextBox*-a *TxtRazlika* (razlika skupa A i B, su svi oni koji su u skupu A, a nisu u skupu B).

Pritiskom na gumb *BtnUnijaSvi* potrebno je ukupnu uniju skupa A i B ispisati unutar *TextBox*-a *TxtUnijaSvi* (unija skupova A i B je skup svih elemenata skupa A i B – čak i onih koji se više puta ponavljaju).

Provjera ispravnosti: A = 789; B = 293; Razlika A i B = 78; Unija Svi A i B = 789293.

```
public partial class Form1 : Form
{
    public Form1()
    {
        InitializeComponent();
    }

    private void BtnRazlika_Click(object sender, EventArgs e)
    {
    }

    private void BtnUnijaSvi_Click(object sender, EventArgs e)
    {
    }
}
```



5. Što će se ispisati u konzoli nakon izvršavanja sljedećeg programskog odsječka?

```
static void Main(string[] args)
{
    Console.WriteLine(TargetMethod(2));
}
public static int TargetMethod(int n)
{
    if (n % 2 == 0)
    {
        Console.WriteLine(n++);
    }
    if (n > 8)
    {
        return n;
    }
    else
    {
        return TargetMethod(n + 1);
    }
}
```



Napomena: nedostaje peti zadatak!

1. Što će se ispisati u konzolu nakon izvršavanja sljedećeg programskog odsječka?

```
static void Main(string[] args)
{
    ListaBrojeva();
}
public static void ListaBrojeva()
{
    List<int> lista = new List<int>();
    int i;
    for (i = 32; i > 0; i /= 2)
        lista.Add(i--);
    lista.RemoveAt(3);
    lista.InsertRange(i + 2, new int[] { lista[i + 1] / 2,
        lista[lista.Count - 1 - i] / 2 });
    Console.Write(lista[0]);
    for (i = 1; i < lista.Count; i++)
    {
        Console.Write("; " + lista[i]);
    }
}
```

2. Napišite C# kôd aplikacije koja generira datoteku **data.txt** unutar projekta. Datoteka sadrži 100 slučajno generiranih brzina vozila te ID-eve vozila. Format retka datoteke treba biti **ID;Brzina**.

Brzine su decimalni brojevi u rasponu **[0, 180>**. Generirana brzina treba biti zaokružena na dva decimalna mjesta.

Vrijednost ID-a vozila kreću od **0** te svako novo vozilo ima vrijednost ID-a veću za **1** u usporedbi s prethodnim vozilom.

Nakon popunjavanja, potrebno je pozvati metodu za spremanje sadržaja datoteke, te metodu za zatvaranje datoteke.

Potrebno je u konzolu ispisati najveću generiranu brzinu te pripadni ID vozila. Format ispisa mora biti: *Najveća brzina: X, ID: Y!* (znakove **X** i **Y** potrebno je zamijeniti s odgovarajućim vrijednostima). Potrebno je koristiti **try-catch** blok.



3. Napišite C# kôd klase koja mora imati tri privatna atributa: niz znakova kojima se bilježi status dostave, cijeli broj kojim se bilježi vrijednost kupca te cijeli broj kojim se bilježi vrijednost restorana.

Klasa mora imati konstruktor koji prima sve attribute klase i postavlja im vrijednosti, te javnu metodu **estVrijemeDostave**.

Metoda *estVrijemeDostave* ne prima argumente, ali izračunava očekivano vrijeme dostave hrane. Ako je status dostave **pripremaRobe**, očekivano vrijeme dostave se računa pomoću prvog izraza, u suprotnom pomoću izraza 2, pri čemu **x** predstavlja vrijednost restorana, a **y** vrijednost kupca.

$$(1) t = \frac{x^2}{y^{1.5}}$$

$$(2) t = \frac{x}{y}$$

Unutar glavnog dijela programa (*Main*) potrebno je inicijalizirati tri liste: *statusi* – sadrži nizove znakova { **pripremaRobe**, **dostavlja** }, *kupci* sadrži cijele brojeve { **2**, **4**, **6** }, i *restorani* – sadrži cijele brojeve { **5**, **10**, **15** }.

Potrebno je stvoriti 100 dostava tako da se svakom atributu nove klase dodijeli slučajna (*Random*) vrijednost iz pripadnih lista. Nakon svakog stvaranja objekta klase *Dostava*, potrebno je ispisati u konzolu očekivano vrijeme dostave koristeći metodu *estVrijemeDostave*. Format ispisa mora biti: *Očekivano vrijeme dostave: X* (znak **X** potrebno je zamijeniti s odgovarajućom vrijednosti).

```
public class Dostava
{
    //Nadopunite kod

}
```

4. Napišite C# kôd programa koji preko tipkovnice popunjava matricu cijelih brojeva **M = 3 x 4** s proizvoljnim vrijednostima (cijelim brojevima).

Nakon uspješnog popunjavanja svih elemenata, potrebno je u konzolu ispisati ukupnu i prosječnu učitano vrijednost i ukupan broj unesenih vrijednosti. Format ispisa mora biti: *Sum: X, Avg: Y, Count: Z* (znakove **X**, **Y**, i **Z** potrebno je zamijeniti odgovarajućim vrijednostima).



Napomena: nedostaju četvrti i peti zadatak!

1. Što će se ispisati u konzolu nakon izvršavanja sljedećeg programskog odsječka?

```
static void Main(string[] args)
{
    ListaSlova();
}
public static void ListaSlova()
{
    List<string> slova = new List<string>() { "A", "O", "C" };
    slova.AddRange(new List<string> { "B", "O", "O", "R" });
    List<string> listaSlova = new List<string>();
    int i = 0;
    try
    {
        for (i = slova.Count; i > 0; i -= i % 2)
        {
            string tempSlovo = slova[--i];
            listaSlova.Add(tempSlovo);
        }

        Console.WriteLine(listaSlova[--i]);
    }
    catch (Exception ex)
    {
        listaSlova.Insert(listaSlova.Count, "T");
    }
    listaSlova.RemoveAt(4);
    Console.Write(listaSlova[0]);
    for (i = 1; i < listaSlova.Count; i += 2)
    {
        Console.Write("; " + listaSlova[i]);
    }
}
```

2. Napišite C# kôd aplikacije koja učitava podatke iz lokalne datoteke **data.txt**. Datoteka sadrži *n* zabilježenih brzina vozila te ID-eve vozila. Format retka datoteke: **ID;Brzina**. Brzine su decimalni brojevi. Prvi zapis u datoteci ima ID **0** te svako sljedeće vozilo ima ID vrijednosti veću za **1** u usporedbi s prethodnim zapisom.

Potrebno je učitati sve brzine u listu ili u polje (proizvoljno).

Nakon učitavanja podataka potrebno je zatvoriti otvorenu datoteku te je potrebno u konzolu ispisati najmanju zabilježenu brzinu te pripadni ID vozila. Format ispisa mora biti: *Najmanja brzina: X, ID: Y!* (znakove **X** i **Y** potrebno je zamijeniti s odgovarajućim vrijednostima).

Potrebno je koristiti **try-catch** blok.



3. Napišite C# kôd metode *provjeraSimetricnostiMatrice* koja kao argument prima matricu prirodnih brojeva te vraća *bool* vrijednost u ovisnosti o tome da li je matrica simetrična ili ne.

Nakon provjere potrebno je u konzolu ispisati da li je provjerena matrica simetrična ili ne. Format ispisa je proizvoljan.

```
static void Main(string[] args)
{
    //int[,] matrica = new int[n, m];
    bool simetricna = provjeraSimetricnostiMatrice(matrica);
    //Nadopunite kod

}
public static bool provjeraSimetricnostiMatrice(int[,] m)
{
    //Nadopunite kod

}
```



Ispit #84 (31.08.23. - grupa A)

Napomena: uspio sam nabaviti samo drugi zadatak

2. Napišite C# kôd klase *GPS_trajektorija* koja mora imati tri privatna atributa: vrijeme (klasa *DateTime*), oznaku korisnika (niz znakova) i listu GPS točaka (lista decimalnih brojeva).

Klasa treba imati konstruktor koji prima sve atribute klase i postavlja im vrijednosti, te dvije javne metode: *ispis* i *dodaj*.

Metoda *ispis* ne vraća rezultat i ne prima argumente, a treba u konzolu ispisati podatke instance klase *GPS_trajektorija* u formatu: GPS trajektorija zabilježena u vrijeme=VRIJEME sadrži X GPS točaka! (umjesto znakova VRIJEME i X treba napisati odgovarajuće vrijednosti).

Metoda *dodaj* preko argumenata prima dva decimalna broja te ih dodaje u listu GPS točaka.



Ispit #85 (07.09.23. - grupa B)

1. Napišite metodu **MaxAbsDiff** koja preko argumenata prima matricu **M = 3 x 3** cijelih brojeva, te pronalazi i ispisuje najveću apsolutnu razliku između dva elementa matrice – od svih mogućih kombinacija elemenata i njihovih apsolutnih razlika ispisuje onu najveću u sljedećem formatu *Najveća apsolutna razlika između pojedinih elemenata matrice iznosi Z!, a čine ju element x=X i element y=Y* (znakove Z, X i Y potrebno je zamijeniti s odgovarajućim vrijednostima).

Apsolutna razlika između dva broja računa se sljedećim izrazom:

$$au = |x - y|$$

2. Napišite C# metodu *WallFollowing* koja usmjerava kretanje robota kroz labirint. Cilj je provesti robota od početne do krajnje točke. Za rješavanje problema koristit će se „WallFollowing“ algoritam te unaprijed određene metode *wallLeft()*, *wallRight()*, *wallFront()*, *finish()*, *turnLeft()*, *turnRight()*, i *moveForward()*.

Kôd navedenih metoda ne trebate dopunjavati.

Metode pozivate prema potrebi, sukladno njihovim opisima. Algoritam uvijek nastoji držati desnu stranu robota uz zid dok se kreće kroz labirint.

Logika algoritma:

- ako nema zida s desna, skreni desno
- dok je zid ispred tebe, skreći lijevo
- kreći se unaprijed

Opis metoda:

- Provjere:
 - *wallLeft()* – Istina ako se zid nalazi lijevo od robota, u suprotnom Laž
 - *wallRight()* – Istina ako se zid nalazi desno od robota, u suprotnom Laž
 - *wallFront()* – Istina ako se zid nalazi ispred robota, u suprotnom Laž
 - *finish()* – Istina ako se robot nalazi na krajnjoj ćeliji, u suprotnom Laž
- Akcije:
 - *turnLeft()* – okreni robota devedeset stupnjeva u lijevo
 - *turnRight()* – okreni robota devedeset stupnjeva u desno
 - *moveForward()* – pomakni robota jednu ćeliju unaprijed



3. Napišite i pozovite C# metodu *BitSwap* koja kao argument prima listu cijelih brojeva. Lista sadrži samo raspon vrijednosti **[0, 1]** (nije potrebno provjeravati da li elementi liste zadovoljavaju uvjet).

Metoda ulaznom argumentu mijenja vrijednost svakog broja **jedan (1)** u broj **nula (0)**, te svakog broja **nula (0)** u broj **jedan (1)**. Pri čemu redoslijed vrijednosti mora ostati isti.

Unutar metode potrebno je ispisati elemente ulaznog parametra u istoj liniji te elemente izlazne liste u zasebnoj liniji.

Primjer ispisa (dvije linije):

Ulazne vrijednosti: 10110

Izlazne vrijednosti: 01001

4. Napišite C# kôd aplikacije koja generira datoteku **speed.txt** unutar projekta.

Datoteka sadrži 74 slučajno generirane brzine vozila te ID-eve vozila. Format retka datoteke treba biti: **Brzina-ID**.

Brzine su decimalni brojevi u rasponu **[53, 142>**. Generirana brzina treba biti zaokružena na dva decimalna mjesta.

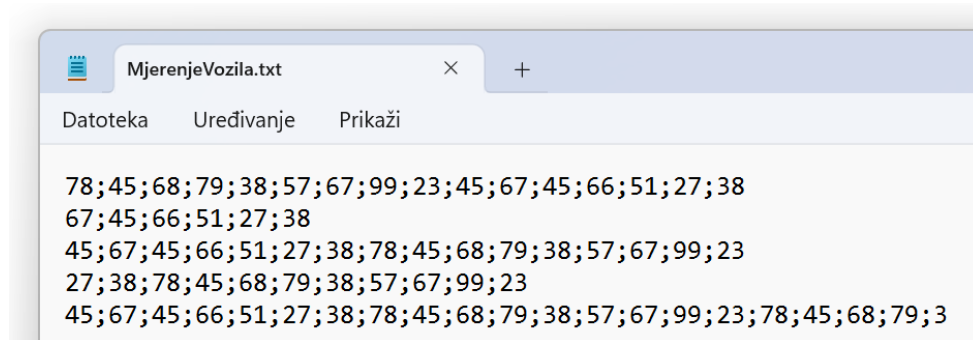
Vrijednost ID-a vozila kreću od **1** te svako novo vozilo ima vrijednost ID-a veću za **1** u usporedbi s prethodnim vozilom.

Nakon popunjavanja, potrebno je pozvati metodu za spremanje sadržaja datoteke, te metodu za zatvaranje datoteke.

Potrebno je u konzolu ispisati najveću generiranu brzinu te pripadni ID vozila. Format ispisa mora biti: *Najveća brzina: X, ID: Y!* (znakove X i Y potrebno je zamijeniti s odgovarajućim vrijednostima). Potrebno je koristiti **try-catch** blok.



5. Napišite C# kôd programa za učitavanje datoteke **MjerenjeVozila.txt** prikazane na sljedećoj slici smještene u direktoriju **D:**.



```
MjerenjeVozila.txt
Datoteka  Uređivanje  Prikaži

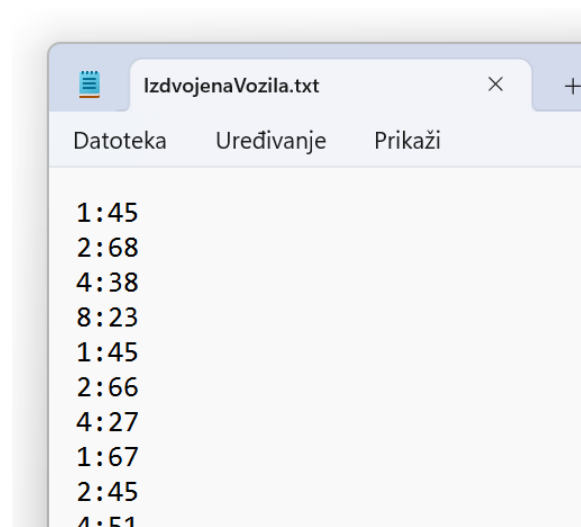
78;45;68;79;38;57;67;99;23;45;67;45;66;51;27;38
67;45;66;51;27;38
45;67;45;66;51;27;38;78;45;68;79;38;57;67;99;23
27;38;78;45;68;79;38;57;67;99;23
45;67;45;66;51;27;38;78;45;68;79;38;57;67;99;23;78;45;68;79;3
```

U datoteci **MjerenjeVozila.txt** u pojedinom retku zapisane su izmjerene brzine vozila (cijeli brojevi) međusobno odvojeni točka-zarezom (;) na određenom dijelu prometnice u određeno vrijeme. Broj zabilježenih brzina po retku je **varijabilan**.

Program treba u novu datoteku **IzdvojenaVozila.txt** smještenu u istom direktoriju **D:** zapisati pojedine brzine iz svakog retka u formatu **X:Y** (u zasebnom retku), gdje znak **X** predstavlja indeks brzine u retku, a znak **Y** vrijednost te brzine.

Iz svakog retka datoteke **MjerenjeVozila.txt** uzimaju se brzine na sljedećim indeksima 1, 2, 4, 8, 16, 32, ..., ∞ , odnosno dok god ima elemenata na pojedinom indeksu (pomoć: eksponencijalna funkcija).

Potrebno je koristiti **try-catch** blok. Primjer zapisa u datoteci prikazan je na sljedećoj slici:



```
IzdvojenaVozila.txt
Datoteka  Uređivanje  Prikaži

1:45
2:68
4:38
8:23
1:45
2:66
4:27
1:67
2:45
1:51
```



Ispit #88 (22.02.24. - grupa A)

Napomena: nedostaje peti zadatak

1. Što će se ispisati nakon izvršavanja sljedećeg programskog odsječka? Priznaje se samo rješenje u kojem je vidljiv sadržaj varijabli.

```
1 int z = 2 + 4 / 3;
2 double x = 3.0 / 2;
3 int y = 3 * z;
4 do
5 {
6     if (y / 3 != 3)
7     {
8         z = z * 2;
9         y = y % 2;
10    }
11    else
12    {
13        z = y + 2;
14        y = y - 3;
15    }
16    x--;
17    x--;
18    Console.WriteLine(z + "," + y);
19 }
20 while (x > -1);
```

2. Napišite C# funkciju *BitSwap* koja kao argument prima listu cijelih brojeva. Lista sadrži samo raspon vrijednosti [0, 1] (nije potrebno provjeravati da li elementi liste zadovoljavaju uvjet).

Funkcija ulaznom argumentu mijenja vrijednost svakog broja **jedan (1)** u broj **nula (0)**, te svakog broja **nula (0)** u broj **jedan (1)**. Redoslijed vrijednosti mora ostati isti.

Funkcija kao rezultat vraća listu novih bitova.

Primjer:

Ulazne vrijednosti: 10110

Izlazne vrijednosti: 01001

```
1 static void Main(string[] args)
2 {
3     List<int> x = new List<int>();
4     x.AddRange(new[] { 1, 0, 1, 1, 0 });
5     List<int> y = BitSwap(x);
6 }
```



3. Koliko puta će sljedeći programski isječak ispisati poruku „Pozdrav!“? Priznaje se samo rješenje u kojem je vidljiv sadržaj varijabli.

```
1 static void Main(string[] args)
2 {
3     rekMetoda(999);
4     Console.ReadKey();
5 }
6
7 static void rekMetoda(int x)
8 {
9     if (x >= 0 && x < 10)
10    {
11        Console.WriteLine("Pozdrav!");
12        return;
13    }
14    else
15    {
16        Console.WriteLine("Pozdrav!");
17        rekMetoda(x / 10);
18        rekMetoda(x / 100);
19        Console.WriteLine("Pozdrav!");
20    }
21 }
```

4. Napišite C# kôd klase **Nesreca** koja mora imati tri javna atributa: dužinu (broj s plivajućim zarezom), širinu (broj s plivajućim zarezom) i policijsku upravu (niz znakova).

Klasa dodatno treba imati javnu metodu koja ispisuje sadržaj klase u formatu „*Dužina:X, širina:Y i policijska uprava: Z*“, pri čemu znakove **X**, **Y** i **Z** treba zamijeniti s odgovarajućim vrijednostima.

U glavnom dijelu programa učitajte **20** proizvoljnih instanci klase koristeći **petlju** po vlastitom izboru, dodajte ih u listu proizvoljnog naziva i ispišite sadržaj pojedine instance klase u konzolu koristeći napisanu metodu **ispis**.



Ispit #89 (06.04.24. - grupa A)

1. Što će se ispisati nakon izvršavanja sljedećeg programskog odsječka? Priznaje se samo rješenje u kojem je vidljiv sadržaj varijabli.

```
1 int[,] matrix = new int[3, 2] { { 1, 2 }, { 3, 4 }, { 5, 6 } };
2 for (int i = matrix.GetLength(0) - 2; i >= 0; i--)
3 {
4     for (int j = 0; j < matrix.GetLength(1); j++)
5     {
6         if (matrix[i, j] % 3 == 0)
7         {
8             Console.WriteLine(matrix[i, j]);
9         }
10        else
11        {
12            Console.WriteLine(matrix[i + 1, j]);
13        }
14    }
15 }
```



2. Što će se ispisati nakon izvršavanja sljedećeg programskog odsječka? Priznaje se samo rješenje u kojem je vidljiv sadržaj varijabli.

```
1 public class Avion
2 {
3     private double brzina;
4     private double visina;
5     private double udaljenost;
6     private double vrijemeLeta;
7     public Avion(double brzina, double visina)
8     {
9         this.brzina = brzina;
10        this.visina = visina;
11        this.udaljenost = 0;
12        this.vrijemeLeta = 0;
13    }
14    public void PromijeniParametre(double brzina, double visina)
15    {
16        this.brzina = brzina;
17        this.visina = visina;
18    }
19    public void PromijeniParametre2()
20    {
21        this.brzina = 400;
22        this.visina = 8000;
23    }
24    public void IzracunajVrijemeLeta()
25    {
26        this.vrijemeLeta = this.visina / this.brzina;
27        Console.WriteLine(Math.Round(this.vrijemeLeta, 2));
28    }
29    public void IzracunajUdaljenost()
30    {
31        double udaljenost = this.brzina * this.vrijemeLeta;
32        Console.WriteLine(this.udaljenost);
33    }
34 }
35
36 internal class Program
37 {
38     static void Main(string[] args)
39     {
40         Avion avion = new Avion(250, 10000);
41         avion.IzracunajVrijemeLeta();
42         avion.IzracunajUdaljenost();
43         avion.PromijeniParametre2();
44         avion.IzracunajVrijemeLeta();
45         avion.PromijeniParametre(300, 9000);
46         avion.IzracunajVrijemeLeta();
47     }
48 }
```



3. Napišite konzolnu aplikaciju koja preko konzole učitava vrijeme putovanja t u satima (realan broj) i ID tipa vozila (cijeli broj).

ID tipa vozila dan je stupcem ID u tablici.

ID	Opis	Brzina [°/h]
1	Motor	$v_x = v_y = 0.7622$
2	Automobil	$v_x = v_y = 0.5717$
3	Kamion	$v_x = v_y = 0.5082$

Početna pozicija vozila u geografskim stupnjevima je $T(x_p, y_p) = (15.97790, 45.810857)$.

Ovisno o tome koji je ID vozila unesen potrebno je ispisati poziciju na kojoj će se vozilo naći nakon što se vozi t vremena, pri čemu su brzine zadane u tablici, a konačna pozicija $T(x_k, y_k)$ se računa koristeći izraze:

$$x_k = v_x * t + x_p$$

$$y_k = v_y * t + y_p$$

Format ispisa mora biti kao u primjeru. Primjerice, ukoliko se učitava vrijeme od 2 sata i vozilo pod ID-em 1, potrebno je ispisati: „Motor će se nakon 2 sata vožnje nalaziti na poziciji T (17.502305, 47.335257)!“.

4. Napišite C# metodu *provjeraRijeci* koja kao argument prima jedan znakovni niz i tri znaka: *varRijec*, *sPoc*, *sZavr* i *slzmedu*.

Ukoliko predana riječ (varijabla *varRijec*) počinje znakom *sPoc* ili završava znakom *sZavr* metoda vraća broj pojavljivanja znaka *slzmedu* između početnog i završnog znaka (granice nisu uključujuće). U suprotnom metoda vraća duljinu znakovnog niza *varRijec*.

5. Napišite C# kôd aplikacije koja učitava podatke iz lokalne datoteke **data.txt**. Datoteka sadrži n zabilježenih brzina vozila te ID-eve vozila. Format retka datoteke: **ID;Brzina**.

Brzine su decimalni brojevi. Prvi zapis u datoteci ima ID **0** te svako sljedeće vozilo ima ID vrijednost veću za **1** u usporedbi s prethodnim zapisom.

Potrebno je učitati sve brzine u listu. Nakon učitavanja podataka potrebno je zatvoriti otvorenu datoteku te je potrebno dohvatiti slučajan realan broj unutar intervala $[0, 50>$. Potom je potrebno izračunati i ispisati prosjek učitanih brzina koje su isključivo veće od dohvaćenog slučajnog broja.

Format ispisa mora biti: *Prosjek brzina u listi koje su veće od brzine X iznosi Y!* (znakove **X** i **Y** potrebno je zamijeniti s odgovarajućim vrijednostima).

Potrebno je koristiti **try-catch** blok.



Ispit #90 (27.06.24.)

1. Što će se ispisati u konzoli nakon izvršavanja sljedećeg programskog odsječka? Priznaje se samo rješenje u kojem je vidljiv postupak.

```
1 static void Main(string[] args)
2 {
3     Console.WriteLine(TargetMethod(2));
4 }
5 public static int TargetMethod(int n)
6 {
7     if (n % 2 == 0)
8     {
9         Console.WriteLine(n);
10    }
11    if (n > 4)
12    {
13        return n;
14    }
15    else
16    {
17        return TargetMethod(n + 1);
18    }
19 }
```



2. Što će se ispisati nakon izvršavanja sljedećeg programskog odsječka? Priznaje se samo rješenje u kojem je vidljiv postupak.

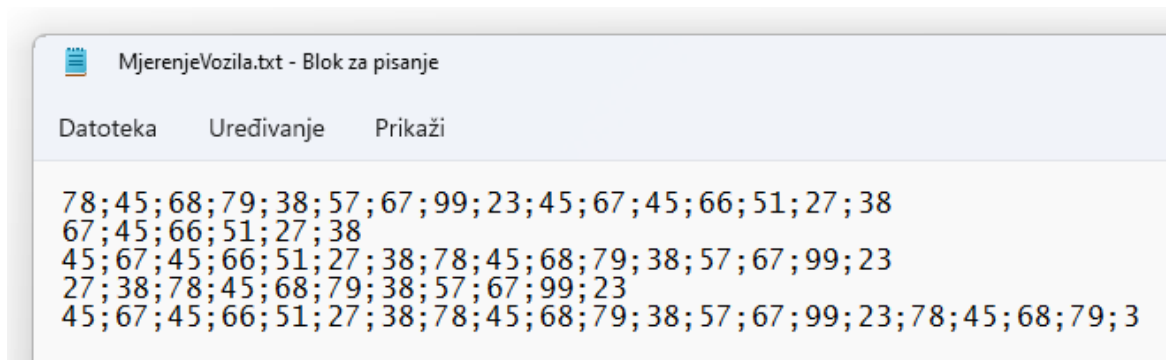
```
1 static void Main(string[] args)
2 {
3     PravokutanTrokut p = new PravokutanTrokut(8.5, 2, 10);
4     Console.WriteLine(p.izracunajPovrsinu(2.5, 4));
5     p.izracunajPovrsinu();
6     p.ispis();
7 }
8 class PravokutanTrokut
9 {
10     private double A;
11     private double B;
12     private double POVRSINA;
13     public PravokutanTrokut(double a, double b, double povrsina)
14     {
15         A = a;
16         B = b;
17         POVRSINA = povrsina;
18         ispis();
19     }
20     public void izracunajPovrsinu()
21     {
22         POVRSINA = (A * B) / 2;
23     }
24     public double izracunajPovrsinu(double a, double b)
25     {
26         double povrsina = (b * a) / 2;
27         ispis();
28         return povrsina;
29     }
30     public void ispis()
31     {
32         Console.WriteLine(POVRSINA);
33     }
34 }
```

3. Napišite C# kôd metode *izracun* koja kao argument prima dvodimenzionalnu kvadratnu matricu cijelih brojeva *M*.

Ako su svi elementi na dijagonali jednaki nula, metoda vraća broj elemenata u matrici većih od nula, inače vraća sumu svih elemenata matrice.



4. Napišite C# kôd programa za učitavanje datoteke **MjerenjeVozila.txt** prikazane na sljedećoj slici smještene na relativnoj putanji.



```
MjerenjeVozila.txt - Blok za pisanje
Datoteka   Uređivanje   Prikaži
78;45;68;79;38;57;67;99;23;45;67;45;66;51;27;38
67;45;66;51;27;38
45;67;45;66;51;27;38;78;45;68;79;38;57;67;99;23
27;38;78;45;68;79;38;57;67;99;23
45;67;45;66;51;27;38;78;45;68;79;38;57;67;99;23;78;45;68;79;3
```

U datoteci **MjerenjeVozila.txt** u pojedinom retku su izmjerene brzine vozila u km/h (cijeli brojevi) međusobno odvojeni točka-zarezom (;) na određenom dijelu prometnice u određeno vrijeme. Broj zabilježenih brzina po retku je **varijabilan**.

Program treba u novu datoteku **IzdvojenaVozila.txt**, smještenu na relativnoj putanji, zapisati brzine koje su veće od ograničenja (70 km/h) u formatu **X;Y;Z** (u zasebnom retku), gdje znak **X** predstavlja broj linije iz datoteke **MjerenjeVozila.txt** (prva linija ima broj 0, a svaka sljedeća za 1 veći broj), znak **Y** predstavlja indeks (poziciju) brzina unutar jednog retka (prva brzina ima indeks 0, a svaka sljedeća za 1 veći), a znak **Z** predstavlja vrijednost te brzine.

Primjerice, za brzinu u datoteci s iznosom 79, u datoteci **IzdvojenaVozila.txt** se treba pojaviti zapis 0;3;79. Potrebno je koristiti **try-catch** blok.

5. Napišite C# kôd klase **Nesreca** koja mora imati tri privatna atributa: dužina (broj s plivajućim zarezom), sirina (broj s plivajućim zarezom) i policijska uprava (niz znakova).

Klasa treba imati konstruktor koji kao parametre prima sve atribute klase i javnu metodu **ispis** koja ispisuje sadržaj klase u formatu „Dužina: **X**, širina: **Y** i policijska uprava: **Z**“, pri čemu znakove **X**, **Y** i **Z** treba zamijeniti s odgovarajućim vrijednostima.

U glavnom dijelu programa učitajte **20** proizvoljnih instanci klase **Nesreca** koristeći **petlju** po vlastitom izboru i ispišite sadržaj pojedine instance klase u konzolu koristeći napisanu metodu **ispis**.



Ispit #91 (XX.XX.24. - demonstrature)

Napomena: nedostaje prvi zadatak.

U drugom zadatku nisu stavili kod Main metode, a time se u zadatku ne bi ispisalo ništa. Zbog toga sam za taj zadatak izmislio Main metodu koju sam prilagodio onome što su ponudili kao rješenje zadatka.

2. Što će se ispisati nakon izvršavanja sljedećeg programskog odsječka? Priznaje se samo rješenje u kojem je vidljiv sadržaj varijabli.

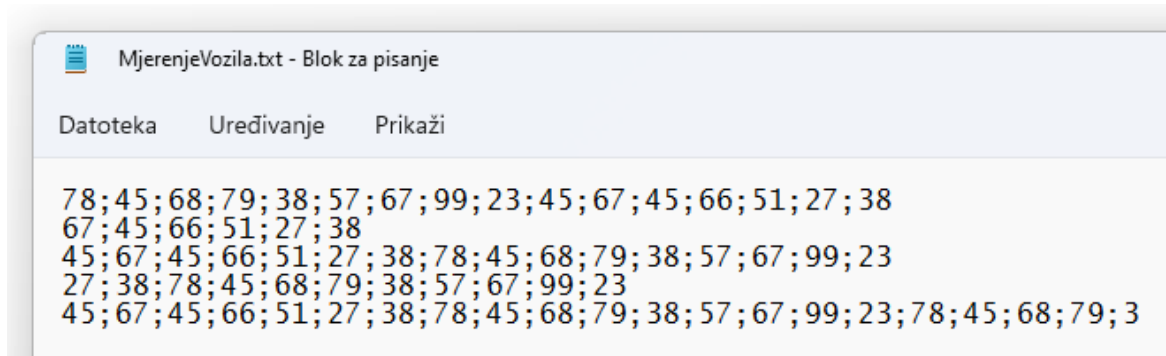
```
1 static void Main(string[] args)
2 {
3     PravokutanTrokut p = new PravokutanTrokut(4, 8);
4     p.izracunajPovrsinu();
5     Console.WriteLine(p.izracunajPovrsinu(2, 1));
6     p.ispis();
7 }
8
9 class PravokutanTrokut
10 {
11     private double A;
12     private double B;
13     private double povrsina;
14
15     public PravokutanTrokut(double a, double b)
16     {
17         A = a;
18         B = b;
19         povrsina = 0;
20         ispis();
21     }
22
23     public double izracunajPovrsinu(int a, int b)
24     {
25         povrsina = (b * a) / 2;
26         ispis();
27         return povrsina;
28     }
29
30     public void izracunajPovrsinu()
31     {
32         povrsina = (A * B) / 2;
33         ispis();
34     }
35
36     public void ispis()
37     {
38         Console.WriteLine(povrsina);
39     }
40 }
```



3. Napišite C# kôd metode *ProvjeriMatricu* koja kao argument prima dvodimenzionalnu matricu cijelih brojeva *M*. Metoda vraća bool vrijednost *istina* ako su svi elementi matrice veći od nula, u suprotnom metoda vraća vrijednost *laž*.

Unutar glavnog dijela programa potrebno je deklarirati matricu proizvoljne veličine, pozvati metodu te ovisno o vraćenoj vrijednosti ispisati: „*Matrica ne sadrži negativne vrijednosti!*“, ili „*Matrica sadrži negativne vrijednosti!*“.

4. Napišite C# kôd programa za učitavanje datoteke **MjerenjeVozila.txt** prikazane na sljedećoj slici smještene u direktoriju **D:**.

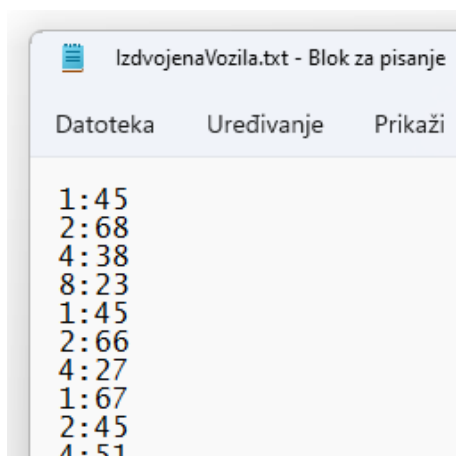


```
MjerenjeVozila.txt - Blok za pisanje
Datoteka   Uređivanje   Prikaži
78;45;68;79;38;57;67;99;23;45;67;45;66;51;27;38
67;45;66;51;27;38
45;67;45;66;51;27;38;78;45;68;79;38;57;67;99;23
27;38;78;45;68;79;38;57;67;99;23
45;67;45;66;51;27;38;78;45;68;79;38;57;67;99;23;78;45;68;79;3
```

U datoteci **MjerenjeVozila.txt** u pojedinom retku su izmjerene brzine vozila (cijeli brojevi) međusobno odvojeni točka-zarezom (;) na određenom dijelu prometnice u određeno vrijeme. Broj zabilježenih brzina po retku je varijabilan.

Program treba u novu datoteku **IzdvojenaVozila.txt**, smještenu u istom direktoriju **D:**, zapisati pojedine brzine iz svakog retka u formatu **X:Y** (u zasebnom retku), gdje znak **X** predstavlja indeks brzine u retku, a znak **Y** vrijednost te brzine. Iz svakog retka datoteke **MjerenjeVozila.txt** uzimaju se brzine na sljedećim indeksima 1, 2, 4, 8, 16, 32, ..., ∞ , odnosno dok god ima elemenata u retku na pojedinom indeksu (pomoć: eksponencijalna funkcija).

Potrebno je koristiti **try-catch** blok. Primjer zapisa u datoteci prikazan je na sljedećoj slici.



```
IzdvojenaVozila.txt - Blok za pisanje
Datoteka   Uređivanje   Prikaži
1:45
2:68
4:38
8:23
1:45
2:66
4:27
1:67
2:45
4:51
```



5. Napišite C# kôd klase **Ured** koja mora imati tri javna atributa: površina (broj s plivajućim zarezom), brojStolova (cijeli broj) i ID (cijeli broj – identifikacijski broj).

Klasa mora imati konstruktor preko kojeg se dohvaćaju i spremaju sve tri vrijednosti atributa u objekt klase.

Klasa dodatno treba imati javnu metodu **ispis** koja ispisuje sadržaj klase u formatu *Površina: X, broj stolova: Y i ID: Z*, pri čemu znakove X, Y i Z treba zamijeniti s odgovarajućim vrijednostima.

U glavnom dijelu programa učitajte **2** proizvoljne instance klase **Ured** koristeći **petlju** po vlastitom izboru, dodajte ih u listu proizvoljnog naziva i ispišite sadržaj pojedine instance klase u konzolu koristeći napisanu metodu **ispis**.

Primjer unosa i ispisa (napomena: format unosa/ispisa je proizvoljan):

Primjer izvođenja programa:

1. Unesite površinu ureda na indeksu 0: 55.5
2. Unesite broj stolova ureda na indeksu 0: 3
3. Unesite identifikacijski broj ureda na indeksu 0: 17
4. (ispis) Površina: 55.5, broj stolova: 3 i ID: 17
5. Unesite površinu ureda na indeksu 1: 34.67
6. Unesite broj stolova ureda na indeksu 1: 2
7. Unesite identifikacijski broj ureda na indeksu 1: 0
8. (ispis) Površina: 34.67, broj stolova: 2 i ID: 0



Ispit #92 (11.07.24. - demonstrature)

1. Nadopunite sljedeći program na način da pronađete prosječnu neparnu (cijeli broj) vrijednost elemenata u polju i ispišite je u konzolu (petlja po izboru).

```
1 static void Main(string[] args)
2 {
3     int[] polje = new int[] { 2, 6, 3, 4, 5 };
4
5     //Nadopuni:
6
7 }
```

2. Što će se ispisati u konzoli nakon izvršavanja sljedećeg programskog odsječka? Priznaje se samo rješenje u kojem je vidljiv postupak.

```
1 static void Main(string[] args)
2 {
3     MAYDAY_MAYDAY mayday = new MAYDAY_MAYDAY();
4     mayday.DistressCall();
5 }
6 public class MAYDAY_MAYDAY
7 {
8     public string[] nautical_alphabet = new string[]
9     {
10         "Alpha", "Bravo", "Charlie", "Delta", "Echo", "Foxtrot", "Golf", "Hotel",
11         "India", "Juliett", "Kilo", "Lima", "Mike", "November", "Oscar", "Papa",
12         "Quebec", "Romeo", "Sierra", "Tango", "Uniform", "Victor", "Whiskey",
13         "X-ray", "Yankee", "Zulu"
14     };
15     public void Method886()
16     {
17         Console.WriteLine(nautical_alphabet[4]);
18         Method675();
19     }
20     public void DistressCall()
21     {
22         Method543();
23         Method1337();
24     }
25     public void Method1234()
26     {
27         Console.WriteLine(nautical_alphabet[0]);
28         Method433();
29     }
30     public void Method1337()
31     {
32         Console.WriteLine("<->");
33         return;
34         Method456();
35     }
36     public void Method543()
37     {
38         Console.WriteLine(nautical_alphabet[3]);
```



```

39         Method335();
40     }
41     public void Method433()
42     {
43         Console.WriteLine(nautical_alphabet[18]);
44     }
45     public void Method335()
46     {
47         Method886();
48         Method456();
49     }
50     public void Method675()
51     {
52         Console.WriteLine(nautical_alphabet[12]);
53     }
54     public void Method456()
55     {
56         Console.WriteLine(nautical_alphabet[14]);
57         Method433();
58     }
59 }

```

3. Napišite konzolnu aplikaciju koja generira 10 realnih brojeva unutar intervala [0, 50] koji predstavljaju simulirane vrijednosti brzine komaraca te ih zapisuje u datoteku **brzine.txt**. Nije potrebno daodati imenski prostor *System.IO*.

4. Napišite C# kôd programa koji preko tipkovnice popunjava matricu cijelih brojeva **M = 3 x 2** s proizvoljnim vrijednostima (cijelim brojevima).

5. Napišite program koji učitava tri (3) znakovna niza (string) i pohranjuje ih u 1-D listu, a zatim pronalazi najkraći znakovni niz u polju te ga ispisuje (pomoć: za provjeru duljine znakovni niz promatrajte kao polje). Format upita i ispisa prikazan je u sljedećem primjeru:

Unesite 1. znakovni niz: Demos

Unesite 2. znakovni niz: Spasi

Unesite 3. znakovni niz: Nas

Najkraći znakovni niz sadrži 3 znaka, pri čemu je najkraći niz: Nas.



Ispit #93 (11.07.24.)

1. Nadopunite C# kôd programa koji preko tipkovnice popunjava matricu niza cijelih brojeva $M = 2 \times 2$ s proizvoljnim vrijednostima (cijelim brojevima).

Nakon unosa broja u matricu, program treba provjeriti da li je broj strogo duži od 3 znamenke (pomoć: za lakšu provjeru duljine broj se može pretvoriti u tip podataka za niz znakova). Za svaki broj matrice treba ispisati koliko znamenaka sadrži u formatu: Broj *BROJ* sadrži *X* znamenaka (znakove *BROJ* i *X* potrebno je zamijeniti odgovarajućim vrijednostima). Ukoliko broj ima manje ili jednako 3 znamenke, potrebno je uz ispis dodati da je broj neispravan.

Primjer unosa i ispisa (napomena: format unosa/ispisa je proizvoljan):

Primjer izvođenja programa:

1. Unesite broj na poziciji (0,0): 1337
2. (ispis) Broj 1337 sadrži 4 znamenke!
3. Unesite broj na poziciji (0,1): 630
4. (ispis) Broj 630 sadrži 3 znamenke i broj je neispravan!
5. Unesite broj na poziciji (1,0): 6300
6. (ispis) Broj 6300 sadrži 4 znamenke!
7. Unesite broj na poziciji (1,1): 1862
8. (ispis) Broj 1862 sadrži 4 znamenke!

2. Napišite program koji učitava tri (3) znakovna niza (string) i pohranjuje ih u 1-D polje, a zatim pronalazi najduži znakovni niz u polju te ga ispisuje zajedno s brojem znakova (pomoć: za provjeru duljine znakovni niz promatrajte kao polje).

Format upita i ispisa prikazan je u sljedećem primjeru:

Unesite 1. znakovni niz: Algoritmi

Unesite 2. znakovni niz: Su

Unesite 3. znakovni niz: Zabavni

Najdulji znakovni niz sadrži 9 znakova, pri čemu je najdulji niz: Algoritmi.

3. Napišite konzolnu aplikaciju koja generira 1000 realnih brojeva unutar intervala $[0, 100>$ koji predstavljaju simulirane vrijednosti brzine te ih zapisuje u datoteku **brzine.txt**.

Nije potrebno dodati imenski prostor *System.IO*.



4. Što će se ispisati u konzoli nakon izvršavanja sljedećeg programskog odsječka? Priznaje se samo rješenje u kojem je vidljiv postupak.

```
1 static void Main(string[] args)
2 {
3     MAYDAY_MAYDAY mayday = new MAYDAY_MAYDAY();
4     mayday.DistressCall();
5 }
6 public class MAYDAY_MAYDAY
7 {
8     public string[] nautical_alphabet = new string[]
9     {
10         "Alpha", "Bravo", "Charlie", "Delta", "Echo", "Foxtrot", "Golf", "Hotel",
11         "India", "Juliett", "Kilo", "Lima", "Mike", "November", "Oscar", "Papa",
12         "Quebec", "Romeo", "Sierra", "Tango", "Uniform", "Victor", "Whiskey",
13         "X-ray", "Yankee", "Zulu"
14     };
15     public void Method886()
16     {
17         Console.WriteLine(nautical_alphabet[11]);
18         Method675();
19     }
20     public void DistressCall()
21     {
22         Method543();
23         Method1337();
24     }
25     public void Method1337()
26     {
27         return;
28         Method456();
29     }
30     public void Method543()
31     {
32         Console.WriteLine(nautical_alphabet[0]);
33         Method335();
34     }
35     public void Method433()
36     {
37         Console.WriteLine(nautical_alphabet[17]);
38         Method566();
39     }
40     public void Method335()
41     {
42         Method886();
43         Method456();
44     }
45     public void Method675()
46     {
47         Console.WriteLine(nautical_alphabet[6]);
48     }
49     public void Method456()
50     {
51         Console.WriteLine(nautical_alphabet[15]);
52         Method433();
53     }
```



```
54     public void Method566()  
55     {  
56         Console.WriteLine(nautical_alphabet[14]);  
57     }  
58 }
```

5. Nadopunite sljedeći program na način da pronađete sumu parnih vrijednosti elemenata u polju i ispišite ju na konzolu (petlja po izboru).

```
1 static void Main(string[] args)  
2 {  
3     int[] polje = new int[] { 8, 2, 1, 9, 7 };  
4  
5     //Nadopuni:  
6  
7 }
```



Ispit #94 (28.08.24. - demonstrature)

1. Nadopunite C# kôd programa koji učitava ocjene za pet studenata te računa prosjek ocjena i ispisuje odgovarajuću poruku. Nije potrebno vršiti provjeru da li je učitana dobra ocjena, očekuje se od korisnika da unese ispravnu ocjenu.

Ako je prosjek ocjena u intervalu $[0, 1.5>$ potrebno je ispisati „Nedovoljan (1)!“.

Ako je prosjek ocjena u intervalu $[1.5, 2.5>$ potrebno je ispisati „Dovoljan (2)!“.

Ako je prosjek ocjena u intervalu $[2.5, 3.5>$ potrebno je ispisati „Dobar (3)!“.

Ako je prosjek ocjena u intervalu $[3.5, 4.5>$ potrebno je ispisati „Vrlo dobar (4)!“.

Inače je potrebno ispisati „Odličan (5)!“.

2. Što će se ispisati u konzoli nakon izvršavanja sljedećeg programskog odsječka? Priznaje se samo rješenje u kojem je vidljiv postupak.

```
List<int> lista = new List<int>();  
lista.Add(-1);  
lista.AddRange(new int[] { -1, 0, 1 });  
lista.RemoveAt(lista.Count - 2);  
lista.Insert(lista.Count - 2, 2);  
for (int i = 0; i < lista.Count; i++)  
{  
    Console.WriteLine(lista[i]);  
}  
lista.Clear();
```

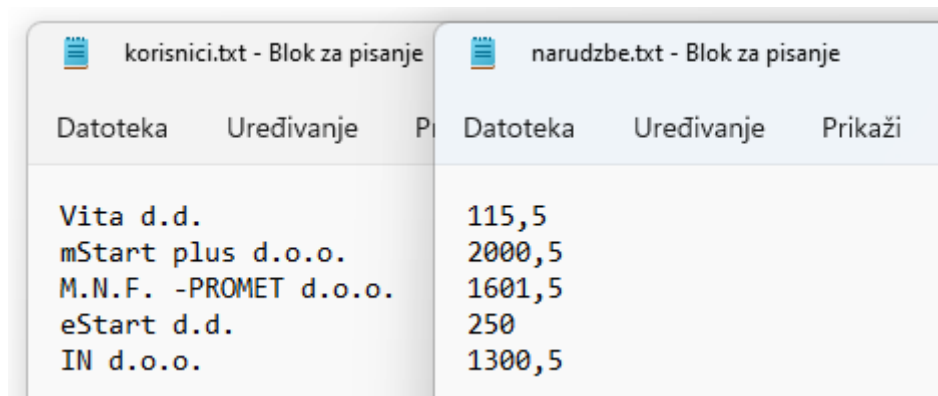
3. Napišite konzolnu aplikaciju koja učitava 1000 cijelih slučajnih brojeva unutar intervala $<0, 100>$.

U dvije liste spremaju se brojevi ovisno o njihovoj vrijednosti. Ako je slučajan broj djeljiv s 2 i s 3 te nije djeljiv s 5 dodaje ga se u jednu listu, inače ga se dodaje u drugu listu.

Na kraju je potrebno ispisati u konzolu koliko ima elemenata u pojedinoj listi u proizvoljnom formatu.



4. Napišite C# kôd programa koja učitava podatke iz dvije datoteke *korisnici.txt* i *narudzbe.txt*. Primjer izgleda datoteka prikazan je na sljedećoj slici:



The image shows two side-by-side text editors. The left editor, titled 'korisnici.txt - Blok za pisanje', contains a list of company names. The right editor, titled 'narudzbe.txt - Blok za pisanje', contains a list of numerical values corresponding to the companies in the first editor. Both editors have a menu bar with 'Datoteka', 'Uređivanje', and 'Prikaži'.

Datoteka	Uređivanje	Prikaži
Vita d.d.	115,5	
mStart plus d.o.o.	2000,5	
M.N.F. -PROMET d.o.o.	1601,5	
eStart d.d.	250	
IN d.o.o.	1300,5	

U datoteci *korisnici.txt* zapisani su nazivi korisnika, a u datoteci *narudzbe.txt* u odgovarajućem retku količina njihove narudžbe (realan broj).

Program iz svake datoteke treba čitati svaki drugi redak.

Program treba ispisati korisnika i njegovu narudžbu u formatu *Korisnik X je naručio Y paketa!* (znakove X i Y potrebno je zamijeniti s odgovarajućim vrijednostima) te sumu učitanih narudžbi.

Datotekama se pristupa preko relativne putanje.

Napomena: potrebno je koristiti *try-catch* blok te nije potrebno dodati imenski prostor *System.IO*.

5. Napišite konzolnu aplikaciju koja učitava matricu udaljenosti između n gradova.

Potrebno je preko tipkovnice unijeti vrijednost n , potom kreirati matricu $M = n \times n$, te ju popuniti realnim brojevima koji predstavljaju udaljenosti između gradova.

Na kraju je potrebno ispisati prosječnu vrijednost svakog stupca matrice u proizvoljnom formatu.



Ispit #95 (29.08.24.)

1. Napišite C# kôd programa koji učitava 1000 cijelih slučajnih brojeva unutar intervala [0, 250>, a koji predstavljaju trenutni broj automobila na nekom parkiralištu.

Ako je broj automobila na parkiralištu veći od 200, broj se sprema u listu *zagusenje*, a ako je manji i različit od 0 sprema se u listu *protocno*. U slučaju da je broj automobila 0, parkiralište je zatvoreno i taj broj se ne dodaje niti u jednu listu.

Na kraju je potrebno ispisati u konzolu koliko ima elemenata u pojedinoj listi u proizvoljnom formatu te koliko puta je parkiralište bilo zatvoreno.

2. Napišite C# kôd programa koja učitava podatke iz dvije datoteke *korisnici.txt* i *narudzbe.txt*. Primjer izgleda datoteka prikazan je na sljedećoj slici:

 korisnici.txt - Blok za pisanje

Datoteka	Uređivanje	Prikaži
Vita d.d.		115,5
mStart plus d.o.o.		2000,5
M.N.F. -PROMET d.o.o.		1601,5
eStart d.d.		250
IN d.o.o.		1300,5

 narudzbe.txt - Blok za pisanje

115,5
2000,5
1601,5
250
1300,5

U datoteci *korisnici.txt* zapisani su nazivi korisnika, a u datoteci *narudzbe.txt* u odgovarajućem retku količina njihove narudžbe (realan broj).

Program nakon čitanja podataka iz datoteke na konzolu treba ispisati naziv korisnika s najvećom količinom narudžbe.

Datotekama se pristupa preko relativne putanje.

Napomena: potrebno je koristiti *try-catch* blok te nije potrebno dodati imenski prostor *System.IO*.

3. Što će se ispisati u konzoli nakon izvršavanja sljedećeg programskog odsječka? Priznaje se samo rješenje u kojem je vidljiv postupak.

```
List<int> lista = new List<int>();
lista.Add(0);
lista.AddRange(new int[] { -2, 0, 3 });
lista.Remove(0);
lista.Insert(1, lista.Count - 1);
lista.RemoveAt(lista.Count - 3);
for (int i = 0; i < lista.Count; i++)
{
    Console.WriteLine(lista[i]);
}
lista.Clear();
Console.WriteLine(lista.Count);
```



4. Napišite C# kôd programa koji učitava dva broja s tipkovnice n i m koji predstavljaju redom broj studenata koji je pristupio ispitu i broj zadataka na ispitu te kreira matricu veličine $n \times m$.

U matricu se redom upisuju s tipkovnice bodovi studenata na ispitu po pojedinom zadatku (jedan redak matrice su bodovi jednog studenta na ispitu za m zadataka).

Nakon učitavanja u proizvoljnom formatu potrebno je ispisati za svakog studenta indeks zadatka na kojem je student ostvario najviše bodova. Dodatno, potrebno je ispisati indeks studenta koji ima najveće odstupanje od vlastitog prosječnog broja bodova po zadatku.

5. Napišite C# kôd programa koji učitava dva realna broja s tipkovnice r i v koji redom predstavljaju polumjer baze i visinu stošca.

Potrebno je napisati metodu *izracunajVolumen* koja kao argumente prima polumjer i visinu, a vraća izračunati volumen stošca.

U glavnom dijelu programa potrebno je ispisati volumen stošca samo ako su sve učitane vrijednosti veće od 0, inače treba ispisati *Nisu učitane dobre vrijednosti!*.

Format ispisa mora biti kao u primjeru.

Primjer: ako se redom učitaju vrijednosti 2.4 i 4 potrebno je ispisati: *Volumen stošca određenog polumjerom $r = 2.5$ i visinom $v = 4$ je 26.18!*

Izraz za izračun volumena stošca dan je izrazom:

$$V = \frac{r^2 \times \pi \times v}{3}$$



Ispit #96 (09.09.24. - demonstrature)

1. Napišite program koji učitava 8 prirodnih brojeva te ih sprema u polje.

Potrebno je ispisati sve brojeve na parnim pozicijama u polju koji su veći od srednje vrijednosti svih brojeva u polju.

Primjer:

Ulaz: 3, 7, 2, 8, 5, 1, 6, 4

Izlaz: Brojevi na parnim pozicijama veći od srednje vrijednosti (4.5): 7, 8

2. Napišite metodu Hipotenuza koja prima dvije duljine stranica pravokutnog trokuta te vraća vrijednost hipotenuze.

Dohvaćeni rezultat ispišite u proizvoljnom formatu.

Napraviti poziv metode Hipotenuza.

Primjer:

Ulaz: 3, 4

Izlaz: Hipotenuza trokuta sa stranicama 3 i 4 je 5.0

3. Napišite program koji provjerava je li učitani broj prost broj. Ako broj nije prost, od korisnika se traži unos novog broja sve dok se ne unese prost broj.

Nakon unos prostog broja, program ispisuje sve proste brojeve manje od unesenog broja.

Primjer:

Ulaz: 4, 6, 11

Izlaz: 2, 3, 5, 7

4. Napišite program koji učitava duljinu putovanja d u kilometrima (realan broj) i ID tipa prijevoza (cijeli broj). ID tipa prijevoza je dan stupcem ID u sljedećoj tablici:

ID	Brzina
1	15
2	80
3	120

Početna točka putovanja je (0,0), a prijevozni put ovisi o brzini prijevoznog sredstva. Program treba izračunati i ispisati vrijeme putovanja na osnovu unesene duljine puta i brzine vozila.

Primjer:

Ulaz: $d = 120\text{km}$ i vozilo pod ID-em 3

Izlaz: Vlak će prijeći 120 km za 1.5 sata!



5. Napišite C# kôd rekurzivne metode naziva *matemackiIzraz* koja računa matematički izraz:

$$r = \sum_{i=1}^n ((i)^{i-1} + 5)$$

Unutar *Main* metode potrebno je učitati prirodan broj n te pozvati rekurzivnu metodu za izračun navedenog matematičkog izraza s argumentom n . Ispis rezultata napraviti unutar *Main* metode u proizvoljnom formatu.

Primjer: za $n=3$:

$$r = (1^{1-1} + 5) + (2^{2-1} + 5) + (3^{3-1} + 5) =$$



Ispit #97 (10.09.24.)

1. Napišite program koji provjerava je li učitani broj djeljiv s **5**. Ako vrijednost broja nije djeljiva s **5** od korisnika se traži unos novog broja sve dok se ne unese broj djeljiv s **5**. Nakon unosa ispravne vrijednosti program ispisuje brojeve u intervalu od **9** do unesenog broja djeljive s **9**.

Primjer tijeka programa:

Unesite broj: 9

Unesite broj: 18

Unesite broj: 35

Ispis brojeva djeljivih s 9 do broja 35: 9, 18, 27



2. Što će se ispisati u konzoli nakon izvršavanja sljedećeg programskog odsječka? U zadatku je potrebno napisati redoslijed i vrijednosti koje se ispisuju.

```
static void Main(string[] args)
{
    Player ReadyPlayerOne = new Player("WadeWatts");
    Console.WriteLine(RPSimulator(ReadyPlayerOne));
}
public static int RPSimulator(Player tempPlayer)
{
    int modifier = 2;
    if (tempPlayer.GetHealth() % 2 == 0)
    {
        tempPlayer.Damage(tempPlayer.GetHealth() / 10 - modifier);
        tempPlayer.BroadcastHealth();
    }
    else
        return tempPlayer.GetHealth() + modifier;
    if (tempPlayer.GetHealth() % 3 == 0)
    {
        tempPlayer.Heal(tempPlayer.GetHealth() / 2);
        tempPlayer.BroadcastHealth();
    }
    else
    {
        tempPlayer.Damage(tempPlayer.GetHealth() % 2);
        return RPSimulator(tempPlayer);
    }
    return RPSimulator(tempPlayer);
}
public class Player
{
    private int health = 100;
    public string name;
    public Player()
    {
        Console.WriteLine("Welcome to the OASIS!");
    }
    public Player(string Name)
    {
        name = Name;
    }
    public void BroadcastHealth()
    {
        Console.WriteLine(health);
    }
    public int GetHealth()
    {
        return health;
    }
    public void Damage(int incomingDmg)
    {
        health -= incomingDmg;
    }
    public void Heal(int incomingHealing)
    {
        health += incomingHealing;
    }
}
```



3. Napišite metodu **SumaKvadrata** koja prima dva prirodna broja te vraća korijen sume kvadrata predanih brojeva. Dohvaćeni rezultat ispišite u proizvoljnom formatu. Potrebno je učitati podatke s tipkovnice (konzola) i zatim napraviti poziv metode **SumaKvadrata**.

4. Napišite program koji učitava 9 prirodnih brojeva te ih sprema u matricu **3 x 3**. Potrebno je u proizvoljnom formatu ispisati sve parne brojeve na **dijagonali** matrice. Primjer matrice u kojoj je potrebno ispisati brojeve **6** i **10**.

i/j	0	1	2
0	1	3	5
1	5	6	5
2	3	4	10

5. Napišite program koji preko konzole učitava vrijeme putovanja t u satima (realan broj) i ID tipa vozila (cijeli broj).

ID tipa vozila dan je stupcem ID u tablici.

ID	Opis	Brzina [°/h]
1	Motor	$v_x = v_y = 0.7622$
2	Automobil	$v_x = v_y = 0.5717$
3	Kamion	$v_x = v_y = 0.5082$

Početna pozicija vozila u geografskim stupnjevima je $T(x_p, y_p) = (15.97790, 45.810857)$.

Ovisno o tome koji je ID vozila unesen potrebno je ispisati poziciju na kojoj će se vozilo naći nakon što se vozi t vremena, pri čemu su brzine zadane u tablici, a konačna pozicija $T(x_k, y_k)$ se računa koristeći izraze:

$$x_k = v_x * t + x_p$$

$$y_k = v_y * t + y_p$$

Format ispisa mora biti kao u primjeru. Primjerice, ukoliko se učitava vrijeme od 2 sata i vozilo pod ID-em 1, potrebno je ispisati: *Motor će se nakon 2 sata vožnje nalaziti na poziciji T (17.502305, 47.335257)!*.



Dodatni zadaci

Radi se o zadacima s pismenih i usmenih ispita koje nisam mogao smjestiti u neki od ispitnih rokova jer sam dobio samo dijelove rokova. Uglavnom se radi o rokovima iz 2017. godine ili starijima od toga.

1. Nadopunite tablicu s odgovarajućim tipovima podataka potrebnim za spremanje navedenih vrijednosti.

Vrijednost	Tip podatka
3.14f	
"3.14"	
3	
'3'	

2. Napišite konzolnu aplikaciju koja učitava dva cijela broja i izračunava njihovu sumu. Format ispisa rezultata mora izgledati kao na sljedećem primjeru „5 + 5 = 10“.

3. Napišite konzolnu aplikaciju koja učitava cijeli broj preko tipkovnice. Zatim za taj broj ispituje da li je broj djeljiv s 9 i s 4. Ako je broj djeljiv s 9 i 4 potrebno je ispisati poruku „Broj x je djeljiv s 9 i 4“, a ako nije „Broj x nije djeljiv s 9 i 4“. Varijabla „x“ predstavlja broj koji se učitava preko tipkovnice.

4. Napišite konzolnu aplikaciju koja učitava cijeli broj preko tipkovnice. Zatim za taj broj ispituje da li je broj djeljiv s 3 ili s 8. Ako je broj djeljiv s 3 ili 8 potrebno je ispisati poruku „Broj x je djeljiv s 3 ili 8“, a ako nije „Broj x nije djeljiv s 3 ili 8“. Varijabla „x“ predstavlja broj koji se učitava preko tipkovnice.

5. Nadopunite sljedeći program koji izračunava prosječnu vrijednost svih elemenata polja i ispisuje rezultat u konzolu.

```
1 static void Main(string[] args)
2 {
3     int[] polje = new int[4] { 6, 8, 2, 1 };
4     //Nadopuni
5 }
```



Ispitni rokovi – rješenja

Ispit #1 (XX.XX.XX. - grupa A)

Zadatak 1

Nadopunite tablicu s odgovarajućim tipovima podataka potrebnim za spremanje navedenih vrijednosti.

Vrijednost	Tip podatka
"false"	
"2"	
2	
'2'	

Rješenje:

Vrijednost	Tip podatka
"false"	string
"2"	string
2	int
'2'	char

Zadatak 2

Napišite konzolnu aplikaciju koja učitava jedan cijeli broj i provjerava da li je taj broj djeljiv s 3. Ako je broj djeljiv ispisuje poruku „Broj je djeljiv s 3“, a ako nije „Broj nije djeljiv s 3“.

Rješenje:

```
static void Main(string[] args)
{
    Console.WriteLine("Unesite cijeli broj: ");
    int x = Convert.ToInt32(Console.ReadLine());

    if (x % 3 == 0)
    {
        Console.WriteLine("Broj je djeljiv s 3");
    }
    else
    {
        Console.WriteLine("Broj nije djeljiv s 3");
    }

    Console.ReadKey();
}
```



Zadatak 3

Korištenjem algoritma za jednostavno sortiranje umetanjem (insertion sort) nadopunite ostatak koraka algoritma.

91	33	56	48	88	66	6
91	33	56	48	88	66	6

Rješenje:

91	33	56	48	88	66	6
91	33	56	48	88	66	6
33	91	56	48	88	66	6
33	56	91	48	88	66	6
33	48	56	91	88	66	6
33	48	56	88	91	66	6
33	48	56	66	88	91	6
6	33	48	56	66	88	91

Objašnjenje: skup elemenata za sortiranje dijeli se na sortirani i nesortirani. Svakim korakom povećava se sortirani dio tako da se iz nesortiranog dijela uzima prvi element i ubacuje ga se na odgovarajuće mjesto u sortiranom dijelu.

Zadatak 4

Napišite konzolnu aplikaciju koja traži unos središnjeg kuta i radijusa kružnice (realni brojevi) te računa površinu kružnog isječka prema formuli:

$$P = \frac{r^2 \pi \alpha}{360}$$

Rješenje:

```
static void Main(string[] args)
{
    Console.WriteLine("Unesite središnji kut kružnice: ");
    double a = Convert.ToDouble(Console.ReadLine());

    Console.WriteLine("Unesite radijus kružnice: ");
    double r = Convert.ToDouble(Console.ReadLine());

    double P = Math.Pow(r, 2) * Math.PI * a / 360;

    Console.WriteLine("P = " + P);

    Console.ReadKey();
}
```



Zadatak 5

Kreirajte polje od 200 elemenata koji su cijeli brojevi generatorom slučajnih brojeva od 20 do 30 i ispišite zadnji element polja.

Rješenje:

```
static void Main(string[] args)
{
    int[] polje = new int[200];

    Random rnd = new Random();

    for (int i = 0; i < polje.Length; i++)
    {
        polje[i] = rnd.Next(20, 31);
    }

    Console.WriteLine("Zadnji element polja: " + polje[polje.Length - 1]);

    Console.ReadKey();
}
```



Ispit #2 (XX.XX.XX. - grupa B)

Zadatak 1

Napišite konzolnu aplikaciju koja učitava dva cijela broja s tipkovnice (a i b). Ako je broj a neparan program ispisuje razliku brojeva a i b . Inače ispisuje umnožak učitanih brojeva. Ispis rezultata mora biti u formatu kao u primjeru.

Primjer: ako je za a učitani broj 9, a za b je učitani broj 2, ispis mora biti „Razlika brojeva 9 i 2 je 7“.

Rješenje:

```
static void Main(string[] args)
{
    Console.Write("Unesite cijeli broj a: ");
    int a = Convert.ToInt32(Console.ReadLine());

    Console.Write("Unesite cijeli broj b: ");
    int b = Convert.ToInt32(Console.ReadLine());

    if (a % 2 != 0)
    {
        int razlika = a - b;
        Console.WriteLine("Razlika brojeva " + a + " i " + b + " je " + razlika);
    }
    else
    {
        int umnozak = a * b;
        Console.WriteLine("Umnožak brojeva " + a + " i " + b + " je " + umnozak);
    }

    Console.ReadKey();
}
```



Zadatak 2

Napišite konzolnu aplikaciju koja ispisuje sve brojeve između 1 i n koji nisu djeljivi ni sa 3 ni sa 7, pri čemu se prirodan broj n učitava s tipkovnice.

Rješenje:

```
static void Main(string[] args)
{
    Console.Write("Unesite prirodan broj n: ");
    int n = Convert.ToInt32(Console.ReadLine());

    Console.WriteLine("Brojevi u rasponu [1, " + n
        + "] koji nisu djeljivi ni sa 3 ni sa 7 su:");

    for (int i = 1; i <= n; i++)
    {
        if (i % 3 != 0 && i % 7 != 0)
        {
            Console.WriteLine(i);
        }
    }

    Console.ReadKey();
}
```

Zadatak 3

Napišite metodu *ZbrojKvadrata* koja kao argument prima tri realna broja te vraća zbroj kvadrata predanih brojeva. Potrebno je napisati deklaraciju metode *ZbrojKvadrata*.

Rješenje:

```
static double ZbrojKvadrata(double x, double y, double z)
{
    double zbroj = Math.Pow(x, 2) + Math.Pow(y, 2) + Math.Pow(z, 2);
    return zbroj;
}
```



Zadatak 4

Nadopunite sljedeći program na način da učitajte cijeli broj n s tipkovnice te prebrojite koliko elemenata u polju je veće ili jednako n .

Primjer: ako se s tipkovnice učitava broj 6 potrebno je ispisati „2 broja u polju su veća ili jednaka 6“.

Koristite petlje prema vlastitom izboru.

```
static void Main(string[] args)
{
    int[] polje = new int[5] { 1, 2, 6, 7, 4 };

    //Nadopuni

}
```

Rješenje:

```
static void Main(string[] args)
{
    int[] polje = new int[5] { 1, 2, 6, 7, 4 };

    //Nadopuni
    Console.Write("Unesite cijeli broj n: ");
    int n = Convert.ToInt32(Console.ReadLine());

    int brojTrazenih = 0;

    for (int i = 0; i < polje.Length; i++)
    {
        if (polje[i] >= n)
        {
            brojTrazenih++;
        }
    }

    Console.WriteLine(brojTrazenih + " broja u polju su veća ili jednaka 6");

    Console.ReadKey();
}
```



Zadatak 5

Nadopunite kod u glavnoj metodi tako da se doda jedan novi student JMBAG-a 0135123123, Pero Perić. Uvećajte njegov broj ECTS bodova za 5.

```
static void Main(string[] args)
{
    //Nadopuni

}

public class Student
{
    public string JMBAG { get; set; }
    public string Prezime { get; set; }
    public string Ime { get; set; }
    public int Ects { get; set; }

    public Student(string jmbag, string prezime, string ime)
    {
        JMBAG = jmbag;
        Prezime = prezime;
        Ime = ime;
        Ects = 0;
    }

    public void UvecajEcts(int ects)
    {
        Ects += ects;
    }
}
```



Rješenje:

```
static void Main(string[] args)
{
    //Nadopuni

    Student s = new Student("0135123123", "Perić", "Pero");
    s.UvecajEcts(5);

    Console.ReadKey();
}

public class Student
{
    public string JMBAG { get; set; }
    public string Prezime { get; set; }
    public string Ime { get; set; }
    public int Ects { get; set; }

    public Student(string jmbag, string prezime, string ime)
    {
        JMBAG = jmbag;
        Prezime = prezime;
        Ime = ime;
        Ects = 0;
    }

    public void UvecajEcts(int ects)
    {
        Ects += ects;
    }
}
```



Ispit #3 (XX.XX.XX. - grupa C)

Zadatak 1

Napišite konzolnu aplikaciju koja učitava realan broj r koji predstavlja polumjer kruga. Ako je polumjer kruga manji ili jednak nuli aplikacija ispisuje "Niste unijeli dobar polumjer!", inače ispisuje površinu i opseg kruga kao u primjeru.

Npr. ako je za r učitani broj 2.5 ispis mora biti sljedeći: "Površina kruga polumjera 2.5 je $P = 19.635$, a opseg je $O = 15.7!$ ".

Rješenje:

```
static void Main(string[] args)
{
    Console.Write("Unesite polumjer kruga r: ");
    double r = Convert.ToDouble(Console.ReadLine());

    if (r > 0)
    {
        double O = 2 * r * Math.PI;
        double P = Math.Pow(r, 2) * Math.PI;

        Console.WriteLine("Površina kruga polumjera " + r + " je P = " + P
            + ", a opseg je O = " + O + "!");
    }
    else
    {
        Console.WriteLine("Niste unijeli dobar polumjer!");
    }

    Console.ReadKey();
}
```



Zadatak 2

Napišite metodu *Suma* koja kao argument prima dva prirodna broja te ispisuje sumu svih brojeva između dva predana broja (granice su uključujuće). Potrebno je napisati deklaraciju i implementaciju metode *Suma*.

Rješenje:

```
static void Suma(int x, int y)
{
    int suma = 0;

    for (int i = x; i <= y; i++)
    {
        suma += i;
    }

    Console.WriteLine("Suma svih brojeva u intervalu [" + x + ", " + y + "] je "
        + suma + ".");
}
```

Zadatak 3

Napišite konzolnu aplikaciju koja učitava sadržaj datoteke „*data.txt*“ u kojoj su upisani cijeli brojevi svaki u zasebnom retku. Program treba na kraju ispisati sumu svih učitanih cijelih brojeva.

Napomena: nije potrebno u program dodati *using System.IO* niti *try-catch* blok.

Rješenje:

```
using System.IO;

static void Main(string[] args)
{
    StreamReader reader = new StreamReader("data.txt");

    int suma = 0;

    while (!reader.EndOfStream)
    {
        string linija = reader.ReadLine();
        int x = Convert.ToInt32(linija);
        suma += x;
    }

    reader.Close();

    Console.WriteLine("Suma svih brojeva iz datoteke: " + suma);

    Console.ReadKey();
}
```



Zadatak 4

Napišite konzolnu aplikaciju koja učitava 1000 realnih vrijednosti u jednodimenzionalno polje te ispisuje koliko je elemenata polja negativno (prebrojava broj negativnih vrijednosti).

Rješenje:

```
static void Main(string[] args)
{
    double[] polje = new double[1000];

    int brojNegativnih = 0;

    for (int i = 0; i < polje.Length; i++)
    {
        Console.Write("Unesite " + (i + 1) + ". broj: ");
        int x = Convert.ToInt32(Console.ReadLine());

        polje[i] = x;

        if (polje[i] < 0)
        {
            brojNegativnih++;
        }
    }

    Console.WriteLine("Broj negativnih elemenata u polju: " + brojNegativnih);

    Console.ReadKey();
}
```



Zadatak 5

Nadopunite sljedeći kod unutar Main metode na način da stvorite novi kolegij naziva „ALGPRO“ na zavodu „ITS“ koji nosi 7 ECTS-a i ima 270 studenata. Potom pozovite metodu s kojom će se povećati broj studenata za 1.

```
static void Main(string[] args)
{
    //Nadopuni

}

public class Kolegij
{
    private string naziv;
    private string zavod;
    private int ects;
    private int brojStudenata;

    public Kolegij(string ZAVOD, string NAZIV, int ECTS,int BROJSTUDENATA)
    {
        naziv = NAZIV;
        zavod = ZAVOD;
        ects = ECTS;
        brojStudenata = BROJSTUDENATA;
    }

    public void UvecajBrojStudenata1(int brStud)
    {
        brojStudenata += brStud;
    }

    private void UvecajBrojStudenata2()
    {
        brojStudenata++;
    }
}
```



Rješenje:

```
static void Main(string[] args)
{
    //Nadopuni

    Kolegij k = new Kolegij("ITS", "ALGPRO", 7, 270);
    k.UvecajBrojStudenata1(1);

    Console.ReadKey();
}

public class Kolegij
{
    private string naziv;
    private string zavod;
    private int ects;
    private int brojStudenata;

    public Kolegij(string ZAVOD, string NAZIV, int ECTS, int BROJSTUDENATA)
    {
        naziv = NAZIV;
        zavod = ZAVOD;
        ects = ECTS;
        brojStudenata = BROJSTUDENATA;
    }

    public void UvecajBrojStudenata1(int brStud)
    {
        brojStudenata += brStud;
    }

    private void UvecajBrojStudenata2()
    {
        brojStudenata++;
    }
}
```



Ispit #9 (16.05.16. - grupa A)

Zadatak 1

Napišite konzolnu aplikaciju koja učitava polumjer kruga r (decimalni broj).

Ako je polumjer kruga $r \leq 0$, aplikacija ispisuje „Niste unijeli dobar polumjer!“.

Ako je polumjer kruga $r == 2.5$, aplikacija ispisuje površinu kruga 19.63“.

Ako je polumjer kruga $r > 0$ && $r \neq 2.5$, računa površinu kruga po unesenom polumjeru.

```
static void Main(string[] args)
{
    Console.Write("Unesite polumjer kruga r: ");
    double r = Convert.ToDouble(Console.ReadLine());

    if (r <= 0)
    {
        Console.WriteLine("Niste unijeli dobar polumjer !");
    }
    else if (r == 2.5)
    {
        Console.WriteLine("Površina kruga je 19.63 !");
    }
    else if (r > 0 && r != 2.5)
    {
        double površina = Math.Pow(r, 2) * Math.PI;
        Console.WriteLine("Površina kruga je " + površina + " !");
    }
    Console.ReadKey();
}
```



Zadatak 2

Napišite konzolnu aplikaciju koja ispisuje sve prirodne brojeve između n i m (učitavaju se s tipkovnice).

```
static void Main(string[] args)
{
    Console.Write("Unesite n: ");
    int n = Convert.ToInt32(Console.ReadLine());

    Console.Write("Unesite m: ");
    int m = Convert.ToInt32(Console.ReadLine());

    for (int i = n; i <= m; i++)
    {
        Console.WriteLine(i);
    }

    Console.ReadKey();
}
```



Zadatak 3

Napišite metodu Umnožak koja kao argument prima prirodni broj te vraća umnožak svih prirodnih brojeva do tog broja. Potrebno je napisati i deklaraciju metode Umnožak.

```
static void Main(string[] args)
{
    Console.Write("Unesite broj n: ");
    int n = Convert.ToInt32(Console.ReadLine());

    Console.WriteLine("Umnožak svih prirodnih brojeva od 1 do " + n + " je " + Umnozak(n) + " !");

    Console.ReadKey();
}

static int Umnozak(int broj)
{
    int umnozak = 1;

    for (int i = 1; i <= broj; i++)
    {
        umnozak = umnozak * i;
    }
    return umnozak;
}
```



Zadatak 4

Nadopunite sljedeći program na način da pronađete sumu parnih vrijednosti elemenata u polju i ispišite je u konzolu (petlja po izboru).

```
static void Main(string[] args)
{
    int[] polje = new int[5] { 8, 2, 1, 9, 7 };

    //Nadopuni

    int suma = 0;

    for (int i = 0; i < polje.Length; i++)
    {
        if (polje[i] % 2 == 0)
        {
            suma = suma + polje[i];
        }
    }

    Console.WriteLine("Suma parnih elemenata u polju je " + suma + ".");

    Console.ReadKey();
}
```



Zadatak 5

Nadopunite sljedeći zadatak tako što ćete upisati ime i prezime studenta Pero Perić, njegov JMBAG 0135231212 i dodati mu 5 ECTS bodova.

```
static void Main(string[] args)
{
    //Nadopuni

    Student student1 = new Student("0135231212", "Perić", "Pero");
    student1.UvecajEcts(5);

    Console.ReadKey();
}

public class Student
{
    public string JMBAG { get; set; }
    public string Prezime { get; set; }
    public string Ime { get; set; }
    public int Ects { get; set; }

    public Student(string jmbag, string prezime, string ime)
    {
        JMBAG = jmbag;
        Prezime = prezime;
        Ime = ime;
        Ects = 0;
    }

    public void UvecajEcts(int ects)
    {
        Ects += ects;
    }
}
```



Ispit #10 (16.05.16. - grupa B)

Zadatak 1

Napišite konzolnu aplikaciju koja učitava dva broja a i b.

Ako je broj $a \geq b$, treba izračunati njihovu sumu.

Ako je $a < b$, onda treba izračunati njihovu razliku.

```
static void Main(string[] args)
{
    Console.Write("Unesite broj a: ");
    int a = Convert.ToInt32(Console.ReadLine());

    Console.Write("Unesite broj b: ");
    int b = Convert.ToInt32(Console.ReadLine());

    if (a >= b)
    {
        int suma = a + b;
        Console.WriteLine(a + " + " + b + " = " + suma);
    }

    else if (a < b)
    {
        int razlika = a - b;
        Console.WriteLine(a + " - " + b + " = " + razlika);
    }

    Console.ReadKey();
}
```



Zadatak 2

Napišite konzolnu aplikaciju koja ispisuje sve parne brojeve između 1 i n. Prirodni broj n učitajte s tipkovnice.

```
static void Main(string[] args)
{
    Console.Write("Unesite broj n: ");
    int n = Convert.ToInt32(Console.ReadLine());

    Console.WriteLine("Svi parni brojevi između 1 i " + n + " su: ");
    for (int i = 1; i <= n; i++)
    {
        if (i % 2 == 0)
        {
            Console.WriteLine(i);
        }
    }

    Console.ReadKey();
}
```



Zadatak 3

Napišite metodu Suma koja kao argument prima dva prirodna broja te vraća sumu svih brojeva između dva prirodna broja.

Potrebno je napisati i deklaraciju metode Suma.

```
static void Main(string[] args)
{
    Console.Write("Unesite broj a: ");
    int a = Convert.ToInt32(Console.ReadLine());

    Console.Write("Unesite broj b: ");
    int b = Convert.ToInt32(Console.ReadLine());

    Console.WriteLine("Suma svih brojeva između " + a + " i " + b + " je: " + Suma(a, b));

    Console.ReadKey();
}

static int Suma(int x, int y)
{
    int zbroj = 0;

    for (int i = x; i <= y; i++)
    {
        zbroj = zbroj + i;
    }

    return zbroj;
}
```



Zadatak 4

Nadopunite sljedeći program na način da pronađete maksimalnu vrijednost elemenata u polju i ispišete tu vrijednost u konzolu.

```
static void Main(string[] args)
{
    int[] polje = new int[6] { 12, 10, 6, 8, 2, 1 };

    //Nadopuna

    int max = polje[0];

    for (int i = 0; i < polje.Length; i++)
    {
        if (polje[i] > max)
        {
            max = polje[i];
        }
    }
    Console.WriteLine("Maksimalna vrijednost je " + max + " !");

    Console.ReadKey();
}
```



Zadatak 5

Nadopunite sljedeći kod metodom "Ubrzaj" unutar klase Auto na način da njezini parametri odgovaraju pozivu unutar Main dijela programa. U tijelu metode potrebno je povećati atribut brzinu za vrijednost parametra kojeg metoda primi.

```
static void Main(string[] args)
{
    Auto auto1 = new Auto();
    auto1.Ubrzaj(5);

    Console.ReadKey();
}

public class Auto
{
    public int id { get; set; }
    public int Model { get; set; }
    public int Marka { get; set; }
    public int Brzina { get; set; }

    //Nadopuni

    public void Ubrzaj(int brzina)
    {
        Brzina += brzina;
    }
}
```



Ispit #11 (26.06.17. - grupa A)

Zadatak 1

Napišite konzolnu aplikaciju koja učitava broj bodova s ispita unutar intervala [0,5].

Ako učitani bodovi nisu unutar zadanog intervala program ispisuje "Niste unijeli bodove u intervalu [0,5]".

Ako su ispravno učitani bodovi potrebno je ispisati odgovarajuću poruku: ako je broj bodova unutar intervala $[0,2>$ (0 uključujuće, 2 isključujuće) potrebno je ispisati "Student je pao na ispitu!". Inače je potrebno ispisati "Student je prošao na ispitu!".

```
static void Main(string[] args)
{
    Console.Write("Unesite broj bodova s ispita: ");
    int bodovi = Convert.ToInt32(Console.ReadLine());

    if (bodovi < 0 || bodovi > 5)
    {
        Console.WriteLine("Niste unijeli bodove u intervalu [0,5]");
    }
    else if (bodovi >= 0 && bodovi < 2)
    {
        Console.WriteLine("Student je pao na ispitu!");
    }
    else
    {
        Console.WriteLine("Student je prošao na ispitu!");
    }

    Console.ReadKey();
}
```



Zadatak 2

Napišite metodu Ispis koja ispisuje sve brojeve djeljive s 4 u intervalu [m - n], pri čemu su m i n argumenti koji se predaju metodi.

Nije potrebno vršiti provjeru da li su učitane ispravne granice intervala.

Potrebno je napisati implementaciju i deklaraciju metode Ispis.

Poziv unutar Main metode nije potreban.

```
static void Main(string[] args)
{
    Console.Write("Unesite m: ");
    int m = Convert.ToInt32(Console.ReadLine());

    Console.Write("Unesite n: ");
    int n = Convert.ToInt32(Console.ReadLine());

    Ispis(m, n);

    Console.ReadKey();
}

static void Ispis(int m, int n)
{
    Console.WriteLine("Brojevi djeljivi sa 4 u rasponu od " + m + " do " +
        n + " su: ");

    for (int i = m; i <= n; i++)
    {
        if (i % 4 == 0)
        {
            Console.WriteLine(i);
        }
    }
}
```



Zadatak 3

Napišite konzolnu aplikaciju koja učitava sadržaj datoteke "dat.txt" u kojoj su upisani cijeli brojevi svaki u zasebnom retku.

Program treba ispisati sadržaj datoteke s ispisom svakog od učitanih cijelih brojeva u zasebnom retku.

```
using System.IO;

static void Main(string[] args)
{
    try
    {
        StreamReader reader = new StreamReader("dat.txt");

        List<int> lista = new List<int>();

        while (!reader.EndOfStream)
        {
            lista.Add(Convert.ToInt32(reader.ReadLine()));
        }

        reader.Close();

        Console.WriteLine(@"Učitani brojevi iz datoteke " + "dat.txt" + " su:");
        for (int i = 0; i < lista.Count; i++)
        {
            Console.WriteLine(lista[i]);
        }
    }
    catch (Exception ex)
    {
        Console.WriteLine("Greška: " + ex);
    }

    Console.ReadKey();
}
```



Zadatak 4

Napišite konzolnu aplikaciju koja učitava 10 cjelobrojnih vrijednosti u jednodimenzionalno polje, te računa i ispisuje prosječnu vrijednost učitanih elemenata polja.

```
static void Main(string[] args)
{
    int[] polje = new int[10];
    int zbroj = 0;

    for (int i = 0; i < polje.Length; i++)
    {
        Console.Write("Unesite " + (i + 1) + ". broj: ");
        polje[i] = Convert.ToInt32(Console.ReadLine());

        zbroj += polje[i];
    }

    double prosjek = zbroj / (polje.Length * 1.0);

    Console.WriteLine("Prosjek svih 10 brojeva u polju je: " + prosjek +
        ".");

    Console.ReadKey();
}
```



Zadatak 5

Nadopunite sljedeći kod s metodama UvecajKilometre i PromijeniVlasnika tako da njihovi parametri odgovaraju pozivu iz Main metode.

Metoda UvecajKilometre treba uvećati kilometre za zadani broj koji se predaje preko parametra, dok metoda PromijeniVlasnika treba promijeniti ime vlasnika u ime koje se predaje preko parametra.

```
static void Main(string[] args)
{
    Auto a = new Auto("Insignia", 50000, "Marko");
    a.UvecajKilometre(10000);
    a.PromijeniVlasnika("Ivo");
}

public class Auto
{
    private string imeAuta;
    private int kilometri;
    private string imeVlasnika;

    public Auto(string IMEAUTA, int KILOMETRI, string IMEVLASNIKA)
    {
        imeAuta = IMEAUTA;
        kilometri = KILOMETRI;
        imeVlasnika = IMEVLASNIKA;
    }

    //Nadopuni

    public void UvecajKilometre(int KILOMETRI)
    {
        kilometri += KILOMETRI;
    }

    public void PromijeniVlasnika(string IMEVLASNIKA)
    {
        imeVlasnika = IMEVLASNIKA;
    }
}
```



Ispit #12 (26.06.17. - grupa B)

Zadatak 1

Napišite konzolnu aplikaciju koja učitava dva cijela broja s tipkovnice, a i b.

Ako je broj a manji ili jednak b program ispisuje sumu brojeva, a ako je a veći od b program ispisuje razliku brojeva.

Ispis rezultata mora biti u formatu kao u primjeru.

Npr. ako je za a učitani broj 8, a za b je učitani broj 3 ispis mora biti sljedeći: "Razlika brojeva 8 i 3 je 5."

```
static void Main(string[] args)
{
    Console.Write("Unesite a: ");
    int a = Convert.ToInt32(Console.ReadLine());

    Console.Write("Unesite b: ");
    int b = Convert.ToInt32(Console.ReadLine());

    if (a <= b)
    {
        int suma = a + b;

        Console.WriteLine("Suma brojeva " + a + " i " + b + " je " + suma
            + ".");
    }
    else if (a > b)
    {
        int razlika = a - b;

        Console.WriteLine("Razlika brojeva " + a + " i " + b + " je " +
            razlika + ".");
    }

    Console.ReadKey();
}
```



Zadatak 2

Napišite metodu Suma koja kao argument prima dva prirodna broja m i n te vraća sumu svih brojeva između ta dva broja (granice su uključujuće).

Nije potrebno vršiti provjeru da li su učitani ispravni brojevi.

Potrebno je napisati implementaciju i deklaraciju metode Suma.

Poziv unutar Main metode nije potreban.

```
static void Main(string[] args)
{
    Console.Write("Unesite m: ");
    int m = Convert.ToInt32(Console.ReadLine());

    Console.Write("Unesite n: ");
    int n = Convert.ToInt32(Console.ReadLine());

    Console.WriteLine("Suma svih brojeva od " + m + " do " + n + " je " +
        Suma(m, n));

    Console.ReadKey();
}

static int Suma(int m, int n)
{
    int suma = 0;

    for (int i = m; i <= n; i++)
    {
        suma = suma + i;
    }

    return suma;
}
```



Zadatak 3

Napišite konzolnu aplikaciju koja učitava realan broj x s tipkovnice.

U datoteku naziva "dat.txt" potrebno je zapisati dvije vrijednosti: vrijednost učitanoog broja x i dvostruko veću vrijednost učitanoog broja x.

Svaki zapis mora biti u zasebnom retku.

```
static void Main(string[] args)
{
    Console.Write("Unesite x: ");
    double x = Convert.ToDouble(Console.ReadLine());

    StreamWriter writer = new StreamWriter("dat.txt");

    writer.WriteLine(x);
    writer.WriteLine(x * 2);

    writer.Flush();
    writer.Close();

    Console.ReadKey();
}
```



Zadatak 4

Napišite konzolnu aplikaciju koja učitava 10 cjelobrojnih vrijednosti u jednodimenzionalno polje, te pronalazi i ispisuje minimalnu vrijednost učitanih elemenata polja.

```
static void Main(string[] args)
{
    int[] polje = new int[10];

    for (int i = 0; i < polje.Length; i++)
    {
        Console.Write("Unesite " + (i + 1) + ". broj: ");
        polje[i] = Convert.ToInt32(Console.ReadLine());
    }

    int najmanji = polje[0];

    for (int i = 0; i < polje.Length; i++)
    {
        if (polje[i] < najmanji)
        {
            najmanji = polje[i];
        }
    }

    Console.WriteLine("Najmanji broj u polju je " + najmanji + ".");

    Console.ReadKey();
}
```



Zadatak 5

Nadopunite sljedeći kod odgovarajućim konstruktorom unutar klase Student na način da njezini parametri odgovaraju pozivu unutar Main metode.

```
static void Main(string[] args)
{
    Student st = new Student("Ivo", "Ivić", "ALGPRO", 3);
}

public class Student
{
    private string Ime;
    private string Prezime;
    private string Kolegij;
    private int Ocjena;

    //Nadopuni

    public Student(string IME, string PREZIME, string KOLEGIJ, int OCJENA)
    {
        Ime = IME;
        Prezime = PREZIME;
        Kolegij = KOLEGIJ;
        Ocjena = OCJENA;
    }
}
```



Ispit #13 (06.07.17. - grupa A)

Zadatak 1

Napišite konzolnu aplikaciju koja učitava dva realna broja D i a koji predstavljaju diskriminantu i vrijednost prvog koeficijenta kvadratne funkcije.

Ako su vrijednosti D i a manje od nule potrebno je ispisati: "Za D i a funkcija ne siječe x-os te su joj sve vrijednosti negativne!".

Ako je D manje od nule i a veće od nule potrebno je ispisati: "Za D i a funkcija ne siječe x-os te su joj sve vrijednosti pozitivne!".

Inače je potrebno ispisati: "Funkcija siječe x-os!".

Primjer: ako se za D učitava vrijednost -1.25. a za a vrijednost 0.5, potrebno je ispisati:

"Za $D = -1.25$ i $a = 0.5$ funkcija ne siječe x-os te su joj sve vrijednosti pozitivne!".

```
static void Main(string[] args)
{
    Console.Write("Unesite diskriminantu D: ");
    double D = Convert.ToDouble(Console.ReadLine());

    Console.Write("Unesite parametar a: ");
    double a = Convert.ToDouble(Console.ReadLine());

    if (D < 0 && a < 0)
    {
        Console.WriteLine("Za D = " + D + " i a = " + a + " funkcija ne siječe x-os te su joj sve vrijednosti pozitivne!");
    }
    else if (D < 0 && a > 0)
    {
        Console.WriteLine("Za D = " + D + " i a = " + a + " funkcija ne siječe x-os te su joj sve vrijednosti pozitivne!");
    }
    else
    {
        Console.WriteLine("Funkcija siječe x-os!");
    }

    Console.ReadKey();
}
```



Zadatak 2

Napišite metodu PotenciranjeRazlike koja kao argument prima tri realna broja.

Metoda razliku prva dva broja potencira s trećim brojem te vraća rezultat izvršene operacije.

Potrebno je napisati deklaraciju i implementaciju metode PotenciranjeRazlike.

```
static void Main(string[] args)
{
    Console.Write("Unesite 1. broj: ");
    double a = Convert.ToDouble(Console.ReadLine());

    Console.Write("Unesite 2. broj: ");
    double b = Convert.ToDouble(Console.ReadLine());

    Console.Write("Unesite 3. broj: ");
    double c = Convert.ToDouble(Console.ReadLine());

    Console.WriteLine("Rezultat potenciranja razlike je " +
        PotenciranjeRazlike(a, b, c) + ".");

    Console.ReadKey();
}

static double PotenciranjeRazlike(double a, double b, double c)
{
    return Math.Pow((a - b), c);
}
```



Zadatak 3

Napišite konzolnu aplikaciju koja učitava sadržaj datoteke "data.txt" u kojoj su upisani realni brojevi svaki u zasebnom retku.

Program treba na kraju ispisati sumu svih učitanih realnih brojeva.

Napomena: nije potrebno u programu dodati using System.IO.

```
static void Main(string[] args)
{
    StreamReader reader = new StreamReader("data.txt");

    double sum = 0.0;

    while (!reader.EndOfStream)
    {
        sum += Convert.ToDouble(reader.ReadLine());
    }

    reader.Close();

    Console.WriteLine("Suma učitanih brojeva je " + sum + ".");

    Console.ReadKey();
}
```



Zadatak 4

Napišite konzolnu aplikaciju koja učitava 1000 cjelobrojnih vrijednosti u jednodimenzionalno polje te ispisuje one indekse polja čija je vrijednost polja jednaka 0.

```
static void Main(string[] args)
{
    int[] polje = new int[1000];

    for (int i = 0; i < 1000; i++)
    {
        Console.Write("Unesite " + (i + 1) + ". broj: ");
        polje[i] = Convert.ToInt32(Console.ReadLine());
    }

    Console.WriteLine("Indeksi polja čija je vrijednost polja jednaka 0 su:");
    for (int i = 0; i < polje.Length; i++)
    {
        if (polje[i] == 0)
        {
            Console.WriteLine(i);
        }
    }

    Console.ReadKey();
}
```



Zadatak 5

Nadopunite sljedeći kod unutar Main metode na način da stvorite novi grad "Zagreb" s koordinatama x=15.9773, y=45.8129 i brojem stanovnika 800 000.

Potom pozovite metodu koja će ispisati informacije vezane za stvoreni grad.

```
static void Main(string[] args)
{
    //Nadopuni

    Grad g = new Grad(15.9773, 45.8129, "Zagreb", 800000);
    g.ispis2();

    Console.ReadKey();
}

public class Grad
{
    private string naziv;
    private double x;
    private double y;
    private int brojStanovnika;

    public Grad(double X, double Y, string NAZIV, int BROJSTANOVNIKA)
    {
        naziv = NAZIV;
        x = X;
        y = Y;
        brojStanovnika = BROJSTANOVNIKA;
    }

    private void ispis1()
    {
        Console.WriteLine("Naziv grada: " + naziv + ", x= " + x + ", y= "
            + y + ", brSt= " + brojStanovnika);
    }

    public void ispis2()
    {
        Console.WriteLine("Naziv grada: " + naziv + ", x= " + x + ", y= "
            + y + ", brSt= " + brojStanovnika);
    }
}
```



Ispit #14 (06.07.17. - grupa B)

Zadatak 1

Napišite konzolnu aplikaciju koja učitava dva realna broja x i y koji predstavljaju točku $T(x,y)$ u kartezijevom koordinatnom sustavu.

Program treba ispisati da li se točka nalazi u prvom, drugom ili niti jednom od navedenih kvadranta. Format ispisa mora biti kao u primjeru.

Primjer: ako se za x učitava vrijednost -3.2, a za y vrijednost -2 potrebno je ispisati: "Točka $T(-3.2,-2)$ se ne nalazi niti u prvom niti u drugom kvadrantu!"

Napomena: pripadnost točke nekom kvadrantu određuje se na temelju pozitivnih i negativnih vrijednosti x i y .

```
static void Main(string[] args)
{
    Console.Write("Unesite x: ");
    double x = Convert.ToDouble(Console.ReadLine());

    Console.Write("Unesite y: ");
    double y = Convert.ToDouble(Console.ReadLine());

    if (x > 0 && y > 0)
    {
        Console.WriteLine("Točka T(" + x + "," + y + ") se nalazi u 1. kvadrantu!");
    }
    else if (x < 0 && y > 0)
    {
        Console.WriteLine("Točka T(" + x + "," + y + ") se nalazi u 2. kvadrantu!");
    }
    else
    {
        Console.WriteLine("Točka T(" + x + "," + y + ") se ne nalazi niti u prvom niti u drugom kvadrantu!");
    }

    Console.ReadKey();
}
```



Zadatak 2

Napišite metodu ZbrojPotencija koja kao argument prima tri realna broja.

Metoda prvi i drugi potencira s trećim brojem, potom navedene vrijednosti zbraja te ispisuje rezultat u konzolu.

Potrebno je napisati deklaraciju i implementaciju metode ZbrojPotencija.

```
static void Main(string[] args)
{
    Console.Write("Unesite 1. broj: ");
    double a = Convert.ToDouble(Console.ReadLine());

    Console.Write("Unesite 2. broj: ");
    double b = Convert.ToDouble(Console.ReadLine());

    Console.Write("Unesite 3. broj: ");
    double c = Convert.ToDouble(Console.ReadLine());

    ZbrojPotencija(a, b, c);

    Console.ReadKey();
}

static void ZbrojPotencija(double a, double b, double c)
{
    double rezultat = (Math.Pow(a, c) + Math.Pow(b, c));

    Console.WriteLine("Zbroj potencija je " + rezultat);
}
```



Zadatak 3

Napišite konzolnu aplikaciju koja redom učitava 10 cjelobrojnih vrijednosti s konzole i odmah iz zapisuje u datoteku "podaci.txt".

```
static void Main(string[] args)
{
    StreamWriter writer = new StreamWriter("podaci.txt");

    for (int i = 1; i <= 10; i++)
    {
        Console.Write("Unesite " + i + ". broj: ");
        writer.WriteLine(Convert.ToInt32(Console.ReadLine()));
    }

    writer.Flush();
    writer.Close();

    Console.ReadKey();
}
```



Zadatak 4

Napišite konzolnu aplikaciju koja učitava 1000 realnih vrijednosti u jednodimenzionalno polje, te ispisuje sve elemente polja čiji su indeksi parni brojevi.

```
static void Main(string[] args)
{
    double[] polje = new double[1000];

    for (int i = 0; i < polje.Length; i++)
    {
        Console.Write("Unesite " + (i + 1) + ". broj: ");
        polje[i] = Convert.ToDouble(Console.ReadLine());
    }

    Console.WriteLine("Elementi polja čiji su indeksi parni brojevi su:");
    for (int i = 0; i < polje.Length; i++)
    {
        if (i % 2 == 0)
        {
            Console.WriteLine(polje[i]);
        }
    }

    Console.ReadKey();
}
```



Zadatak 5

Nadopunite sljedeći kod unutar Main metode na način da stvorite novi kolegij naziva "ALGPRO" na zavodu "ITS" koji nosi 7 ECTS-a i ima 270 studenata.

Potom pozovite metodu kojom će se povećati broj studenata za 1.

```
static void Main(string[] args)
{
    //Nadopuni

    Kolegij k = new Kolegij("ITS", 7, "ALGPRO", 270);
    k.UvecajBrojStudenata1(1);

    Console.ReadKey();
}

public class Kolegij
{
    private string naziv;
    private string zavod;
    private int ects;
    private int brojStudenata;

    public Kolegij(string ZAVOD, int ECTS, string NAZIV, int
    BROJSTUDENATA)
    {
        naziv = NAZIV;
        zavod = ZAVOD;
        ects = ECTS;
        brojStudenata = BROJSTUDENATA;
    }

    public void UvecajBrojStudenata1(int brStud)
    {
        brojStudenata += brStud;
    }

    private void UvecajBrojStudenata2()
    {
        brojStudenata++;
    }
}
```



Ispit #15 (13.07.17. - grupa A)

Zadatak 1

Napišite konzolnu aplikaciju koja učitava dva realna broja r_1 i r_2 s tipkovnice koji predstavljaju polumjere odgovarajućih krugova s površinama P_1 i P_2 .

Ako je r_1 veće ili jednako r_2 potrebno je izračunati površinu kružnog vijenca $P = P_1 - P_2$, inače je potrebno izračunati površinu kružnog vijenca $P = P_2 - P_1$.

Format ispisa mora biti kao što je navedeno u primjeru.

Npr. ako je $r_1 = 3$, a $r_2 = 2$ potrebno je ispisati:

"Površina kružnog vijenca koju čine krugovi polumjera $r_1 = 3$ i $r_2 = 2$ je $P = 15.708$ ".

```
static void Main(string[] args)
{
    Console.Write("Unesite polumjer prvog kruga r1: ");
    double r1 = Convert.ToDouble(Console.ReadLine());

    Console.Write("Unesite polumjer prvog kruga r2: ");
    double r2 = Convert.ToDouble(Console.ReadLine());

    double P1 = Math.Pow(r1, 2) * Math.PI;
    double P2 = Math.Pow(r2, 2) * Math.PI;
    double P = 0.0;

    if (r1 >= r2)
    {
        P = P1 - P2;
    }
    else
    {
        P = P2 - P1;
    }

    Console.WriteLine("Površina kružnog vijenca koju čine krugovi  
polumjera r1 = " + r1 + " i r2 = " + r2 + " je P = " + P + ".");

    Console.ReadKey();
}
```



Zadatak 2

Napišite metodu Umnožak koja kao argument prima prirodan broj te ispisuje umnožak svih prirodnih brojeva do uključujući tog broja.

Potrebno je napisati deklaraciju i implementaciju metode Umnožak.

```
static void Main(string[] args)
{
    Console.Write("Unesite n: ");
    int n = Convert.ToInt32(Console.ReadLine());

    Umnozak(n);

    Console.ReadKey();
}

static void Umnozak(int n)
{
    int umnozak = 1;

    for (int i = 1; i <= n; i++)
    {
        umnozak *= i;
    }

    Console.WriteLine("Umnožak svih prirodnih brojeva do broja " + n + " je "
        + umnozak + ".");
}
```



Zadatak 3

Napišite konzolnu aplikaciju koja redom učitava 10 realnih vrijednosti s konzole i odmah ih zapisuje u datoteku "data.txt".

Napomena: nije potrebno u program dodati Using.System.IO niti try-catch blok.

```
using System.IO;

static void Main(string[] args)
{
    StreamWriter writer = new StreamWriter("data.txt");

    for (int i = 1; i <= 10; i++)
    {
        Console.Write("Unesite " + i + ". broj: ");
        writer.WriteLine(Convert.ToDouble(Console.ReadLine()));
    }

    writer.Flush();
    writer.Close();

    Console.ReadKey();
}
```



Zadatak 4

Napišite konzolnu aplikaciju koja učitava 1000 cijelih vrijednosti u jednodimenzionalno polje, te ispisuje sumu onih elemenata koji su djeljivi sa 3.

```
static void Main(string[] args)
{
    int[] polje = new int[1000];

    int suma = 0;

    for (int i = 0; i < polje.Length; i++)
    {
        Console.Write("Unesite " + (i + 1) + ". broj: ");
        polje[i] = Convert.ToInt32(Console.ReadLine());

        if (polje[i] % 3 == 0)
        {
            suma += polje[i];
        }
    }

    Console.WriteLine("Suma svih elemenata dijeljivih sa tri je " + suma + ".");

    Console.ReadKey();
}
```



Zadatak 5

Nadopunite sljedeći kod unutar Main metode na način da stvorite novi grad "Zagreb" s koordinatama x=15.9773, y=45.8129 i brojem stanovnika 800 000.

Potom pozovite metodu koja će ispisati informacije vezane za stvoreni grad.

```
static void Main(string[] args)
{
    //Nadopuni

    Grad g = new Grad(15.9773, 45.8129, "Zagreb", 800000);
    g.ispis2();

    Console.ReadKey();
}

public class Grad
{
    private string naziv;
    private double x;
    private double y;
    private int brojStanovnika;

    public Grad(double X, double Y, string NAZIV, int BROJSTANOVNIKA)
    {
        naziv = NAZIV;
        x = X;
        y = Y;
        brojStanovnika = BROJSTANOVNIKA;
    }

    private void ispis1()
    {
        Console.WriteLine("Naziv grada: " + naziv + ", x= " + x + ", y= "
            + y + ", brSt= " + brojStanovnika);
    }

    public void ispis2()
    {
        Console.WriteLine("Naziv grada: " + naziv + ", x= " + x + ", y= "
            + y + ", brSt= " + brojStanovnika);
    }
}
```



Ispit #16 (13.07.17. - grupa B)

Zadatak 1

Napišite konzolnu aplikaciju koja učitava realan broj r koji predstavlja polumjer kruga.

Ako je polumjer kruga manji ili jednak nuli aplikacija ispisuje "Niste unijeli dobar polumjer!", inače ispisuje površinu i opseg kruga kao u primjeru.

Npr. ako je za r učitani broj 2.5 ispis mora biti sljedeći: "Površina kruga polumjera 2.5 je $P = 19.625$, a opseg je $O = 15.7$!".

```
static void Main(string[] args)
{
    Console.Write("Unesite polumjer kruga r: ");
    double r = Convert.ToDouble(Console.ReadLine());

    if (r <= 0)
    {
        Console.WriteLine("Niste unijeli dobar polumjer!");
    }
    else
    {
        double P = Math.Pow(r, 2) * Math.PI;
        double O = 2 * r * Math.PI;

        Console.WriteLine("Površina kruga polumjera " + r + " je " + "P = " + P + ", a opseg je O = " + O + "!");
    }

    Console.ReadKey();
}
```



Zadatak 2

Napišite metodu Suma koja kao argument prima dva prirodna broja te ispisuje sumu svih brojeva između dva predana broja (granice su uključujuće).

Potrebno je napisati deklaraciju i implementaciju metode Suma.

```
static void Main(string[] args)
{
    Console.Write("Unesite 1. broj: ");
    int x = Convert.ToInt32(Console.ReadLine());

    Console.Write("Unesite 2. broj: ");
    int y = Convert.ToInt32(Console.ReadLine());

    Suma(x, y);

    Console.ReadKey();
}

static void Suma(int x, int y)
{
    int suma = 0;

    for (int i = x; i <= y; i++)
    {
        suma += i;
    }

    Console.WriteLine("Suma svih brojeva između " + x + " i " + y + " je "
        + suma + ".");
}
```



Zadatak 3

Napišite konzolnu aplikaciju koja učitava sadržaj datoteke "data.txt" u kojoj su upisani cijeli brojevi svaki u zasebnom retku.

Program treba na kraju ispisati sumu svih učitanih brojeva.

Napomena: nije potrebno u program dodati Using.System.IO niti try-catch blok.

```
using System.IO;

static void Main(string[] args)
{
    StreamReader reader = new StreamReader("data.txt");

    int suma = 0;

    while (!reader.EndOfStream)
    {
        suma += Convert.ToInt32(reader.ReadLine());
    }

    Console.WriteLine("Suma učitanih brojeva je " + suma + ".");

    reader.Close();

    Console.ReadKey();
}
```



Zadatak 4

Napišite konzolnu aplikaciju koja učitava 1000 realnih vrijednosti u jednodimenzionalno polje, te ispisuje koliko je elemenata polja negativno (prebrojava broj negativnih vrijednosti).

```
static void Main(string[] args)
{
    double[] polje = new double[1000];

    int brojNegativnih = 0;

    for (int i = 0; i < polje.Length; i++)
    {
        Console.Write("Unesite " + (i + 1) + ". broj: ");
        polje[i] = Convert.ToDouble(Console.ReadLine());

        if (polje[i] < 0)
        {
            brojNegativnih++;
        }
    }

    Console.WriteLine("Broj negativnih elemenata polja je " + brojNegativnih + ".");

    Console.ReadKey();
}
```



Zadatak 5

Nadopunite sljedeći kod unutar klase Grad na način da stvorite metodu PovecajBrojStanovnika koja će povećati broj stanovnika grada za 100 000.

Potom je instanci klase Grad stvorenoj u Main metodi potrebno povećati broj stanovnika pomoću metode PovecajBrojStanovnika.

```
static void Main(string[] args)
{
    Grad g = new Grad(15.9773, 45.8129, "Zagreb", 800000);

    //Nadopuni

    g.PovecajBrojStanovnika();

    Console.ReadKey();
}

public class Grad
{
    private string naziv;
    private double x;
    private double y;
    private int brojStanovnika;

    public Grad(double X, double Y, string NAZIV, int BROJSTANOVNIKA)
    {
        naziv = NAZIV;
        x = X;
        y = Y;
        brojStanovnika = BROJSTANOVNIKA;
    }

    private void ispis1()
    {
        Console.WriteLine("Naziv grada: " + naziv + ", x= " + x + ", y= "
            + y + ", brSt= " + brojStanovnika);
    }

    public void ispis2()
    {
        Console.WriteLine("Naziv grada: " + naziv + ", x= " + x + ", y= "
            + y + ", brSt= " + brojStanovnika);
    }

    //Nadopuni

    public void PovecajBrojStanovnika()
    {
        brojStanovnika += 100000;
    }
}
```



Ispit #17 (21.06.18. – grupa A)

Zadatak 1

Napišite program koji učitava četiri realne vrijednosti s tipkovnice, redom x_1 , y_1 , x_2 i y_2 koje predstavljaju dvije točke u koordinatnom sustavu $T_1 (x_1, y_1)$ i $T_2 (x_2, y_2)$, odnosno dužinu $\overline{T_1 T_2}$. Ako su sve unesene vrijednosti unutar intervala $[-10, 10]$ potrebno je izračunati i ispisati točku polovišta dužine u formatu ispisa kao u primjeru, inače je potrebno ispisati „Nisu unesene ispravne vrijednosti!“.

Provjeru da li je pojedina vrijednost unutar intervala $[-10, 10]$ potrebno je napisati u zasebnoj metodi Provjera koja prima samo jedan argument te vraća rezultat provjere.

Polovište se određuje kao $P(x, y) = P\left(\frac{x_1+x_2}{2}, \frac{y_1+y_2}{2}\right)$.

Primjerice, ako se redom učitaju vrijednosti -1.5, 3.0, 4.0, -2.0 potrebno je ispisati: „Polovište učitane dužine je: P (1.25, 0.5)“.

```
static void Main(string[] args)
{
    Console.WriteLine("Koordinate točke T1!");

    Console.Write("Unesite x1: ");
    double x1 = Convert.ToDouble(Console.ReadLine());

    Console.Write("Unesite y1: ");
    double y1 = Convert.ToDouble(Console.ReadLine());

    Console.WriteLine("Koordinate točke T2!");

    Console.Write("Unesite x2: ");
    double x2 = Convert.ToDouble(Console.ReadLine());

    Console.Write("Unesite y2: ");
    double y2 = Convert.ToDouble(Console.ReadLine());
}
```



```

    bool validanUnos = true;

    if (!Provjera(x1))
    {
        validanUnos = false;
    }
    else if (!Provjera(y1))
    {
        validanUnos = false;
    }
    else if (!Provjera(x2))
    {
        validanUnos = false;
    }
    else if (!Provjera(y2))
    {
        validanUnos = false;
    }

    if (validanUnos)
    {
        double px = (x1 + x2) / 2;
        double py = (y1 + y2) / 2;

        Console.WriteLine("Polovište učitavanje dužine je: P(" + px + ", " + py
            + ")");
    }
    else
    {
        Console.WriteLine("Nisu unesene ispravne vrijednosti!");
    }

    Console.ReadKey();
}

static bool Provjera(double n)
{
    if (n >= -10 && n <= 10)
    {
        return true;
    }
    else
    {
        return false;
    }
}
}

```



Zadatak 2

Napišite konzolnu aplikaciju koja učitava sadržaj datoteke bodovi.txt u kojoj su upisani realni brojevi koji predstavljaju bodove studenata na ispitu, svaki u zasebnom retku.

Program treba na kraju ispisati koliko je studenata pristupilo ispitu, prosječan, minimalan i maksimalan broj bodova.

Napomena: potrebno je koristiti try-catch blok, te nije potrebno dodati imenski prostor System.IO.

using System.IO;

```
static void Main(string[] args)
{
    try
    {
        StreamReader reader = new StreamReader("bodovi.txt");

        int brojStudenata = 0;
        double zbrojBodova = 0.0;
        double minimalanBrojBodova = 0.0;
        double maksimalanBrojBodova = 0.0;

        while (!reader.EndOfStream)
        {
            double brojBodova = Convert.ToDouble(reader.ReadLine());

            brojStudenata++;
            zbrojBodova += brojBodova;

            if (brojStudenata == 1)
            {
                minimalanBrojBodova = brojBodova;
                maksimalanBrojBodova = brojBodova;
            }
            else if (brojBodova < minimalanBrojBodova)
            {
                minimalanBrojBodova = brojBodova;
            }
            else if (brojBodova > maksimalanBrojBodova)
            {
                maksimalanBrojBodova = brojBodova;
            }
        }

        reader.Close();

        double prosjekBodova = zbrojBodova / brojStudenata;

        Console.WriteLine("Broj studenata koji je pristupio ispitu: " +
            brojStudenata);
        Console.WriteLine("Prosječan broj bodova na ispitu: " +
            prosjekBodova);
        Console.WriteLine("Minimalan broj bodova na ispitu: " +
            minimalanBrojBodova);
        Console.WriteLine("Maksimalan broj bodova na ispitu: " +
            maksimalanBrojBodova);
    }
}
```



```
        catch
        {
            Console.WriteLine("Datoteka bodovi.txt ne postoji!");
        }

        Console.ReadKey();
    }
}
```



Zadatak 3

Napišite konzolnu aplikaciju koja generira 100 slučajnih cijelih brojeva u intervalu [5,99] te sprema brojeve ovisno o njihovoj parnosti u dvije liste: parnu i neparnu.

U svakoj iteraciji provjerava se parnost dohvaćenog slučajnog broja, te ako je broj paran dodaje ga se u listu parnih, a ako je broj neparan u listu neparnih brojeva.

```
static void Main(string[] args)
{
    Random rnd = new Random();

    List<int> parni = new List<int>();
    List<int> neparni = new List<int>();

    for (int i = 1; i <= 100; i++)
    {
        int x = rnd.Next(5, 100);

        if (x % 2 == 0)
        {
            parni.Add(x);
        }
        else
        {
            neparni.Add(x);
        }
    }

    Console.ReadKey();
}
```



Zadatak 4

Napišite program koji učitava dvije cjelobrojne vrijednosti n i m koje predstavljaju broj redaka i stupaca matrice $M = n \times m$. Potom je potrebno kreirati matricu M i popuniti ju slučajnim cijelim brojevima unutar intervala $[1, 10>$. Na kraju je sadržaj matrice potrebno ispisati u konzolu u proizvoljnom formatu.

```
static void Main(string[] args)
{
    Console.WriteLine("Unesite n: ");
    int n = Convert.ToInt32(Console.ReadLine());

    Console.WriteLine("Unesite m: ");
    int m = Convert.ToInt32(Console.ReadLine());

    int[,] M = new int[n, m];

    Random rnd = new Random();

    for (int i = 0; i < M.GetLength(0); i++)
    {
        for (int j = 0; j < M.GetLength(1); j++)
        {
            M[i, j] = rnd.Next(1, 10);
            Console.Write(M[i, j] + "\t");
        }
        Console.WriteLine();
    }

    Console.ReadKey();
}
```



Zadatak 5

Napišite kod klase Student koja sadrži javne atribute Ime, Prezime i JMBAG, te privatni atribut Ocjena. Konstruktor klase prima argumente imena, prezimena i jmbag, a ocjenu inicijalizira na vrijednost 1.

Potrebno je također napisati i javnu metodu postaviOcjenu koja kao argument prima ocjenu te postavlja studentu ocjenu predanu preko argumenta.

```
class Student
{
    public string Ime;
    public string Prezime;
    public string JMBAG;
    private int Ocjena;

    public Student(string ime, string prezime, string jmbag)
    {
        Ime = ime;
        Prezime = prezime;
        JMBAG = jmbag;
        Ocjena = 1;
    }

    public void postaviOcjenu(int ocjena)
    {
        Ocjena = ocjena;
    }
}
```



Ispit #18 (21.06.18. – grupa B)

Zadatak 1

Napišite program koji učitava četiri cijela broja s tipkovnice, redom x_1 , y_1 , x_2 i y_2 koji predstavljaju dvije točke u koordinatnom sustavu $T_1 (x_1, y_1)$ i $T_2 (x_2, y_2)$, odnosno dužinu $\overline{T_1 T_2}$. Ako su sve unesene vrijednosti unutar intervala $[-20, 20]$ potrebno je izračunati i ispisati udaljenost između dvije točke u formatu ispisa kao u primjeru, inače je potrebno ispisati „Nisu unesene ispravne vrijednosti!“.

Provjeru da li je pojedina vrijednost unutar intervala $[-20, 20]$ potrebno je napisati u zasebnoj metodi Provjera koja prima samo jedan argument te vraća rezultat provjere.

Udaljenost se određuje kao $D(T_1, T_2) = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$.

Primjerice, ako se redom učitaju vrijednosti -1, 3, 4, -2 potrebno je ispisati: „Dužina iznosi: 7.07“.

```
static void Main(string[] args)
{
    Console.WriteLine("Koordinate točke T1!");

    Console.Write("Unesite x1: ");
    int x1 = Convert.ToInt32(Console.ReadLine());

    Console.Write("Unesite y1: ");
    int y1 = Convert.ToInt32(Console.ReadLine());

    Console.WriteLine("Koordinate točke T2!");

    Console.Write("Unesite x2: ");
    int x2 = Convert.ToInt32(Console.ReadLine());

    Console.Write("Unesite y2: ");
    int y2 = Convert.ToInt32(Console.ReadLine());
}
```



```

    bool validanUnos = true;

    if (!Provjera(x1))
    {
        validanUnos = false;
    }
    else if (!Provjera(y1))
    {
        validanUnos = false;
    }
    else if (!Provjera(x2))
    {
        validanUnos = false;
    }
    else if (!Provjera(y2))
    {
        validanUnos = false;
    }

    if (validanUnos)
    {
        double udaljenost = Math.Sqrt(Math.Pow(x2 - x1, 2) +
            Math.Pow(y2 - y1, 2));

        Console.WriteLine("Dužina iznosi: " + udaljenost);
    }
    else
    {
        Console.WriteLine("Nisu unesene ispravne vrijednosti!");
    }

    Console.ReadKey();
}

static bool Provjera(int n)
{
    if (n >= -20 && n <= 20)
    {
        return true;
    }
    else
    {
        return false;
    }
}

```



Zadatak 2

Napišite konzolnu aplikaciju koja učitava cijele brojeve iz konzole sve dok se ne unese vrijednost 0. Uneseni brojevi spremaju se u datoteku brojevi.txt, svaki u zasebnom retku.

Na kraju je u datoteku brojevi.txt potrebno upisati i prosjek učitanih brojeva (računajući i zadnje unesenu 0).

Napomena: potrebno je koristiti try-catch blok.

```
using System.IO;

static void Main(string[] args)
{
    try
    {
        int unos = 0;
        int zbroj = 0;
        int brojUnosa = 0;

        StreamWriter writer = new StreamWriter("brojevi.txt");

        do
        {
            Console.Write("Unesite cijeli broj: ");
            unos = Convert.ToInt32(Console.ReadLine());

            zbroj += unos;
            brojUnosa++;

            writer.WriteLine(unos);
        } while (unos != 0);

        double prosjek = zbroj / Convert.ToDouble(brojUnosa);

        writer.WriteLine("Prosjek unesenih brojeva je: " + prosjek);

        writer.Flush();
        writer.Close();
    }
    catch (Exception ex)
    {
        Console.WriteLine("Greška: " + ex);
    }

    Console.ReadKey();
}
```



Zadatak 3

Napišite konzolnu aplikaciju koja 100 slučajnih cijelih brojeva unutar intervala [1, 100] sprema u dvije liste ovisno o tome da li su brojevi djeljivi s 5 ili ne.

U svakoj iteraciji provjerava se da li je broj djeljiv s 5 ili ne, ako je dodaje ga se u listu djeljivih s 5, inače ga se dodaje u listu nedjeljivih s 5.

```
static void Main(string[] args)
{
    Random rnd = new Random();

    List<int> djeljivi = new List<int>();
    List<int> nedjeljivi = new List<int>();

    for (int i = 1; i <= 100; i++)
    {
        int x = rnd.Next(1, 101);

        if (x % 5 == 0)
        {
            djeljivi.Add(x);
        }
        else
        {
            nedjeljivi.Add(x);
        }
    }

    Console.ReadKey();
}
```



Zadatak 4

Napišite program koji učitava 9 cjelobrojnih vrijednosti u kvadratnu matricu (3 x 3). Pritom je potrebno koristiti petlje po izboru. Nakon što su učitani svi elementi matrice potrebno je ispisati elemente glavne dijagonale matrice. Format ispisa mora biti kao u primjeru. Primjerice, ako su redom učitane vrijednosti po redcima:

1, 2, 3

4, 5, 6

7, 8, 9

Potrebno je ispisati: „Elementi glavne dijagonale matrice su: 1, 5, 9“

```
static void Main(string[] args)
{
    int[,] matrica = new int[3, 3];

    for (int i = 0; i < matrica.GetLength(0); i++)
    {
        for (int j = 0; j < matrica.GetLength(1); j++)
        {
            Console.Write("Unesite broj: ");
            matrica[i, j] = Convert.ToInt32(Console.ReadLine());
        }
    }

    Console.WriteLine("Elementi glavne dijagonale matrice su: " +
        matrica[0, 0] + ", " + matrica[1, 1] + ", " + matrica[2, 2]);

    Console.ReadKey();
}
```



Zadatak 5

Napišite kod klase Student koja sadrži javne atribute Ime, Prezime i JMBAG, te privatni atribut Ocjena. Konstruktor klase prima argumente imena, prezimena i jmbag, a ocjenu inicijalizira na vrijednost -1.

Potrebno je također napisati i javnu metodu Ispis koja ispisuje ime, prezime i ocjenu studenta u formatu: „Student Ime Prezime iz kolegija Algoritmi i programiranje je ostvario ocjenu ocjena!“.

```
class Student
{
    public string Ime;
    public string Prezime;
    public string JMBAG;
    private int Ocjena;

    public Student(string ime, string prezime, string jmbag)
    {
        Ime = ime;
        Prezime = prezime;
        JMBAG = jmbag;
        Ocjena = -1;
    }

    public void Ispis()
    {
        Console.WriteLine("Student " + Ime + " " + Prezime + " iz kolegija  
Algoritmi i programiranje je ostvario ocjenu " + Ocjena + "!");
    }
}
```



Ispit #19 (05.07.18. – grupa A)

Zadatak 1

Nadopunite sljedeći kod metodama UvecajKilometre i PromijeniVlasnika tako da njihovi parametri odgovaraju pozivu iz main metode.

Metoda UvecajKilometre treba uvećati kilometre za zadani broj koji se predaje preko argumenta, dok metoda PromijeniVlasnika treba promijeniti ime vlasnika u ime koje se predaje preko argumenta.

```
static void Main(string[] args)
{
    Auto a = new Auto("Insignia", 50000, "Marko");
    a.UvecajKilometre(10000);
    a.PromijeniVlasnika("Ivo");

    Console.ReadKey();
}

public class Auto
{
    private string imeAuta;
    private int kilometri;
    private string imeVlasnika;

    public Auto(string IMEAUTA, int KILOMETRI, string IMEVLASNIKA)
    {
        imeAuta = IMEAUTA;
        kilometri = KILOMETRI;
        imeVlasnika = IMEVLASNIKA;
    }

    //Nadopuni

    public void UvecajKilometre(int KILOMETRI)
    {
        kilometri += KILOMETRI;
    }

    public void PromijeniVlasnika(string IMEVLASNIKA)
    {
        imeVlasnika = IMEVLASNIKA;
    }
}
```



Zadatak 2

Napišite konzolnu aplikaciju koja učitava sadržaj datoteke paketi.txt u kojoj su upisani realni brojevi koji predstavljaju volumen paketa (m^3) koje je potrebno dostaviti vozilima koja imaju volumni kapacitet 10 m^3 .

Program u proizvoljnom formatu treba ispisati ukupni volumen koji je potrebno prevesti i broj vozila koji je potrebno koristiti.

Napomena: za zaokruživanje na prvi veći cijeli broj koristiti funkciju `Math.Ceiling(double value)` unutar ugrađene matematičke klase.

Potrebno je koristiti try-catch blok, te nije potrebno dodati imenski prostor `System.IO`.

```
using System.IO;

static void Main(string[] args)
{
    try
    {
        StreamReader reader = new StreamReader("paketi.txt");

        double sumaVolumena = 0.0;

        while (!reader.EndOfStream)
        {
            double volumen = Convert.ToDouble(reader.ReadLine());
            sumaVolumena += volumen;
        }

        reader.Close();

        int brojVozila = Convert.ToInt32(Math.Ceiling(sumaVolumena / 10.0));

        Console.WriteLine("Ukupni volumen kojeg je potrebno dostaviti je " +
            sumaVolumena + " [m^3].");
        Console.WriteLine("Za dostavu je potrebno " + brojVozila
            + " vozila.");
    }
    catch (Exception ex)
    {
        Console.WriteLine(ex);
    }

    Console.ReadKey();
}
```



Zadatak 3

Napišite konzolnu aplikaciju koja generira 1000 slučajnih cijelih brojeva koji predstavljaju brzine vozila unutar intervala [20, 100] km/h na određenom prosjeku prometnice.

Ograničenje na prometnici je 80 km/h.

Potrebno je u zasebnu listu spremiti brzine vozila čija je vrijednost iznad ograničenja i u proizvoljnom formatu ispisati koliko ima takvih zabilježenih brzina (većih od ograničenja).

```
static void Main(string[] args)
{
    Random rnd = new Random();

    List<int> brzineIznadOgranicenja = new List<int>();

    for (int i = 1; i <= 1000; i++)
    {
        int brzina = rnd.Next(20, 101);

        if (brzina > 80)
        {
            brzineIznadOgranicenja.Add(brzina);
        }
    }

    Console.WriteLine("Ukupan broj vozila koja su se kretala iznad ograničenja  
je " + brzineIznadOgranicenja.Count + ".");

    Console.ReadKey();
}
```



Zadatak 4

Napišite konzolnu aplikaciju koja učitava JMBAG (niz znakova) studenata i broj bodova na ispitu (realan broj) u jednodimenzionalna polja (dva polja). Ispitni rok prijavilo je 75 studenata.

Nakon učitavanja potrebno je u proizvoljnom formatu ispisati JMBAG i broj bodova onih studenata koji imaju više od 12 bodova.

```
static void Main(string[] args)
{
    string[] JMBAG = new string[75];
    double[] bodovi = new double[75];

    for (int i = 0; i < 75; i++)
    {
        Console.Write("Unesite JMBAG " + (i + 1) + ". studenta: ");
        JMBAG[i] = Console.ReadLine();

        Console.Write("Unesite bodove " + (i + 1) + ". studenta: ");
        bodovi[i] = Convert.ToDouble(Console.ReadLine());
    }

    Console.WriteLine("Studenti koji imaju više od 12 bodova su:");
    for (int i = 0; i < 75; i++)
    {
        if (bodovi[i] > 12)
        {
            Console.WriteLine("JMBAG studenta: " + JMBAG[i]
                + ", broj bodova: " + bodovi[i]);
        }
    }

    Console.ReadKey();
}
```



Zadatak 5

Napišite konzolnu aplikaciju koja učitava četiri realna broja x_1 , y_1 , x_2 , y_2 koji predstavljaju geografske koordinate u stupnjevima.

Potrebno je napisati metodu udaljenost koja prima učitane geografske koordinate te vraća udaljenost u kilometrima između njih.

U glavnom programu potrebno je ispisati poruku u formatu kao u primjeru. Za izračun udaljenosti koristiti euklidsku udaljenost, te koristiti $1^\circ = 111.32$ km.

Primjer: ako se redom učitaju vrijednosti: 15.97, 45.8, 18.68, 45.55 potrebno je ispisati: „Udaljenost između točaka (15.97, 45.8) i (18.68, 45.55) iznosi 302.95 km!“.

Izračun euklidske udaljenosti između dvije točke $T_1(x_1, y_1)$ i $T_2(x_2, y_2)$ računa se prema izrazu:

$$d(T_1, T_2) = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

```
static void Main(string[] args)
{
    Console.WriteLine("Koordinate točke T1!");

    Console.Write("Unesite x1: ");
    double x1 = Convert.ToDouble(Console.ReadLine());

    Console.Write("Unesite y1: ");
    double y1 = Convert.ToDouble(Console.ReadLine());

    Console.WriteLine("Koordinate točke T2!");

    Console.Write("Unesite x2: ");
    double x2 = Convert.ToDouble(Console.ReadLine());

    Console.Write("Unesite y2: ");
    double y2 = Convert.ToDouble(Console.ReadLine());

    Console.WriteLine("Udaljenost između točaka(" + x1 + ", " + y1 + ") i ("
        + x2 + ", " + y2 + ") iznosi " + udaljenost(x1, y1, x2, y2) + " km!");

    Console.ReadKey();
}

static double udaljenost(double x1, double y1, double x2, double y2)
{
    return Math.Sqrt(Math.Pow(x2 - x1, 2) + Math.Pow(y2 - y1, 2)) * 111.32;
}
```



Ispit #20 (05.07.18. – grupa B)

Zadatak 1

Nadopunite sljedeći kod unutar main metode na način da stvorite novi grad Zagreb s koordinatama $x = 15.9773$, $y = 45.8129$ i brojem stanovnika 800000.

Potom pozovite metodu koja će ispisati informacije vezane za stvoreni grad.

```
static void Main(string[] args)
{
    //Nadopuni

    Grad g = new Grad(15.9773, 45.8129, "Zagreb", 800000);
    g.ispis2();

    Console.ReadKey();
}

public class Grad
{
    private string naziv;
    private double x;
    private double y;
    private int brojStanovnika;

    public Grad(double X, double Y, string NAZIV, int BROJSTANOVNIKA)
    {
        naziv = NAZIV;
        x = X;
        y = Y;
        brojStanovnika = BROJSTANOVNIKA;
    }

    private void ispis1()
    {
        Console.WriteLine("Naziv grada: " + naziv + ", x = " + x + ", y = "
            + y + ", brSt=" + brojStanovnika);
    }

    public void ispis2()
    {
        Console.WriteLine("Naziv grada: " + naziv + ", x = " + x + ", y = "
            + y + ", brSt=" + brojStanovnika);
    }
}
```



Zadatak 2

Napišite konzolnu aplikaciju koja učitava sadržaj datoteke vozila.txt u kojoj su upisani realni brojevi koji predstavljaju udaljenost koje su vozila prošla u određenom danu. Jedan redak odgovara prijeđenoj udaljenosti jednog vozila.

Potrebno je odrediti maksimalnu i prosječnu prijeđenu udaljenost tog dana. Rezultat je potrebno ispisati u proizvoljnom formatu.

Potrebno je koristiti try-catch blok, te nije potrebno dodati imenski prostor System.IO.

```
using System.IO;

static void Main(string[] args)
{
    try
    {
        StreamReader reader = new StreamReader("vozila.txt");

        double sumaUdaljenosti = 0.0;
        int brojVozila = 0;
        double maksimalnaUdaljenost = double.MinValue;

        while (!reader.EndOfStream)
        {
            double udaljenost = Convert.ToDouble(reader.ReadLine());

            sumaUdaljenosti += udaljenost;
            brojVozila++;

            if (udaljenost > maksimalnaUdaljenost)
            {
                maksimalnaUdaljenost = udaljenost;
            }
        }

        reader.Close();

        double prosjecnaUdaljenost = sumaUdaljenosti / brojVozila;

        Console.WriteLine("Maksimalna prijeđena udaljenost je "
            + maksimalnaUdaljenost + " km.");
        Console.WriteLine("Prosječna prijeđena udaljenosti je "
            + prosjecnaUdaljenost + " km.");
    }
    catch (Exception ex)
    {
        Console.WriteLine(ex);
    }

    Console.ReadKey();
}
```



Zadatak 3

Nadopunite kod na način da se iz zadane liste vozni park koji predstavljaju tipove vozila u voznom parku, slučajno odabere 5 tipova vozila za današnju dostavu te ih se doda u listu tipovi.

Vrijednosti se mogu ponavljati.

```
static void Main(string[] args)
{
    List<int> vozniPark = new List<int>
        (new int[] { 1, 2, 5, 6, 7, 9, 15, 100 });

    //Nadopuni

    List<int> tipovi = new List<int>();

    Random rnd = new Random();

    for (int i = 1; i <= 5; i++)
    {
        int index = rnd.Next(0, vozniPark.Count);
        tipovi.Add(vozniPark[index]);
    }

    Console.ReadKey();
}
```



Zadatak 4

Napišite konzolnu aplikaciju koja učitava matricu udaljenosti između n gradova.

Potrebno je prvo preko tipkovnice unijeti vrijednost n , potom kreirati matricu $M = n \times n$, te ju popuniti realnim brojevima koji predstavljaju udaljenosti između gradova.

Na kraju je potrebno ispisati sadržaj matrice u proizvoljnom formatu.

```
static void Main(string[] args)
{
    Console.Write("Unesite n: ");
    int n = Convert.ToInt32(Console.ReadLine());

    double[,] M = new double[n, n];

    for (int i = 0; i < M.GetLength(0); i++)
    {
        for (int j = 0; j < M.GetLength(1); j++)
        {
            Console.Write("Unesite udaljenost između gradova određenih indeksima " + i + " - " + j + ": ");
            M[i, j] = Convert.ToDouble(Console.ReadLine());
        }
    }

    Console.WriteLine("Udaljenosti između gradova:");
    Console.Write("\t");
    for (int i = 0; i < M.GetLength(1); i++)
    {
        Console.Write(i + "\t");
    }
    Console.WriteLine();

    for (int i = 0; i < M.GetLength(0); i++)
    {
        Console.Write(i + "\t");
        for (int j = 0; j < M.GetLength(1); j++)
        {
            Console.Write(M[i, j] + "\t");
        }
        Console.WriteLine();
    }

    Console.ReadKey();
}
```



Zadatak 5

Napišite konzolnu aplikaciju koja učitava tri realna broja s , t i v_{ogr} koji označavaju redom duljinu dionice prometnice [m], vrijeme putovanja vozila po dionici [s] i ograničenje brzine na dionici prometnice [km/h].

Potrebno je napisati metodu `jednolikaBrzina` koja kao argumente prima duljinu dionice i vrijeme putovanja vozila po dionici, a vraća izračunatu jednoliku brzinu kretanja u km/h (pretvorba: $1 \text{ m/s} = 3.6 \text{ km/h}$).

U glavnom dijelu programa potrebo je provjeriti da li je izračunata brzina veća od učitane brzine ograničenja, te ako je ispisati poruku u formatu kao u primjeru.

Primjer: ako se redom učitaju vrijednosti 1000, 40, i 80 potrebno je ispisati: „Vozilo se kreće brzinom 90 km/h, odnosno iznad ograničenja brzine!“.

```
static void Main(string[] args)
{
    Console.WriteLine("Unesite duljinu dionice s [m]: ");
    double s = Convert.ToDouble(Console.ReadLine());

    Console.WriteLine("Unesite vrijeme putovanjavozila po dionici t [s]: ");
    double t = Convert.ToDouble(Console.ReadLine());

    Console.WriteLine("Unesite ograničenje brzine na dionici  
prometnice vOgr [km/h]: ");
    double vOgr = Convert.ToDouble(Console.ReadLine());

    double v = jednolikaBrzina(s, t);

    Console.WriteLine("Vozilo se kreće brzinom " + v + " km/h, odnosno ");
    if (v > vOgr)
    {
        Console.WriteLine("iznad ograničenja brzine!");
    }
    else
    {
        Console.WriteLine("unutar ograničenja brzine!");
    }

    Console.ReadKey();
}

static double jednolikaBrzina(double s, double t)
{
    return s / t * 3.6;
}
```



Ispit #21 (12.07.18. – grupa A)

Zadatak 1

Nadopunite sljedeći kod unutar main metode na način da stvorite novi grad Zagreb s koordinatama $x = 15.9773$, $y = 45.8129$ i brojem stanovnika 800000.

Potom u konzolu ispišite x koordinatu grada (vrijednost atributa unutar instance).

```
static void Main(string[] args)
{
    //Nadopuni

    Tocka t = new Tocka(15.9773, 45.8129);
    Grad g = new Grad(t, "Zagreb", 800000);

    Console.WriteLine(g.t.x);

    Console.ReadKey();
}

public class Grad
{
    private string naziv;
    public Tocka t;
    private int brojStanovnika;

    public Grad(Tocka T, string NAZIV, int BROJSTANOVNIKA)
    {
        t = T;
        naziv = NAZIV;
        brojStanovnika = BROJSTANOVNIKA;
    }
}

public class Tocka
{
    public double x;
    public double y;

    public Tocka(double X, double Y)
    {
        x = X;
        y = Y;
    }
}
```



Zadatak 2

Napišite konzolnu aplikaciju koja učitava mase paketa (kg, cijeli broj) koje je potrebno dostaviti sve dok se ne unese masa paketa manja ili jednaka nuli (zadnji paket se ne pribrojava).

Paketi se spremaju u vozila masenog kapaciteta 500 kg.

Program u proizvoljnom formatu treba ispisati ukupnu masu koju je potrebno prevesti i broj vozila koji je potrebno koristiti.

Napomena: za zaokruživanje na prvi veći cijeli broj koristiti funkciju `Math.Ceiling(double value)`.

```
static void Main(string[] args)
{
    int m = 0;
    int sumaMasa = 0;

    do
    {
        Console.Write("Unesite masu paketa: ");
        m = Convert.ToInt32(Console.ReadLine());

        if (m > 0)
        {
            sumaMasa = sumaMasa + m;
        }
    } while (m > 0);

    int brojVozila = Convert.ToInt32(Math.Ceiling(sumaMasa / 500.0));

    Console.WriteLine("Ukupna masa koju je potrebno prevesti: " + sumaMasa
        + " kg.");
    Console.WriteLine("Potreban broj vozila za prijevoz je " + brojVozila
        + ".");

    Console.ReadKey();
}
```



Zadatak 3

Napišite konzolnu aplikaciju koja generira 1000 realnih brojeva unutar intervala [0, 100> koji predstavljaju simulirane vrijednosti brzine te ih zapisuje u datoteku brzine.txt.

Nije potrebno dodati imenski prostor System.IO.

```
using System.IO;

static void Main(string[] args)
{
    StreamWriter writer = new StreamWriter("brzine.txt");
    Random rnd = new Random();

    for (int i = 1; i <= 1000; i++)
    {
        writer.WriteLine(rnd.NextDouble() * 100);
    }

    writer.Flush();
    writer.Close();

    Console.ReadKey();
}
```



Zadatak 4

Napišite konzolnu aplikaciju koja učitava sadržaj datoteke brojVozila.txt u jednodimenzionalno polje.

U datoteci je u zasebnom retku za svaki dan u mjesecu (31 redak) zapisan broj zabilježenih vozila (realan broj) u tome danu na određenom segmentu prometnice.

Potrebno je izračunati prosječan broj zabilježenih vozila te ga ispisati u konzolu u proizvoljnom formatu.

Potrebno je koristiti try-catch blok, te nije potrebno dodati imenski prostor System.IO.

```
using System.IO;

static void Main(string[] args)
{
    try
    {
        double[] vozilaPoDanima = new double[31];
        int i = 0;
        double sumaBrojaVozila = 0;

        StreamReader reader = new StreamReader("brojVozila.txt");

        while (!reader.EndOfStream)
        {
            double brojVozila = Convert.ToDouble(reader.ReadLine());
            vozilaPoDanima[i] = brojVozila;
            sumaBrojaVozila += brojVozila;
            i++;
        }

        reader.Close();

        double prosjekVozilaPoDanu = sumaBrojaVozila / 31;

        Console.WriteLine("Prosječan broj vozila u danu je "
            + prosjekVozilaPoDanu + ".");
    }
    catch (Exception ex)
    {
        Console.WriteLine(ex);
    }

    Console.ReadKey();
}
```



Zadatak 5

Napišite metodu Najveci koja prima tri realna broj a, b i c te u konzolu ispisuje najveći od tri predana broja.

Unutar glavnog dijela programa potrebno je iz konzole učitati brojeve te pozvati metodu.

Format ispisa bora biti kao u primjeru.

Primjer: ako se redom učitaju vrijednosti 10, -11.25, 11.05 potrebno je ispisati: „Najveći od tri učitana broja 10, -11.25, 11.05 je 11.05.“.

```
static void Main(string[] args)
{
    Console.Write("Unesite a: ");
    double a = Convert.ToDouble(Console.ReadLine());

    Console.Write("Unesite b: ");
    double b = Convert.ToDouble(Console.ReadLine());

    Console.Write("Unesite c: ");
    double c = Convert.ToDouble(Console.ReadLine());

    Najveci(a, b, c);

    Console.ReadKey();
}

static void Najveci(double a, double b, double c)
{
    double najveci = 0.0;

    if (a > b)
    {
        if (a > c)
        {
            najveci = a;
        }
        else
        {
            najveci = c;
        }
    }
    else
    {
        if (b > c)
        {
            najveci = b;
        }
        else
        {
            najveci = c;
        }
    }

    Console.WriteLine("Najveći od tri učitana broja " + a + ", " + b + " i "
        + c + " je " + najveci + ".");
}
```



Ispit #22 (12.07.18. – grupa B)

Zadatak 1

Nadopunite sljedeći kod unutar main metode na način da stvorite novi auto naziva Insignia, kojem je vlasnik Marko, te koji se trenutno nalazi na GPS lokaciji $x = 15.9773$, $y = 45.8129$.

Potom u konzolu ispišite vrijednost y GPS lokacije vozila (vrijednost atributa unutar instance).

```
static void Main(string[] args)
{
    //Nadopuni

    GPSLokacija l = new GPSLokacija(15.9773, 45.8129);
    Auto a = new Auto("Insignia", "Marko", l);

    Console.WriteLine(a.lokacija.y);

    Console.ReadKey();
}

public class GPSLokacija
{
    public double x;
    public double y;

    public GPSLokacija(double X, double Y)
    {
        x = X;
        y = Y;
    }
}

public class Auto
{
    private string imeAuta;
    private string imeVlasnika;
    public GPSLokacija lokacija;

    public Auto(string IMEAUTA, string IMEVLASNIKA, GPSLokacija LOKACIJA)
    {
        imeAuta = IMEAUTA;
        lokacija = LOKACIJA;
        imeVlasnika = IMEVLASNIKA;
    }
}
```



Zadatak 2

Napišite konzolnu aplikaciju koja predstavlja punjenje vozila s paketima.

Mase paketa (realni brojevi) se učitavaju iz konzole sve dok se ne prekorači nosivost vozila koja iznosi 400 kg.

Kada se prekorači nosivost potrebno je odbaciti učitani paket i ispisati poruku „Odbacujem paket zbog prekoračenja mase! “.

Na kraju je potrebno ispisati ukupnu masu koja se prevozi vozilom u proizvoljnom formatu.

```
static void Main(string[] args)
{
    double sumaMasa = 0.0;

    do
    {
        Console.Write("Unesite masu paketa: ");
        double m = Convert.ToDouble(Console.ReadLine());

        if (sumaMasa + m <= 400)
        {
            sumaMasa = sumaMasa + m;
        }
        else
        {
            Console.WriteLine("Odbacujem paket zbog prekoračenja mase!");
            break;
        }
    } while (sumaMasa < 400);

    Console.WriteLine("Masa koja se prevozi vozilom je " + sumaMasa + " kg.");
    Console.ReadKey();
}
```



Zadatak 3

Napišite konzolnu aplikaciju koja generira 900 realnih brojeva unutar intervala [0, 10> koji predstavljaju simulirane vrijednosti brzine te ih zapisuje u datoteku simuliraneBrzine.txt.

Nije potrebno dodati imenski prostor System.IO.

```
using System.IO;

static void Main(string[] args)
{
    Random rnd = new Random();

    StreamWriter writer = new StreamWriter("simuliraneBrzine.txt");

    for (int i = 1; i <= 900; i++)
    {
        double x = rnd.NextDouble() * 10;
        writer.WriteLine(x);
    }

    writer.Flush();
    writer.Close();

    Console.ReadKey();
}
```



Zadatak 4

Napišite konzolnu Aplikaciju koja učitava sadržaj datoteke vozilaUdaljenosti.txt u jednodimenzionalno polje.

U datoteci su u zasebnom retku za svaki dan u godini (365 redaka) zapisane udaljenosti (cijeli broj) koje je vozilo toga dana ostvarilo.

Potrebno je izračunati prosječnu prijeđenu udaljenost vozila kroz godinu te ju ispisati u konzolu u proizvoljnom formatu.

Potrebno je koristiti try-catch blok, te nije potrebno dodati imenski prostor System.IO.

```
using System.IO;

static void Main(string[] args)
{
    try
    {
        StreamReader reader = new StreamReader("voziloUdaljenosti.txt");

        int[] udaljenosti = new int[365];
        int sumaUdaljenosti = 0;
        int i = 0;

        while (!reader.EndOfStream)
        {
            int udaljenost = Convert.ToInt32(reader.ReadLine());
            sumaUdaljenosti = sumaUdaljenosti + udaljenost;
            udaljenosti[i] = udaljenost;
            i++;
        }

        reader.Close();

        double prosjecnaUdaljenost = sumaUdaljenosti / 365.0;

        Console.WriteLine("Prosječna udaljenost vozila kroz godinu je "
            + prosjecnaUdaljenost + ".");
    }
    catch (Exception ex)
    {
        Console.WriteLine(ex);
    }

    Console.ReadKey();
}
```



Zadatak 5

Napišite metodu Najmanji koja prima tri cijela broja a, b i c te u konzolu ispisuje najmanji od tri predana broja.

Unutar glavnog dijela programa potrebno je iz konzole učitati brojeve te pozvati metodu.

Format ispisa mora biti kao u primjeru.

Primjer: ako se redom učitaju vrijednosti 10, -11, 11 potrebno je ispisati: „Najmanji od tri učitana broja 10, -11 i 11 je -11.“.

```
static void Main(string[] args)
{
    Console.Write("Unesite a: ");
    int a = Convert.ToInt32(Console.ReadLine());

    Console.Write("Unesite b: ");
    int b = Convert.ToInt32(Console.ReadLine());

    Console.Write("Unesite c: ");
    int c = Convert.ToInt32(Console.ReadLine());

    Najmanji(a, b, c);

    Console.ReadKey();
}

private static void Najmanji(int a, int b, int c)
{
    int najmanji;

    if (a < b)
    {
        if (a < c)
        {
            najmanji = a;
        }
        else
        {
            najmanji = c;
        }
    }
    else
    {
        if (b < c)
        {
            najmanji = b;
        }
        else
        {
            najmanji = c;
        }
    }

    Console.WriteLine("Najmanji od tri učitana broja " + a + ", " + b + " i "
        + c + " je " + najmanji + ".");
}
```



Ispit #23 (06.09.18. - grupa A)

Zadatak 1

Nadopunite sljedeći kod unutar Main metode na način da stvorite novi grad Zagreb s koordinatama x = 15.9773, y = 45.8129 i brojem stanovnika 800000.

Potom pozovite metodu koja će ispisati informacije vezane na stvoreni grad.

```
static void Main(string[] args)
{
    //Nadopuni
    Grad g = new Grad(45.8129, 15.9773, "Zagreb", 800000);
    g.ispis2();

    Console.ReadKey();
}

public class Grad
{
    private string naziv;
    private double x;
    private double y;
    private int brojStanovnika;

    public Grad(double Y, double X, string NAZIV, int BROJSTANOVNIKA)
    {
        naziv = NAZIV;
        x = X;
        y = Y;
        brojStanovnika = BROJSTANOVNIKA;
    }

    private void ispis1()
    {
        Console.WriteLine("Naziv grada: " + naziv + ", x = " + x + ", y = "
            + y + ", brSt = " + brojStanovnika);
    }

    public void ispis2()
    {
        Console.WriteLine("Naziv grada: " + naziv + ", x = " + x + ", y = "
            + y + ", brSt = " + brojStanovnika);
    }
}
```



Zadatak 2

Napišite konzolnu aplikaciju koja iz datoteke poslovnice.txt učitava mjesečni ostvareni prihod određene logističke tvrtke.

Za svaku poslovnicu u zasebnom retku navedena je ukupna mjesečna zarada kao realan broj u kunama.

Potrebno je u drugu datoteku poboljsanja.txt zapisati indekse redaka onih poslovnica čija je mjesečna zarada bila manja od 50000 kn.

Smatra se da prvi redak, odnosno prva poslovnica u datoteci poslovnice.txt ima indeks 0, druga 1 itd.

```
using System.IO;

static void Main(string[] args)
{
    try
    {
        StreamReader reader = new StreamReader("poslovnice.txt");
        StreamWriter writer = new StreamWriter("poboljsanja.txt");
        int indeks = 0;

        while (!reader.EndOfStream)
        {
            double prihod = Convert.ToDouble(reader.ReadLine());

            if (prihod < 50000)
            {
                writer.WriteLine(indeks);
            }

            indeks++;
        }

        reader.Close();
        writer.Flush();
        writer.Close();
    }
    catch (Exception ex)
    {
        Console.WriteLine(ex.Message);
    }

    Console.ReadKey();
}
```



Zadatak 3

Napišite konzolnu aplikaciju koja generira 1000 slučajnih cijelih brojeva unutar intervala [80, 200] koji predstavljaju brzine vozila na prometnici [km/h].

U svakoj iteraciji provjerava se da li je generirana brzina veća od 140 km/h, te ako je dodaje ju se u listu prekoracenjaBrzina.

Na kraju je potrebno ispisati koliko je bilo brzina većih od 140 km/h.

```
static void Main(string[] args)
{
    Random rnd = new Random();
    List<int> prekoracenjaBrzina = new List<int>();

    for (int i = 1; i <= 1000; i++)
    {
        int brzina = rnd.Next(80, 201);

        if (brzina > 140)
        {
            prekoracenjaBrzina.Add(brzina);
        }
    }

    Console.WriteLine("Broj brzina iznad 140 km/h je "
        + prekoracenjaBrzina.Count + ".");

    Console.ReadKey();
}
```



Zadatak 4

Napišite program koji učitava 6 cjelobrojnih vrijednosti iz konzole u pravokutnu matricu (2×3). Pritom je potrebno koristiti petlje po izboru.

Nakon što su učitani svi elementi matrice potrebno je ispisati one elemente čija je vrijednost veća od 3. Primjerice, ako se učitava matrica:

1, 2, 4

3, 5, 6

potrebno je ispisati:

Elementi matrice veći od 3 su: 4 5 6

```
static void Main(string[] args)
{
    int[,] matrica = new int[2, 3];

    for (int i = 0; i < matrica.GetLength(0); i++)
    {
        for (int j = 0; j < matrica.GetLength(1); j++)
        {
            Console.Write("Unesite element matrice na indeksu [" + i + ", "
                + j + "]: ");
            matrica[i, j] = Convert.ToInt32(Console.ReadLine());
        }
    }

    Console.Write("Elementi matrice veći od 3 su: ");
    for (int i = 0; i < matrica.GetLength(0); i++)
    {
        for (int j = 0; j < matrica.GetLength(1); j++)
        {
            if (matrica[i, j] > 3)
            {
                Console.Write(matrica[i, j] + " ");
            }
        }
    }

    Console.ReadKey();
}
```



Zadatak 5

Napišite konzolnu aplikaciju koja učitava dva realna broja m i n , koji označavaju redom masu jednog paketa [kg] i broj paketa koji je potrebno ukrcati u kamion.

Potrebno je izračunati ukupnu masu koju je potrebno ukrcati u kamion te pozvati metodu `ProvjeraMase`.

Metoda kao argument prima iznos mase koji je potrebno ukrcati, te provjerava i ispisuje poruku da li je premašena dopuštena nosiva masa vozila u iznosu od 1000 kg. Metoda ne vraća rezultat, a format ispisa treba biti kao u primjerima.

Primjer: ako se redom učitaju vrijednosti 100 i 15 potrebno je ispisati: „Ukupna masa iznosi 1500 kg, te premašuje dopuštenu nosivost vozila.“.

Ako se redom učitaju vrijednosti 10 i 10 potrebno je ispisati: „Ukupna masa iznosi 100 kg, te ne premašuje dopuštenu nosivost vozila.“.

```
static void Main(string[] args)
{
    Console.Write("Unesite jediničnu masu paketa [kg]: ");
    double m = Convert.ToDouble(Console.ReadLine());

    Console.Write("Unesite broj paketa koji je potrebno ukrcati u kamion: ");
    double n = Convert.ToDouble(Console.ReadLine());

    double ukupnaMasa = m * n;

    ProveraMase(ukupnaMasa);

    Console.ReadKey();
}

static void ProveraMase(double masa)
{
    if (masa > 1000)
    {
        Console.WriteLine("Ukupna masa iznosi " + masa + " kg, te premašuje dopuštenu nosivost vozila.");
    }
    else
    {
        Console.WriteLine("Ukupna masa iznosi " + masa + " kg, te ne premašuje dopuštenu nosivost vozila.");
    }
}
```



Ispit #24 (06.09.18. - grupa B)

Zadatak 1

Nadopunite sljedeći kod s metodama UvecajKilometre i PromijeniVlasnika tako da njihovi parametri odgovaraju pozivu iz Main metode.

Metoda UvecajKilometre treba uvećati kilometre za zadani broj koji se predaje preko argumenta, dok metoda PromijeniVlasnika treba promijeniti ime vlasnika u ime koje se predaje preko argumenta.

```
static void Main(string[] args)
{
    Auto a = new Auto("Insignia", 50000, "Marko");
    a.UvecajKilometre(10000);
    a.PromijeniVlasnika("Ivo");

    Console.ReadKey();
}

public class Auto
{
    private string imeAuta;
    private int kilometri;
    private string imeVlasnika;

    public Auto(string IMEAUTA, int KILOMETRI, string IMEVLASNIKA)
    {
        imeAuta = IMEAUTA;
        kilometri = KILOMETRI;
        imeVlasnika = IMEVLASNIKA;
    }

    //Nadopuni
    public void UvecajKilometre(int KILOMETRI)
    {
        kilometri += KILOMETRI;
    }

    public void PromijeniVlasnika(string IMEVLASNIKA)
    {
        imeVlasnika = IMEVLASNIKA;
    }
}
```



Zadatak 2

Napišite konzolnu aplikaciju koja iz datoteke udaljenosti.txt učitava godišnju prijeđenu udaljenost vozila u voznom parku. Za svako vozilo u zasebnom retku navedena je ukupna godišnja prijeđena udaljenost kao realan broj u kilometrima.

Potrebno je u drugu datoteku servisVozila.txt zapisati indekse redaka onih vozila čija je godišnja prijeđena udaljenost veća od 50000 km. Smatra se da prvi redak, odnosno prvo vozilo u datoteci udaljenosti.txt ima indeks 0, drugi 1 itd.

```
using System.IO;

static void Main(string[] args)
{
    try
    {
        StreamReader reader = new StreamReader("udaljenosti.txt");
        StreamWriter writer = new StreamWriter("servisVozila.txt");
        int indeks = 0;

        while (!reader.EndOfStream)
        {
            double udaljenost = Convert.ToDouble(reader.ReadLine());

            if (udaljenost > 50000)
            {
                writer.WriteLine(indeks);
            }

            indeks++;
        }

        reader.Close();
        writer.Flush();
        writer.Close();
    }
    catch (Exception ex)
    {
        Console.WriteLine("Greška: " + ex.Message);
    }

    Console.ReadKey();
}
```



Zadatak 3

Napišite konzolnu aplikaciju koja generira 1000 slučajnih cijelih brojeva unutar intervala $[0, 5>$ koji predstavljaju simulirane vrijednosti veličina paketa $[m^3]$ na traci koje je potrebno spremiti u vozilo.

U svakoj iteraciji provjerava se da li je generirana veličina paketa manja od $3 m^3$, te ako je dodaje ju se u listu paketi.

Na kraju je potrebno ispisati koliko je bilo paketa čija je veličina manja od $3 m^3$.

```
static void Main(string[] args)
{
    Random rnd = new Random();
    List<int> paketi = new List<int>();

    for (int i = 1; i <= 1000; i++)
    {
        int velicinaPaketa = rnd.Next(0, 5);

        if (velicinaPaketa < 3)
        {
            paketi.Add(velicinaPaketa);
        }
    }

    Console.WriteLine("Broj paketa koji su manji od 3 m^3 je "
        + paketi.Count + ".");

    Console.ReadKey();
}
```



Zadatak 4

Napišite program koji učitava 6 cjelobrojnih vrijednosti iz konzole u pravokutnu matricu (3×2). Pritom je potrebno koristiti petlje po izboru.

Nakon što su učitani svi elementi matrice potrebno je ispisati one elemente čija je vrijednost manja od -1. Primjerice ako se učitava matrica:

-1, -2

-3, 3

4, -5

potrebno je ispisati:

Elementi matrice manji od -1 su: -2 -3 -5

```
static void Main(string[] args)
{
    int[,] matrica = new int[2, 3];

    for (int i = 0; i < matrica.GetLength(0); i++)
    {
        for (int j = 0; j < matrica.GetLength(1); j++)
        {
            Console.Write("Unesite element matrice na indeksu ["
                + i + ", " + j + "]: ");
            matrica[i, j] = Convert.ToInt32(Console.ReadLine());
        }
    }

    Console.Write("Elementi matrice manji od -1 su: ");
    for (int i = 0; i < matrica.GetLength(0); i++)
    {
        for (int j = 0; j < matrica.GetLength(1); j++)
        {
            if (matrica[i, j] < -1)
            {
                Console.Write(matrica[i, j] + " ");
            }
        }
    }

    Console.ReadKey();
}
```



Zadatak 5

Napišite konzolnu aplikaciju koja učitava dva realna broja v i s , koji označavaju redom prosječnu brzinu kretanja vozila [km/h] i duljinu prometnice [km].

Potrebno je napisati metodu `VrijemePutovanja` koja kao argumente prima prosječnu brzinu kretanja vozila i duljinu prometnice, a kao rezultat vraća vrijeme u satima potrebno za prolazak prometnice.

U glavnom dijelu programa potrebno je provjeriti da li je vrijeme putovanja veće od 2 h ili ne, te ispisati poruku kao u primjerima.

Primjer: Ako se redom učitaju vrijednosti 100 i 280 potrebno je ispisati: „Vrijeme putovanja iznosi 2.8 h, te vozilo neće na vrijeme stići na odredište.“.

Ako se redom učitaju vrijednosti 100 i 150 potrebno je ispisati: „Vrijeme putovanja iznosi 1.5 h, te će vozilo na vrijeme stići na odredište.“.

```
static void Main(string[] args)
{
    Console.Write("Unesite prosječnu brzinu kretanja vozila [km/h]: ");
    double v = Convert.ToDouble(Console.ReadLine());

    Console.Write("Unesite duljinu prometnice [km]: ");
    double s = Convert.ToDouble(Console.ReadLine());

    double vrijeme = VrijemePutovanja(v, s);

    if (vrijeme > 2)
    {
        Console.WriteLine("Vrijeme putovanja iznosi " + vrijeme
            + " h, te vozilo neće na vrijeme stići na odredište.");
    }
    else
    {
        Console.WriteLine("Vrijeme putovanja iznosi " + vrijeme
            + " h, te će vozilo na vrijeme stići na odredište.");
    }

    Console.ReadKey();
}

static double VrijemePutovanja(double brzinaVozila, double duljinaPrometnice)
{
    double vrijeme = duljinaPrometnice / brzinaVozila;

    return vrijeme;
}
```



Ispit #25 (17.11.18. - grupa A)

Zadatak 1

Napišite kod klase Student koja sadrži javne attribute Ime, Prezime i JMBAG, te privatni atribut Ocjena.

Konstruktor klase prima argumente imena, prezimena i jmbag, a ocjenu inicijalizira na vrijednost 1.

Potrebno je također napisati i javnu metodu PostaviOcjenu koja kao argument prima ocjenu te postavlja studentu zadanu ocjenu.

```
class Student
{
    public string Ime;
    public string Prezime;
    public string JMBAG;
    private int Ocjena;

    public Student(string ime, string prezime, string jmbag)
    {
        Ime = ime;
        Prezime = prezime;
        JMBAG = jmbag;
        Ocjena = 1;
    }

    public void PostaviOcjenu(int ocjena)
    {
        Ocjena = ocjena;
    }
}
```



Zadatak 2

Napišite program koji učitava šest realnih vrijednosti s tipkovnice, redom x1, y1, t1, x2, y2 i t2 koje predstavljaju geografske koordinate (x, y) i vremenski trenutak zapisa signala t (u satima [h]).

Pomoću geografskih koordinata T1 (x1, y1) i T2 (x2, y2) potrebno je izračunati udaljenost između točaka. Udaljenost se računa kao Euklidska udaljenost $d(T_1, T_2) = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$ pomnožena s vrijednošću 111.32 kako bi se dobio iznos u kilometrima.

Potrebno je izračunati i ispisati prosječnu brzinu kretanja vozila te ispisati da li se vozilo kreće iznad ili ispod ograničenja koje iznosi 100 km/h. Format ispisa mora biti kao u primjeru.

Primjerice ako se redom učitaju vrijednosti 15.7, 45.3, 4, 16.6, 44.2, 5.5 potrebno je ispisati: „Prosječna brzina kretanja vozila iznosi: 105.47 km/h! Vozilo se kreće iznad ograničenja!“.

```
static void Main(string[] args)
{
    Console.WriteLine("Unesite x1: ");
    double x1 = Convert.ToDouble(Console.ReadLine());
    Console.WriteLine("Unesite y1: ");
    double y1 = Convert.ToDouble(Console.ReadLine());
    Console.WriteLine("Unesite t1: ");
    double t1 = Convert.ToDouble(Console.ReadLine());

    Console.WriteLine("Unesite x2: ");
    double x2 = Convert.ToDouble(Console.ReadLine());
    Console.WriteLine("Unesite y2: ");
    double y2 = Convert.ToDouble(Console.ReadLine());
    Console.WriteLine("Unesite t2: ");
    double t2 = Convert.ToDouble(Console.ReadLine());

    double udaljenost = Math.Sqrt(Math.Pow((x1 - x2), 2)
    + Math.Pow((y1 - y2), 2)) * 111.32;

    double brzinaKretanja = udaljenost / (t2 - t1);

    if (brzinaKretanja <= 100)
    {
        Console.WriteLine("Prosječna brzina kretanja vozila iznosi: "
        + brzinaKretanja + " km/h! Vozilo se kreće unutar ograničenja!");
    }
    else
    {
        Console.WriteLine("Prosječna brzina kretanja vozila iznosi: "
        + brzinaKretanja + " km/h! Vozilo se kreće iznad ograničenja!");
    }

    Console.ReadKey();
}
```



Zadatak 3

Napišite konzolnu aplikaciju koja generira 1000 slučajnih cijelih brojeva u intervalu [5, 20] koji predstavljaju simulirane vrijednosti očitanih duljina vozila s radara.

Potrebno je prebrojati koliko je vrijednosti duljina vozila unutar intervala [5, 10>, a koliko unutar intervala [10, 20].

Program treba na kraju u proizvoljnom zapisu zapisati navedene dvije vrijednosti u datoteku vozila.txt.

```
static void Main(string[] args)
{
    Random rnd = new Random();
    int brojac1 = 0;
    int brojac2 = 0;

    for (int i = 1; i <= 1000; i++)
    {
        int duljinaVozila = rnd.Next(5, 21);

        if (duljinaVozila < 10)
        {
            brojac1++;
        }
        else
        {
            brojac2++;
        }
    }

    StreamWriter writer = new StreamWriter("vozila.txt");
    writer.WriteLine("Broj vozila s duljinom unutar intervala [5,10>: "
        + brojac1);
    writer.WriteLine("Broj vozila s duljinom unutar intervala [10,20]: "
        + brojac2);

    writer.Flush();
    writer.Close();

    Console.ReadKey();
}
```



Zadatak 4

Napišite konzolnu aplikaciju koja učitava sadržaj datoteke `dostava_17_11_2018.txt` u kojoj su upisani realni brojevi koji predstavljaju udaljenosti (u kilometrima) koje su dostavna vozila u tvrtki prošla u današnjem danu.

Potrebno je izračunati koliko je ukupno utrošeno kuna na dostavu ako je potrošnja 0.6 kn/km. Format ispisa je proizvoljan.

Potrebno je koristiti try-catch blok.

```
using System.IO;

static void Main(string[] args)
{
    try
    {
        StreamReader reader = new StreamReader("dostava_17_11_2018.txt");
        double ukupnoKilometara = 0;

        while (!reader.EndOfStream)
        {
            double udaljenost = Convert.ToDouble(reader.ReadLine());
            ukupnoKilometara += udaljenost;
        }

        reader.Close();

        double trosak = ukupnoKilometara * 0.6;
        Console.WriteLine("Za dostavu je utrošeno " + trosak + " kn.");
    }
    catch (Exception ex)
    {
        Console.WriteLine("Greška: " + ex.Message);
    }

    Console.ReadKey();
}
```



Zadatak 5

Napišite konzolnu aplikaciju koja učitava matricu udaljenosti između n gradova.

Potrebno je prvo preko tipkovnice unijeti vrijednost n , potom kreirati matricu $M = n \times n$, te ju popuniti realnim brojevima koji predstavljaju udaljenosti između gradova.

Na kraju je potrebno ispisati sadržaj matrice u proizvoljnom formatu.

```
static void Main(string[] args)
{
    Console.Write("Unesite broj gradova: ");
    int n = Convert.ToInt32(Console.ReadLine());

    double[,] M = new double[n, n];

    for (int i = 0; i < M.GetLength(0); i++)
    {
        for (int j = 0; j < M.GetLength(1); j++)
        {
            Console.Write("Unesite udaljenost između gradova koji se nalaze na indeksu [" + i + ", " + j + "]: ");
            M[i, j] = Convert.ToDouble(Console.ReadLine());
        }
    }

    Console.Write("Udaljenost između gradova su:\n\t");
    for (int i = 0; i < M.GetLength(1); i++)
    {
        Console.Write(i + "\t");
    }
    Console.WriteLine();

    for (int i = 0; i < M.GetLength(0); i++)
    {
        Console.Write(i + "\t");
        for (int j = 0; j < M.GetLength(1); j++)
        {
            Console.Write(M[i, j] + "\t");
        }
        Console.WriteLine();
    }

    Console.ReadKey();
}
```



Ispit #26 (07.02.19. - grupa A)

Zadatak 1

Napišite kod klase Kolegij koji sadrži privatne attribute Naziv, ECTS, Nositelj i Broj_studenata.

Konstruktor klase prima argumente naziva, ects-a i nositelja, a broj studenata inicijalizira na vrijednost 0.

Potrebno je napisati i javnu metodu Ispis koja ispisuje naziv, ects-e, nositelja i broj studenata u formatu: „Kolegij Naziv, ima ECTS ects-a, nositelj kolegija je Nositelj, a kolegij je upisalo Broj_studenata studenata!“. Umjesto vrijednosti u kurzivu potrebno je upisati odgovarajuće vrijednosti.

```
class Kolegij
{
    private string Naziv;
    private int ECTS;
    private string Nositelj;
    private int Broj_studenata;

    public Kolegij(string naziv, int ects, string nositelj)
    {
        Naziv = naziv;
        ECTS = ects;
        Nositelj = nositelj;
        Broj_studenata = 0;
    }

    public void Ispis()
    {
        Console.WriteLine("Kolegij " + Naziv + ", ima " + ECTS
            + " ects-a, nositelj kolegija je " + Nositelj
            + ", a kolegij je upisalo " + Broj_studenata + " studenata!");
    }
}
```



Zadatak 2

Napišite program koji učitava šest realnih vrijednosti s tipkovnice, redom x1, y1, t1, x2, y2 i t2 koje predstavljaju geografske koordinate (x, y) i vremenski trenutak zapisa signala t (u satima [h]).

Pomoću geografskih koordinata T1 (x1, y1) i T2 (x2, y2) potrebno je izračunati udaljenost između točaka. Udaljenost se računa kao Euklidska udaljenost $d(T_1, T_2) = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$ pomnožena s vrijednošću 111.32 kako bi se dobio iznos u kilometrima.

Potrebno je izračunati i ispisati prosječnu brzinu kretanja vozila te ispisati da li se vozilo kreće iznad ili ispod ograničenja koje iznosi 100 km/h. Format ispisa mora biti kao u primjeru.

Primjerice ako se redom učitaju vrijednosti 15.7, 45.3, 4, 16.6, 44.2, 5.5 potrebno je ispisati: „Prosječna brzina kretanja vozila iznosi: 105.47 km/h! Vozilo se kreće iznad ograničenja!“.

```
static void Main(string[] args)
{
    Console.WriteLine("Unesite x1: ");
    double x1 = Convert.ToDouble(Console.ReadLine());
    Console.WriteLine("Unesite y1: ");
    double y1 = Convert.ToDouble(Console.ReadLine());
    Console.WriteLine("Unesite t1: ");
    double t1 = Convert.ToDouble(Console.ReadLine());

    Console.WriteLine("Unesite x2: ");
    double x2 = Convert.ToDouble(Console.ReadLine());
    Console.WriteLine("Unesite y2: ");
    double y2 = Convert.ToDouble(Console.ReadLine());
    Console.WriteLine("Unesite t2: ");
    double t2 = Convert.ToDouble(Console.ReadLine());

    double udaljenost = Math.Sqrt(Math.Pow((x1 - x2), 2)
    + Math.Pow((y1 - y2), 2)) * 111.32;

    double brzinaKretanja = udaljenost / (t2 - t1);

    if (brzinaKretanja <= 100)
    {
        Console.WriteLine("Prosječna brzina kretanja vozila iznosi: "
        + brzinaKretanja + " km/h! Vozilo se kreće unutar ograničenja!");
    }
    else
    {
        Console.WriteLine("Prosječna brzina kretanja vozila iznosi: "
        + brzinaKretanja + " km/h! Vozilo se kreće iznad ograničenja!");
    }

    Console.ReadKey();
}
```



Zadatak 3

Napišite konzolnu aplikaciju koja generira 1000 slučajnih cijelih brojeva u intervalu [5, 20] koji predstavljaju simulirane vrijednosti očitanih duljina vozila s radara.

Potrebno je prebrojati koliko je vrijednosti duljina vozila unutar intervala [5, 10>, a koliko unutar intervala [10, 20].

Program treba na kraju u proizvoljnom zapisu zapisati navedene dvije vrijednosti u datoteku vozila.txt.

```
using System.IO;

static void Main(string[] args)
{
    Random rnd = new Random();
    int brojac1 = 0;
    int brojac2 = 0;

    for (int i = 1; i <= 1000; i++)
    {
        int duljinaVozila = rnd.Next(5, 21);

        if (duljinaVozila < 10)
        {
            brojac1++;
        }
        else
        {
            brojac2++;
        }
    }

    StreamWriter writer = new StreamWriter("vozila.txt");
    writer.WriteLine("Broj vozila s duljinom unutar intervala [5,10>: "
        + brojac1);
    writer.WriteLine("Broj vozila s duljinom unutar intervala [10,20]: "
        + brojac2);

    writer.Flush();
    writer.Close();

    Console.ReadKey();
}
```



Zadatak 4

Napišite konzolnu aplikaciju koja učitava sadržaj datoteke `brzineNaPrometnici.txt` u kojoj su upisani cijeli brojevi svaki u zasebnom retku koji predstavljaju izmjerene brzine vozila na prometnici.

Nakon što se učitava vrijednost brzine `-1` potrebno je završiti učitavanje podataka i u konzolu ispisati broj zabilježenih brzina i prosječnu brzinu u proizvoljnom formatu.

Napomena: potrebno je koristiti try-catch blok, te nije potrebno dodati imenski prostor `System.IO`.

```
using System.IO;

static void Main(string[] args)
{
    try
    {
        StreamReader reader = new StreamReader("brzineNaPrometnici.txt");
        int brojBrzina = 0;
        int sumaBrzina = 0;

        while (!reader.EndOfStream)
        {
            int brzina = Convert.ToInt32(reader.ReadLine());

            if (brzina == -1)
            {
                break;
            }

            sumaBrzina += brzina;
            brojBrzina++;
        }

        reader.Close();

        double prosjecnaBrzina = sumaBrzina / Convert.ToDouble(brojBrzina);

        Console.WriteLine("Broj zabilježenih brzina je " + brojBrzina
            + ", a prosječna brzina je " + prosjecnaBrzina + " km/h.");
    }
    catch (Exception ex)
    {
        Console.WriteLine("Greška: " + ex.Message);
    }

    Console.ReadKey();
}
```



Zadatak 5

Napišite konzolnu aplikaciju koja učitava matricu udaljenosti između n gradova.

Potrebno je prvo preko tipkovnice unijeti vrijednost n , potom kreirati matricu $M = n \times n$, te ju popuniti realnim brojevima koji predstavljaju udaljenosti između gradova.

Na kraju je potrebno ispisati sadržaj matrice u proizvoljnom formatu.

```
static void Main(string[] args)
{
    Console.Write("Unesite broj gradova: ");
    int n = Convert.ToInt32(Console.ReadLine());

    double[,] M = new double[n, n];

    for (int i = 0; i < M.GetLength(0); i++)
    {
        for (int j = 0; j < M.GetLength(1); j++)
        {
            Console.Write("Unesite udaljenost između gradova koji se nalaze na indeksu [" + i + ", " + j + "]: ");
            M[i, j] = Convert.ToDouble(Console.ReadLine());
        }
    }

    Console.Write("Udaljenost između gradova su:\n\t");
    for (int i = 0; i < M.GetLength(1); i++)
    {
        Console.Write(i + "\t");
    }
    Console.WriteLine();

    for (int i = 0; i < M.GetLength(0); i++)
    {
        Console.Write(i + "\t");
        for (int j = 0; j < M.GetLength(1); j++)
        {
            Console.Write(M[i, j] + "\t");
        }
        Console.WriteLine();
    }

    Console.ReadKey();
}
```



Ispit #27 (21.02.19. - grupa A)

Zadatak 1

Napišite kod klase Student koja sadrži javne attribute Ime, Prezime i JMBAG, te privatni atribut Ocjena.

Konstruktor klase prima argumente imena, prezimena i jmbag, a ocjenu inicijalizira na vrijednost 1.

Potrebno je također napisati i javnu metodu PostaviOcjenu koja kao argument prima ocjenu te postavlja studentu zadanu ocjenu.

```
class Student
{
    public string Ime;
    public string Prezime;
    public string JMBAG;
    private int Ocjena;

    public Student(string ime, string prezime, string jmbag)
    {
        Ime = ime;
        Prezime = prezime;
        JMBAG = jmbag;
        Ocjena = 1;
    }

    public void PostaviOcjenu(int ocjena)
    {
        Ocjena = ocjena;
    }
}
```



Zadatak 2

Napišite konzolnu aplikaciju koja učitava dva realna broja D i a , koji predstavljaju diskriminantu i vrijednost prvog koeficijenta kvadratne funkcije.

Ako su vrijednosti D i a manje od nule potrebno je ispisati „Za D i a funkcija ne siječe x-os te su joj sve vrijednosti negativne!“.

Ako je D manje od nule, a a veće od nule potrebno je ispisati „Za D i a funkcija ne siječe x-os te su joj sve vrijednosti pozitivne!“.

Inače je potrebno ispisati „Funkcija siječe x-os!“.

Primjer: ako se za D učitava vrijednost -1.25 , a za a vrijednost 0.5 , potrebno je ispisati: „Za $D = -1.25$ i $a = 0.5$ funkcija ne siječe x-os te su joj sve vrijednosti pozitivne!“.

```
static void Main(string[] args)
{
    Console.Write("Unesite vrijednost diskriminante D: ");
    double D = Convert.ToDouble(Console.ReadLine());

    Console.Write("Unesite vrijednost koeficijenta a: ");
    double a = Convert.ToDouble(Console.ReadLine());

    if (D < 0 && a < 0)
    {
        Console.WriteLine("Za D = " + D + " i a = " + a
            + " funkcija ne siječe x-os te su joj sve vrijednosti negativne!");
    }
    else if (D < 0 && a > 0)
    {
        Console.WriteLine("Za D = " + D + " i a = " + a
            + " funkcija ne siječe x-os te su joj sve vrijednosti pozitivne!");
    }
    else
    {
        Console.WriteLine("Funkcija siječe x-os!");
    }

    Console.ReadKey();
}
```



Zadatak 3

Napišite konzolnu aplikaciju koja generira 1000 slučajnih cijelih brojeva u intervalu [5, 20] koji predstavljaju simulirane vrijednosti očitanih duljina vozila s radara.

Potrebno je prebrojati koliko je vrijednosti duljina vozila unutar intervala [5, 10>, te izračunati prosječnu duljinu vozila čija je duljina u intervalu [5, 10>.

Program treba na kraju u proizvoljnom zapisu zapisati navedene dvije vrijednosti u datoteku vozila.txt.

```
using System.IO;

static void Main(string[] args)
{
    Random rnd = new Random();
    int brojTrazenih = 0;
    int sumaTrazenih = 0;

    for (int i = 1; i <= 1000; i++)
    {
        int duljinaVozila = rnd.Next(5, 21);

        if (duljinaVozila < 10)
        {
            brojTrazenih++;
            sumaTrazenih += duljinaVozila;
        }
    }

    double prosjekDuljina = sumaTrazenih / Convert.ToDouble(brojTrazenih);
    StreamWriter writer = new StreamWriter("vozila.txt");

    writer.WriteLine("Broj vozila s duljinom unutar intervala [5,10>: "
        + brojTrazenih);
    writer.WriteLine("Prosječna duljina vozila s duljinom unutar intervala
        [5,10>: " + prosjekDuljina);

    writer.Flush();
    writer.Close();

    Console.ReadKey();
}
```



Zadatak 4

Napišite konzolnu aplikaciju koja učitava sadržaj datoteke `brzineNaPrometnici.txt` u kojoj su upisani cijeli brojevi svaki u zasebnom retku koji predstavljaju izmjerene brzine vozila na prometnici.

Nakon što se učitava vrijednost brzine `-1` potrebno je završiti učitavanje podataka i u konzolu ispisati broj zabilježenih brzina i prosječnu brzinu u proizvoljnom formatu.

Napomena: potrebno je koristiti try-catch blok, te nije potrebno dodati imenski prostor `System.IO`.

```
using System.IO;

static void Main(string[] args)
{
    try
    {
        StreamReader reader = new StreamReader("brzineNaPrometnici.txt");
        int brojBrzina = 0;
        int sumaBrzina = 0;

        while (!reader.EndOfStream)
        {
            int brzina = Convert.ToInt32(reader.ReadLine());

            if (brzina == -1)
            {
                break;
            }

            sumaBrzina += brzina;
            brojBrzina++;
        }

        reader.Close();

        double prosjecnaBrzina = sumaBrzina / Convert.ToDouble(brojBrzina);

        Console.WriteLine("Broj zabilježenih brzina je " + brojBrzina
            + ", a prosječna brzina je " + prosjecnaBrzina + " km/h.");
    }
    catch (Exception ex)
    {
        Console.WriteLine("Greška: " + ex.Message);
    }

    Console.ReadKey();
}
```



Zadatak 5

Napišite konzolnu aplikaciju koja učitava matricu udaljenosti između n gradova.

Potrebno je prvo preko tipkovnice unijeti vrijednost n , potom kreirati matricu $M = n \times n$, te ju popuniti realnim brojevima koji predstavljaju udaljenosti između gradova.

Na kraju je potrebno ispisati sadržaj matrice u proizvoljnom formatu.

```
static void Main(string[] args)
{
    Console.Write("Unesite broj gradova: ");
    int n = Convert.ToInt32(Console.ReadLine());

    double[,] M = new double[n, n];

    for (int i = 0; i < M.GetLength(0); i++)
    {
        for (int j = 0; j < M.GetLength(1); j++)
        {
            Console.Write("Unesite udaljenost između gradova koji se nalaze na indeksu [" + i + ", " + j + "]: ");
            M[i, j] = Convert.ToDouble(Console.ReadLine());
        }
    }

    Console.Write("Udaljenost između gradova su:\n\t");
    for (int i = 0; i < M.GetLength(1); i++)
    {
        Console.Write(i + "\t");
    }
    Console.WriteLine();

    for (int i = 0; i < M.GetLength(0); i++)
    {
        Console.Write(i + "\t");
        for (int j = 0; j < M.GetLength(1); j++)
        {
            Console.Write(M[i, j] + "\t");
        }
        Console.WriteLine();
    }

    Console.ReadKey();
}
```



Ispit #28 (13.04.19. - grupa A)

Zadatak 1

Nadopunite sljedeći kod unutar Main metode na način da stvorite novi grad Zagreb s koordinatama x = 15.9773, y = 45.8129 i brojem stanovnika 800000.

```
static void Main(string[] args)
{
    //Nadopuni

    Tocka t = new Tocka(45.8129, 15.9773);
    Grad g = new Grad(800000, "Zagreb", t);
}

public class Grad
{
    private string naziv;
    public Tocka t;
    private int brojStanovnika;

    public Grad(int BROJSTANOVNIKA, string NAZIV, Tocka T)
    {
        t = T;
        naziv = NAZIV;
        brojStanovnika = BROJSTANOVNIKA;
    }
}

public class Tocka
{
    public double x;
    public double y;

    public Tocka(double Y, double X)
    {
        x = X;
        y = Y;
    }
}
```



Zadatak 2

Napišite konzolnu aplikaciju koja učitava cijeli broj koji predstavlja izbor u izborniku: 1 - površina pravokutnika, 2 - opseg kružnice.

Ukoliko se unese 1 potrebno je dodatno zatražiti korisnika unos stranica pravokutnika a i b (realni brojevi) te izračunati i ispisati površinu pravokutnika samo ako su a i b veći od 0. Ukoliko a i b nisu veći od nula potrebno je ispisati „Neispravan unos!“.

Ukoliko se unese 2 potrebno je korisnika zatražiti unos polumjera kružnice r (realan broj) te izračunati i ispisati opseg kružnice ($O = 2\pi r$).

Ukoliko se ne unese 1 ili 2 za izbor potrebno je ispisati „Neispravan unos!“.

Format ispisa mora biti kao u primjeru, tako da se ispišu učitani parametri i izračunata vrijednost.

Primjer: ako se učitava prvo vrijednost 1, a potom vrijednosti 5.5 i 6.7 potrebno je ispisati: „Površina pravokutnika određenog stranicama $a=5.5$ i $b=6.7$ iznosi $P=36.85!$ “.



```

static void Main(string[] args)
{
    Console.WriteLine("Odaberite željenu opciju:");
    Console.WriteLine("1: Površina pravokutnika");
    Console.WriteLine("2: Opseg kružnice");

    Console.Write("Vaš izbor: ");
    int izbor = Convert.ToInt32(Console.ReadLine());

    if (izbor == 1)
    {
        Console.WriteLine("Unesite stranicu a pravokutnika: ");
        double a = Convert.ToDouble(Console.ReadLine());

        Console.WriteLine("Unesite stranicu b pravokutnika: ");
        double b = Convert.ToDouble(Console.ReadLine());

        if (a > 0 && b > 0)
        {
            double P = a * b;
            Console.WriteLine("Površina pravokutnika određenog stranicama a="
                + a + " i b=" + b + " iznosi P=" + P + "!");
        }
        else
        {
            Console.WriteLine("Neipsravan unos!");
        }
    }
    else if (izbor == 2)
    {
        Console.WriteLine("Unesite polumjer kružnice r: ");
        double r = Convert.ToDouble(Console.ReadLine());

        double O = 2 * r * Math.PI;
        Console.WriteLine("Opseg kružnice polumjera r=" + r + " iznosi O=" + O
            + "!");
    }
    else
    {
        Console.WriteLine("Neispravan unos!");
    }

    Console.ReadKey();
}

```



Zadatak 3

Napišite metodu Prosjek koja kao argument prima listu cijelih brojeva. Metoda računa prosječnu vrijednost elemenata u listi i vraća rezultat.

```
static double Prosjek(List<int> lista)
{
    int zbroj = 0;

    for (int i = 0; i < lista.Count; i++)
    {
        zbroj += lista[i];
    }

    double prosjek = zbroj / Convert.ToDouble(lista.Count);

    return prosjek;
}
```



Zadatak 4

Napišite konzolnu aplikaciju koja učitava sadržaj datoteke brzine.txt u kojoj su upisani cijeli brojevi svaki u zasebnom retku koji predstavljaju izmjerene brzine vozila na prometnici.

Potrebno je stvoriti dvije liste, jednu za brzine manje od ograničenja brzine koje iznosi 60 km/h i jednu za brzine veće ili jednake 60 km/h. Nakon što se učitava vrijednost brzine potrebno je učitanu brzinu dodati u odgovarajuću listu.

Na kraju je potrebno pozvati metodu prosjek iz zadatka 3 za svaku od lista kako bi se ispisala prosječna brzina lista u proizvoljnom formatu.

Napomena: potrebno je koristiti try-catch blok, te nije potrebno dodati imenski prostor System.IO.

`using System.IO;`

```
static void Main(string[] args)
{
    try
    {
        StreamReader reader = new StreamReader("brzine.txt");
        List<int> manjeOd60 = new List<int>();
        List<int> veceIliJednako60 = new List<int>();

        while (!reader.EndOfStream)
        {
            int brzina = Convert.ToInt32(reader.ReadLine());

            if (brzina < 60)
            {
                manjeOd60.Add(brzina);
            }
            else
            {
                veceIliJednako60.Add(brzina);
            }
        }

        reader.Close();

        double prosjekManjeOd60 = Prosjek(manjeOd60);
        double prosjekVeceIliJednako60 = Prosjek(veceIliJednako60);

        Console.WriteLine("Prosjek brzina do 60 km/h je " + prosjekManjeOd60
            + " km/h.");
        Console.WriteLine("Prosjek brzina većih ili jednakih 60 km/h je "
            + prosjekVeceIliJednako60 + " km/h.");
    }
    catch (Exception ex)
    {
        Console.WriteLine("Greška: " + ex.Message);
    }

    Console.ReadKey();
}
```



Zadatak 5

Napišite konzolnu aplikaciju koja učitava sadržaj za n gradova pri čemu se n unosi preko tipkovnice.

Za svaki grad unosi se naziv grada (niz znakova) i udaljenost u kilometrima od trenutne lokacije (realan broj).

Prema tome potrebno je kreirati dva jednodimenzionalna polja te ih popuniti odgovarajućim sadržajem.

Potrebno je ispisati naziv i pripadajuću udaljenost za 4 slučajno dohvaćena grada iz polja (mogu biti i isti gradovi). Format ispisa je proizvoljan.

```
static void Main(string[] args)
{
    Console.Write("Unesite broj gradova n: ");
    int n = Convert.ToInt32(Console.ReadLine());

    string[] gradovi = new string[n];
    double[] udaljenosti = new double[n];

    for (int i = 0; i < gradovi.Length; i++)
    {
        Console.Write("Unesite naziv " + (i + 1) + " grada: ");
        gradovi[i] = Console.ReadLine();

        Console.Write("Unesite udaljenost od Vaše trenutne lokacije  
do grada \"" + gradovi[i] + "\": ");
        udaljenosti[i] = Convert.ToDouble(Console.ReadLine());
    }

    Random rnd = new Random();

    for (int i = 0; i < 4; i++)
    {
        int odabraniGrad = rnd.Next(0, gradovi.Length);
        Console.WriteLine("Grad \"" + gradovi[odabraniGrad]
            + "\" je od Vaše trenutne lokacije udaljen "
            + udaljenosti[odabraniGrad] + " km!");
    }

    Console.ReadKey();
}
```

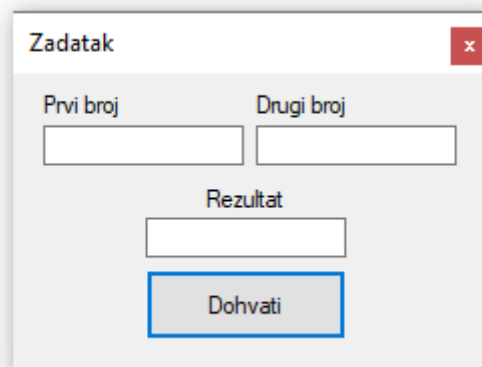


Ispit #29 (13.06.19. - grupa A)

Zadatak 1

Nadopunite sljedeći kod unutar metode BtnDohvati_Click na način da izračunate cjelobrojni ostatak dijeljenja prvog i drugog cijelog broja unesenih u TextBoxove txtPrviBroj i txtDrugiBroj.

Rezultat je također cijeli broj koji je potrebno zapisati u TextBox txtRezultat. Sučelje je prikazano slikom.



```
private void BtnDohvati_Click(object sender, EventArgs e)
{
    //Nadopuni

    int prviBroj = Convert.ToInt32(txtPrviBroj.Text);
    int drugiBroj = Convert.ToInt32(txtDrugiBroj.Text);

    int rezultat = prviBroj % drugiBroj;

    txtRezultat.Text = rezultat.ToString();
}
```



Zadatak 2

Potrebno je napisati metodu `IspisiDio` koja kao argument prima tri vrijednosti: niz znakova `s`, početni indeks `b` (cijeli broj) i krajnji indeks `e` (cijeli broj).

Metoda u konzolu treba ispisati znak po znak predanog znakovnog niza počevši od znaka na indeksu `b` do znaka na indeksu `e` (oba znaka su uključujuća).

Ispis se treba vršiti na način da se znakovi ispisuju u istom retku (bez prelaska pokazivača u novi red).

```
static void IspisiDio(string s, int b, int e)
{
    for (int i = b; i <= e; i++)
    {
        Console.Write(s[i]);
    }
}
```



Zadatak 3

Napišite konzolnu aplikaciju koja preko konzole učitava vrijeme putovanja t u satima (realan broj) i ID tipa vozila (cijeli broj).

ID tipa vozila dan je stupcem ID u tablici.

ID	Opis	Brzina [°/h]
1	Motor	$v_x = v_y = 0.7622$
2	Automobil	$v_x = v_y = 0.5717$
3	Kamion	$v_x = v_y = 0.5082$

Početna pozicija vozila u geografskim stupnjevima je $T(x_p, y_p) = (15.97790, 45.810857)$.

Ovisno o tome koji je ID vozila unesen potrebno je ispisati poziciju na kojoj će se vozilo naći nakon što se vozi t vremena, pri čemu su brzine zadane u tablici, a konačna pozicija $T(x_k, y_k)$ se računa koristeći izraze:

$$x_k = v_x * t + x_p$$

$$y_k = v_y * t + y_p$$

Format ispisa mora biti kao u primjeru. Primjerice, ukoliko se učitava vrijeme od 2 sata i vozilo pod ID-em 1, potrebno je ispisati: „Motor će se nakon 2 sata vožnje nalaziti na poziciji T (17.502305, 47.335257)!“.



```

static void Main(string[] args)
{
    Console.Write("Unesite vrijeme putovanja t u satima: ");
    double t = Convert.ToDouble(Console.ReadLine());

    Console.Write("Unesite tip vozila: ");
    int tipVozila = Convert.ToInt32(Console.ReadLine());

    double xp = 15.977905;
    double yp = 45.810857;
    double v = 0.0;
    double xk = 0.0;
    double yk = 0.0;

    if (tipVozila == 1)
    {
        v = 0.7622;
        xk = v * t + xp;
        yk = v * t + yp;
        Console.WriteLine("Motor će se nakon " + t
            + " sata vožnje nalaziti na poziciji T(" + xk + ", " + yk + ")!");
    }
    else if (tipVozila == 2)
    {
        v = 0.5717;
        xk = v * t + xp;
        yk = v * t + yp;
        Console.WriteLine("Automobil će se nakon " + t
            + " sata vožnje nalaziti na poziciji T(" + xk + ", " + yk + ")!");
    }
    else if (tipVozila == 3)
    {
        v = 0.5082;
        xk = v * t + xp;
        yk = v * t + yp;
        Console.WriteLine("Kamion će se nakon " + t
            + " sata vožnje nalaziti na poziciji T(" + xk + ", " + yk + ")!");
    }
    else
    {
        Console.WriteLine("Nepoznat tip vozila!");
    }

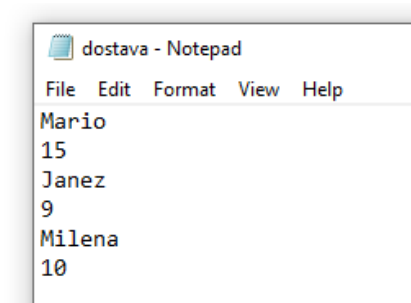
    Console.ReadKey();
}

```



Zadatak 4

Napišite konzolnu aplikaciju koja učitava sadržaj datoteke dostava.txt u kojoj su upisani podaci dostave na način prikazan slikom, odnosno ime dostavljača u jednom retku, pa u sljedećem retku količina paketa (cijeli broj) koju je taj dostavljač dostavio.



Potrebno je u konzolu ispisati količinu i ime dostavljača koji je taj dan dostavio najveći broj paketa. Format ispisa je proizvoljan.

Napomena: potrebno je koristiti try-catch blok, te nije potrebno dodati imenski prostor System.IO.

`using System.IO;`

```
static void Main(string[] args)
{
    try
    {
        StreamReader reader = new StreamReader("dostava.txt");

        int najveciBrojPaketa = 0;
        string najboljiDostavljac = "";

        while (!reader.EndOfStream)
        {
            string dostavljac = reader.ReadLine();
            int brojPaketa = Convert.ToInt32(reader.ReadLine());

            if (brojPaketa > najveciBrojPaketa)
            {
                najveciBrojPaketa = brojPaketa;
                najboljiDostavljac = dostavljac;
            }
        }

        reader.Close();

        Console.WriteLine("Najveći broj paketa (" + najveciBrojPaketa
            + ") dostavio je " + najboljiDostavljac + "!");
    }
    catch (Exception ex)
    {
        Console.WriteLine("Greška: " + ex.Message);
    }

    Console.ReadKey();
}
```



Zadatak 5

Napišite konzolnu aplikaciju za simulaciju bacanja kocke. Bacanje jedne kocke može rezultirati s jednom od šest slučajnih vrijednosti (cijeli brojevi) 1, 2, 3, 4, 5 ili 6.

Potrebno je simulirati tisuću (1000) bacanja jedne kocke te na kraju u proizvoljnom formatu u datoteku vjerojatnosti.txt ispisati vjerojatnosti pojave svake od vrijednosti 1, 2, 3, 4, 5 i 6.



```

using System.IO;

static void Main(string[] args)
{
    Random rnd = new Random();
    int brojac1 = 0;
    int brojac2 = 0;
    int brojac3 = 0;
    int brojac4 = 0;
    int brojac5 = 0;
    int brojac6 = 0;
    for (int i = 1; i < 1000; i++)
    {
        int bacanjeKockice = rnd.Next(1, 7);
        switch (bacanjeKockice)
        {
            case 1:
                brojac1++;
                break;
            case 2:
                brojac2++;
                break;
            case 3:
                brojac3++;
                break;
            case 4:
                brojac4++;
                break;
            case 5:
                brojac5++;
                break;
            case 6:
                brojac6++;
                break;
        }
    }
    double vjerojatnost1 = (brojac1 / 1000.0) * 100;
    double vjerojatnost2 = (brojac2 / 1000.0) * 100;
    double vjerojatnost3 = (brojac3 / 1000.0) * 100;
    double vjerojatnost4 = (brojac4 / 1000.0) * 100;
    double vjerojatnost5 = (brojac5 / 1000.0) * 100;
    double vjerojatnost6 = (brojac6 / 1000.0) * 100;
    StreamWriter writer = new StreamWriter("vjerojatnosti.txt");

    writer.WriteLine("Vjerojatnost pojave broja 1 je " + vjerojatnost1
        + "%.");
    writer.WriteLine("Vjerojatnost pojave broja 2 je " + vjerojatnost2
        + "%.");
    writer.WriteLine("Vjerojatnost pojave broja 3 je " + vjerojatnost3
        + "%.");
    writer.WriteLine("Vjerojatnost pojave broja 4 je " + vjerojatnost4
        + "%.");
    writer.WriteLine("Vjerojatnost pojave broja 5 je " + vjerojatnost5
        + "%.");
    writer.WriteLine("Vjerojatnost pojave broja 6 je " + vjerojatnost6
        + "%.");
    writer.Flush();
    writer.Close();

    Console.ReadKey();
}

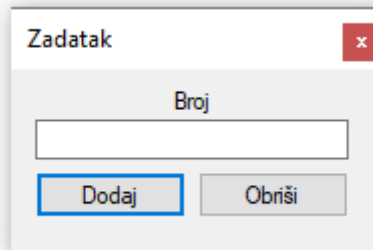
```



Ispit #30 (13.06.19. - grupa B)

Zadatak 1

Nadopunite sljedeći kod unutar odgovarajućih metoda na način da klikom na gumb Dodaj dodate u listu vrijednost broja upisanu u TextBox txtBroj, a klikom na gumb Obriši obrišete iz liste vrijednost broja upisanu u TextBox txtBroj. Sučelje je prikazano slikom.



```
//Nadopuni
List<int> lista = new List<int>();

private void BtnDodaj_Click(object sender, EventArgs e)
{
    lista.Add(Convert.ToInt32(txtBroj.Text));
}

private void BtnObriši_Click(object sender, EventArgs e)
{
    lista.Remove(Convert.ToInt32(txtBroj.Text));
}
```



Zadatak 2

Napišite metodu `prebrojiA` koja preko argumenata prima niz znakova `s`. Metoda treba prebrojati koliko znakova u nizu znakova `s` je jednako znaku (slovu) `a` te vratiti navedeni rezultat (cijeli broj).

```
static int prebrojiA(string s)
{
    int brojA = 0;

    for (int i = 0; i < s.Length; i++)
    {
        if (s[i] == 'a')
        {
            brojA++;
        }
    }

    return brojA;
}
```



Zadatak 3

Napišite konzolnu aplikaciju koja preko konzole učitava vrijeme putovanja t u satima (realan broj) i ID tipa vozila (cijeli broj). ID tipa vozila dan je stupcem ID u tablici.

ID	Opis	Akceleracija [km/h ²]
3	Motor	72000
2	Automobil	36000
1	Kamion	14400

Početna brzina svakog vozila je $v_p = 10$ km/h. Ovisno o tome koji je ID vozila unesen potrebno je izračunati koliku će brzinu imati vozilo nakon što se vozi t vremena, pri čemu su akceleracije zadane u tablici 1, a konačna brzina v_k se računa koristeći izraz:

$$v_k = a * t + v_p$$

Također ukoliko je konačna brzina veća od 130 km/h potrebno je ispisati „Vozilo se vozi prebrzo!“. Format ispisa mora biti kao u primjeru.

Primjerice, ukoliko se učitava vrijeme od 0.0022 sata i vozilo pod ID-em 3, potrebno je ispisati „Motor će nakon 0.0022 sata vožnje imati brzinu 168.4 km/h! Vozilo se vozi prebrzo!“.



```

static void Main(string[] args)
{
    Console.Write("Unesite vrijeme putovanja t u satima: ");
    double t = Convert.ToDouble(Console.ReadLine());

    Console.Write("Unesite tip vozila: ");
    int tipVozila = Convert.ToInt32(Console.ReadLine());

    double vp = 10;
    double vk = 0.0;
    double a = 0.0;

    if (tipVozila == 1)
    {
        a = 14400;
        vk = a * t + vp;
        Console.Write("Kamion će nakon " + t + " sata imati brzinu " + vk
            + " km/h!");
    }
    else if (tipVozila == 2)
    {
        a = 36000;
        vk = a * t + vp;
        Console.Write("Automobil će nakon " + t + " sata imati brzinu " + vk
            + " km/h!");
    }
    else if (tipVozila == 3)
    {
        a = 72000;
        vk = a * t + vp;
        Console.Write("Motor će nakon " + t + " sata imati brzinu " + vk
            + " km/h!");
    }
    else
    {
        Console.Write("Nepoznat tip vozila!");
    }

    if (vk > 130)
    {
        Console.WriteLine("Vozilo se vozi prebrzo!");
    }

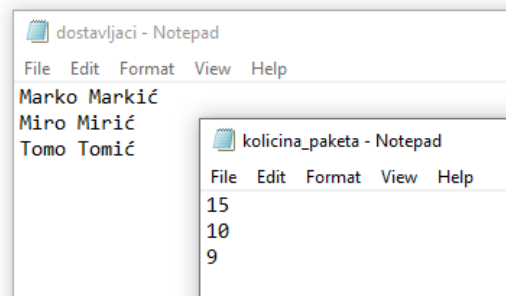
    Console.ReadKey();
}

```



Zadatak 4

Napišite konzolnu aplikaciju koja učitava sadržaj dviju datoteka dostavljac.txt i kolicina_paketa.txt. U datoteci dostavljac.txt upisana su imena i prezimena dostavljača, a u istom retku u datoteci kolicina_paketa.txt upisana je količina paketa koju je odgovarajući dostavljač dostavio.



Potrebno je u konzolu ispisati ime i prezime dostavljača te količinu paketa koju je dostavljač dostavio. Format ispisa mora biti kao u primjeru.

Napomena: potrebno je koristiti try-catch blok, te nije potrebno dodati imenski prostor System.IO. Za primjer sa slike potrebno je ispisati:

„Marko Marić je dostavio 15 paketa!

Miro Mirić je dostavio 10 paketa!

Tomo Tomić je dostavio 9 paketa!“

using System.IO;

```
static void Main(string[] args)
{
    try
    {
        StreamReader dostavljac = new StreamReader("dostavljac.txt");
        StreamReader paketi = new StreamReader("kolicina_paketa.txt");

        while (!dostavljac.EndOfStream)
        {
            string imeDostavljacka = dostavljac.ReadLine();
            int brojpaketa = Convert.ToInt32(paketi.ReadLine());
            Console.WriteLine(imeDostavljacka + " je dostavio "
                + brojpaketa + " paketa!");
        }

        dostavljac.Close();
        paketi.Close();
    }
    catch (Exception ex)
    {
        Console.WriteLine("Greška: " + ex.Message);
    }

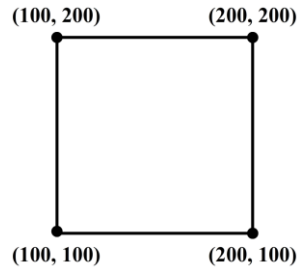
    Console.ReadKey();
}
```



Zadatak 5

Napišite konzolnu aplikaciju za simulaciju odredišnih parova.

Početna točka je ishodište $T_p(x_p, y_p) = (0, 0)$. Potrebno je dohvatiti 100 slučajnih točaka $T_k(x_k, y_k)$ (odredišta) unutar kvadrata prikazanog slikom.



Smatra se da su koordinate x i y cijeli brojevi, te da su granice kvadrata uključujuće.

Euklidska udaljenost između početne točke i pojedinog odredišta računa se izrazom.

$$D(T_p, T_k) = \sqrt{(x_k - x_p)^2 + (y_k - y_p)^2}$$

U svakom koraku u datoteku udaljenosti.txt potrebno je ispisivati u proizvoljnom formatu udaljenost između polazišta i trenutno dohvaćenog odredišta.

```
using System.IO;

static void Main(string[] args)
{
    Random rnd = new Random();

    int xp = 0;
    int yp = 0;

    StreamWriter writer = new StreamWriter("udaljenosti.txt");

    for (int i = 1; i <= 100; i++)
    {
        int xk = rnd.Next(100, 201);
        int yk = rnd.Next(100, 201);

        double D = Math.Sqrt(Math.Pow((xk - xp), 2) + Math.Pow((yk - yp), 2));

        writer.WriteLine("Udaljenosti između početne točke T(" + xp + ", "
            + yp + ") i simulirane točke T(" + xk + ", "
            + yk + ") je " + D + "!");
    }

    writer.Flush();
    writer.Close();

    Console.ReadKey();
}
```



Ispit #31 (02.07.19. - grupa A)

Zadatak 1

Potrebno je napisati metodu Generiraj koja kao argument prima dvije vrijednosti: cijeli broj x i bool vrijednost flag.

Metoda vraća listu cijelih brojeva. Ako je primljena vrijednost varijable flag istina (true) metoda vraća listu veličine x popunjenu brojevima 1, a ako je vrijednost laž (false) metoda vraća listu veličine x popunjenu brojevima 0.

```
static List<int> Generiraj(int x, bool flag)
{
    List<int> lista = new List<int>();

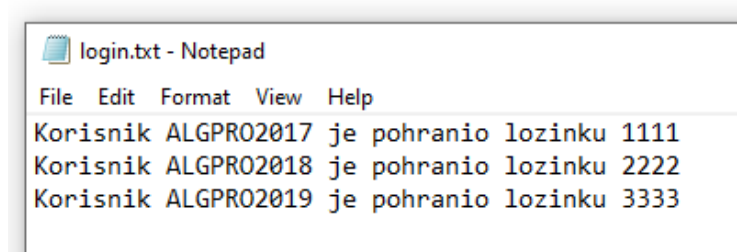
    for (int i = 1; i <= x; i++)
    {
        if (flag)
        {
            lista.Add(1);
        }
        else
        {
            lista.Add(0);
        }
    }

    return lista;
}
```



Zadatak 2

Napišite konzolnu aplikaciju koja omogućuje korisniku zapisivanje 3 proizvoljna korisnička imena i lozinke u datoteku. Format zapisa korisničkog imena i lozinke u datoteci prikazan je primjerom na slici.



Napomena: potrebno je koristiti try-catch blok, te nije potrebno dodati imenski prostor System.IO.

```
using System.IO;
static void Main(string[] args)
{
    try
    {
        StreamWriter writer = new StreamWriter("login.txt");

        for (int i = 1; i <= 3; i++)
        {
            Console.Write("Unesite korisničko ime: ");
            string korisnickoIme = Console.ReadLine();

            Console.Write("Unesite lozinku za korisnika "
                + korisnickoIme + ": ");

            string lozinka = Console.ReadLine();

            writer.WriteLine("Korisnik " + korisnickoIme
                + " je pohranio lozinku " + lozinka);
        }

        writer.Flush();
        writer.Close();
    }
    catch (Exception ex)
    {
        Console.WriteLine("Greška: " + ex.Message);
    }

    Console.ReadKey();
}
```



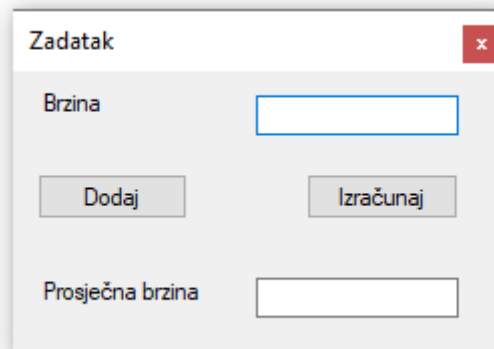
Zadatak 3

Nadopunite sljedeći kod unutar odgovarajućih metoda na način da se klikom na gumb Dodaj upisuje u listu vrijednost brzine automobila upisana u TextBox txtBrzina.

Klikom na gumb Izračunaj izračunava se i ispisuje prosječna vrijednost svih brzina u listi u TextBox txtProsječnaBrzina.

Prosječna vrijednost mora biti zaokružena na dva decimalna mjesta.

Sučelje je prikazano slikom.



```
//Nadopuni
List<int> brzine = new List<int>();

private void BtnDodaj_Click(object sender, EventArgs e)
{
    brzine.Add(Convert.ToInt32(txtBrzina.Text));
}

private void BtnIzracunaj_Click(object sender, EventArgs e)
{
    txtProsječnaBrzina.Text = Convert.ToString(Math.Round(
        brzine.Average(), 2));
}
```



Zadatak 4

Napišite program koji učitava ocjene za pet studenata te računa prosjek ocjena i ispisuje odgovarajuću poruku.

Nije potrebno vršiti provjeru da li je učitana dobra ocjena, očekuje se od korisnika da unese ispravnu ocjenu.

Ako je prosjek ocjena u intervalu $[0 - 1.5>$ potrebno je ispisati "Nedovoljan (1)!".

Ako je prosjek ocjena u intervalu $[1.5 - 2.5>$ potrebno je ispisati "Dovoljan (2)!".

Ako je prosjek ocjena u intervalu $[2.5 - 3.5>$ potrebno je ispisati "Dobar (3)!".

Ako je prosjek ocjena u intervalu $[3.5 - 4.5>$ potrebno je ispisati "Vrlo dobar (4)!".

Inače je potrebno ispisati "Odličan (5)!".

```
static void Main(string[] args)
{
    int suma = 0;

    for (int i = 1; i <= 5; i++)
    {
        Console.Write("Unesite ocjenu " + i + ". studenta: ");
        int ocjena = Convert.ToInt32(Console.ReadLine());

        suma = suma + ocjena;
    }

    double prosjek = suma / 5.0;

    if (prosjek >= 0 && prosjek < 1.5)
    {
        Console.WriteLine("Nedovoljan (1)!");
    }
    else if (prosjek >= 1.5 && prosjek < 2.5)
    {
        Console.WriteLine("Dovoljan (2)!");
    }
    else if (prosjek >= 2.5 && prosjek < 3.5)
    {
        Console.WriteLine("Dobar (3)!");
    }
    else if (prosjek >= 3.5 && prosjek < 4.5)
    {
        Console.WriteLine("Vrlo dobar (4)!");
    }
    else
    {
        Console.WriteLine("Odličan (5)!");
    }

    Console.ReadKey();
}
```



Zadatak 5

Napišite konzolnu aplikaciju koja učitava prirodni broj n sa konzole te potom kreira matricu $M = n \times n$.

Matricu je potrebno popuniti slučajnim prirodnim brojevima u intervalu $[1, 30]$.

Zatim je potrebno u jednodimenzionalno polje veličine n zapisati prosječne vrijednosti elemenata svakog retka matrice.

```
static void Main(string[] args)
{
    Random rnd = new Random();

    Console.Write("Unesite n: ");
    int n = Convert.ToInt32(Console.ReadLine());

    int[,] M = new int[n, n];
    double[] polje = new double[n];

    int suma = 0;

    for (int i = 0; i < M.GetLength(0); i++)
    {
        for (int j = 0; j < M.GetLength(1); j++)
        {
            M[i, j] = rnd.Next(1, 31);
            suma = suma + M[i, j];
        }
        polje[i] = suma / Convert.ToDouble(n);
        suma = 0;
    }

    Console.ReadKey();
}
```



Ispit #32 (02.07.19. - grupa B)

Zadatak 1

Napišite metodu Ispitaj koja preko argumenata prima listu cijelih brojeva, a vraća broj negativnih elemenata liste.

Ako nisu pronađeni negativni elementi u listi, metoda ispisuje na konzolu „Nisu pronađeni negativni elementi“ te vraća broj 0.

```
static int Ispitaj(List<int> lista)
{
    int brojNegativnih = 0;

    for (int i = 0; i < lista.Count; i++)
    {
        if (lista[i] < 0)
        {
            brojNegativnih++;
        }
    }

    if (brojNegativnih == 0)
    {
        Console.WriteLine("Nisu pronađeni negativni elementi");
    }

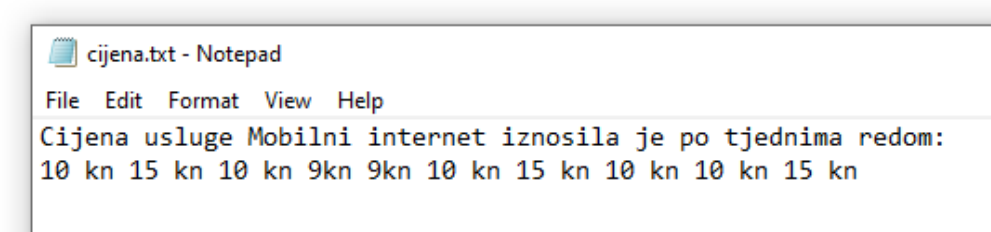
    return brojNegativnih;
}
```



Zadatak 2

Napišite konzolnu aplikaciju koja omogućuje korisniku pohranu proizvoljnog naziva usluge i 10 cijena te usluge po tjednima u datoteku.

Primjer formata zapisa naziva usluge (u primjeru Mobilni internet) i cijena u datoteci prikazan je slikom.



Napomena: potrebno je koristiti try-catch blok, te nije potrebno dodati imenski prostor System.IO.

```
static void Main(string[] args)
{
    try
    {
        StreamWriter writer = new StreamWriter("cijena.txt");

        Console.Write("Unesite naziv usluge: ");
        string usluga = Console.ReadLine();

        writer.WriteLine("Cijena usluge " + usluga
            + " iznosila je tjednima redom: ");

        for (int i = 1; i <= 10; i++)
        {
            Console.Write("Unesite cijenu usluge \"" + usluga + "\" za " + i
                + ". tjedan: ");

            int cijena = Convert.ToInt32(Console.ReadLine());

            writer.Write(cijena + " kn ");
        }

        writer.Flush();
        writer.Close();
    }
    catch (Exception ex)
    {
        Console.WriteLine("Greška: " + ex.Message);
    }

    Console.ReadKey();
}
```



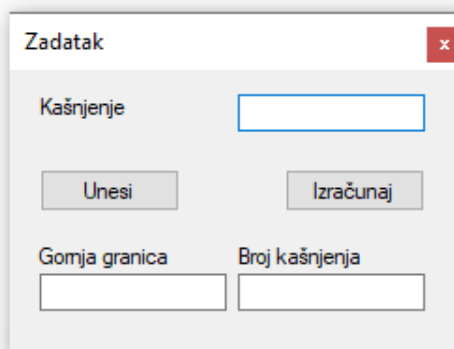
Zadatak 3

Nadopunite sljedeći kod unutar odgovarajućih metoda na način da se klikom na gumb Unesi dodaje u listu vrijednost kašnjenja zrakoplova upisana u TextBox txtKasnjenje.

Klikom na gumb Izračunaj prebrojavaju se sva kašnjenja u listi koja su veća od vrijednosti gornje granice upisane u TextBox txtGornjaGranica.

Broj kašnjenja ispisuje se u TextBox txtbrKasnjenja.

Sučelje je prikazano slikom.



//Nadopuni

```
List<int> kasnjenje = new List<int>();

private void BtnUnesi_Click(object sender, EventArgs e)
{
    kasnjenje.Add(Convert.ToInt32(txtKasnjenje.Text));
}

private void BtnIzracunaj_Click(object sender, EventArgs e)
{
    int gornjaGranica = Convert.ToInt32(txtGornjaGranica.Text);
    int brojKasnjenja = 0;

    for (int i = 0; i < kasnjenje.Count; i++)
    {
        if (kasnjenje[i] > gornjaGranica)
        {
            brojKasnjenja++;
        }
    }

    txtbrKasnjenja.Text = Convert.ToString(brojKasnjenja);
}
```



Zadatak 4

Napišite program koji učitava broj bodova unutar intervala [0, 5]. Ako učitani bodovi nisu unutar zadanog intervala program ispisuje "Niste unijeli bodove u intervalu [0, 5]!".

Ako su ispravno učitani bodovi potrebno je ispisati odgovarajuću ocjenu.

Ako je broj bodova u intervalu [0 – 1.5> potrebno je ispisati "Nedovoljan (1)!".

Ako je broj bodova u intervalu [1.5 – 2.5> potrebno je ispisati "Dovoljan (2)!".

Ako je broj bodova u intervalu [2.5 – 3.5> potrebno je ispisati "Dobar (3)!".

Ako je broj bodova u intervalu [3.5 – 4.5> potrebno je ispisati "Vrlo dobar (4)!".

Inače je potrebno ispisati "Odličan (5)!".

```
static void Main(string[] args)
{
    Console.WriteLine("Unesite broj bodova unutar intervala [0, 5]: ");
    double brojBodova = Convert.ToDouble(Console.ReadLine());

    if (brojBodova < 0 || brojBodova > 5)
    {
        Console.WriteLine("Niste unijeli bodove u intervalu [0, 5]!");
    }
    else if (brojBodova >= 0 && brojBodova < 1.5)
    {
        Console.WriteLine("Nedovoljan (1)!");
    }
    else if (brojBodova >= 1.5 && brojBodova < 2.5)
    {
        Console.WriteLine("Dovoljan (2)!");
    }
    else if (brojBodova >= 2.5 && brojBodova < 3.5)
    {
        Console.WriteLine("Dobar (3)!");
    }
    else if (brojBodova >= 3.5 && brojBodova < 4.5)
    {
        Console.WriteLine("Vrlo dobar (4)!");
    }
    else
    {
        Console.WriteLine("Odličan (5)!");
    }

    Console.ReadKey();
}
```



Zadatak 5

Napišite konzolnu aplikaciju koja učitava prirodni broj n s konzole te potom kreira matricu $M = (n - 1) \times (n + 1)$.

Matricu je potrebno popuniti slučajnim prirodnim brojevima u intervalu [30, 59].

Zatim je potrebno u jednodimenzionalno polje upisati n slučajno odabranih elemenata iz matrice M .

```
static void Main(string[] args)
{
    Console.WriteLine("Unesite veličinu kvadratne matrice: ");
    int n = Convert.ToInt32(Console.ReadLine());

    int[,] M = new int[n - 1, n + 1];

    Random rnd = new Random();

    for (int i = 0; i < M.GetLength(0); i++)
    {
        for (int j = 0; j < M.GetLength(1); j++)
        {
            M[i, j] = rnd.Next(30, 60);
        }
    }

    int[] polje = new int[n];

    for (int i = 0; i < polje.Length; i++)
    {
        int indeksRetka = rnd.Next(0, M.GetLength(0) - 1);
        int indeksStupca = rnd.Next(0, M.GetLength(1) - 1);
        polje[i] = M[indeksRetka, indeksStupca];
    }

    Console.ReadKey();
}
```



Ispit #33 (11.07.19. - grupa A)

Zadatak 1

Napišite kod klase Grad koja sadrži privatne atribute naziv, broj stanovnika (cijeli broj) i javne atribute x i y (realni brojevi) koji predstavljaju geografsku dužinu, odnosno širinu.

Konstruktor klase prima argumente za sve atribute klase.

Potrebno je napisati i javnu metodu PovecajBrStanovnika koja povećava vrijednost atributa broja stanovnika za vrijednost predanu kao argument metode i ispisuje poruku: „Grad naziv_grada ima novi broj stanovnika broj_stanovnika!“.

```
class Grad
{
    private string naziv;
    private int brojStanovnika;
    public double x;
    public double y;

    public Grad(string NAZIV, int BROJSTANOVNIKA, double X, double Y)
    {
        naziv = NAZIV;
        brojStanovnika = BROJSTANOVNIKA;
        x = X;
        y = Y;
    }

    public void PovecajBrStanovnika(int brojStanovnikaZaDodavanje)
    {
        brojStanovnika += brojStanovnikaZaDodavanje;

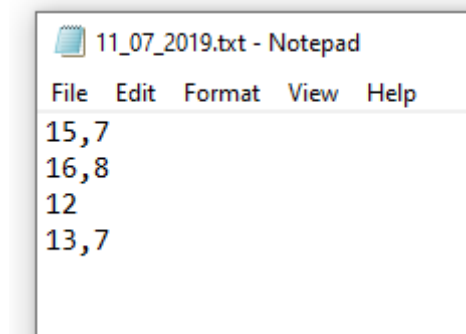
        Console.WriteLine("Grad " + naziv + " ima novi broj stanovnika "
            + brojStanovnika + "!");
    }
}
```



Zadatak 2

Napišite metodu Ucitaj koja kao argument prima niz znakova (riječ) koji predstavljaju naziv jedne datoteke.

Sve datoteke koje se obrađuju neovisno o nazivu sadrže u zasebnim recima količine paketa u kilogramima (realni brojevi) koje je taj dan potrebno dostaviti kao što je to prikazano na slici, a naziv datoteke dogovara datumu za koji se vrši dostava npr. 11_07_2019.txt.



Metoda treba vratiti ukupnu količinu paketa koju je potrebno dostaviti za predani naziv datoteke koja se učitava.

Napomena: nije potrebno koristiti try-catch blok, te nije potrebno dodati imenski prostor System.IO.

```
static double Ucitaj(string naziv)
{
    StreamReader reader = new StreamReader(naziv);

    double ukupnoPaketa = 0.0;

    while (!reader.EndOfStream)
    {
        ukupnoPaketa += Convert.ToDouble(reader.ReadLine());
    }

    reader.Close();

    return ukupnoPaketa;
}
```

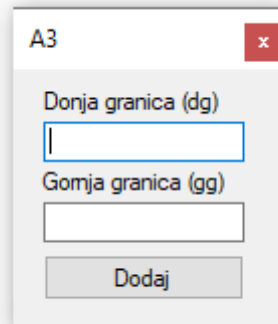


Zadatak 3

Nadopunite sljedeći kod unutar odgovarajuće metode na način da se klikom na gumb Dodaj u listu vrijednost brzine automobila dodaje nova brzina čija se vrijednost određuje slučajno unutar intervala [dd, gg>.

Donja i gornja granica upisuju se preko TextBoxova naziva txtDonja i txtGornja.

Sučelje je prikazano slikom.



```
//Nadopuni
List<int> brzine = new List<int>();
Random rnd = new Random();

private void BtnDodaj_Click(object sender, EventArgs e)
{
    int dg = Convert.ToInt32(txtDonja.Text);
    int gg = Convert.ToInt32(txtGornja.Text);

    int trenutnaBrzina = rnd.Next(dg, gg);

    brzine.Add(trenutnaBrzina);
}
```

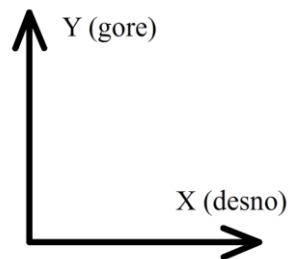


Zadatak 4

Napišite program koji simulira kretanje zmije kao u igrici Snake.

Program treba učitavati niz znakova s konzole koji predstavljaju kretanje zmije, sve dok ne učitava nevaljani niz znakova. Nevaljani niz znakova je svaki niz koji nije iz skupa niza znakova {gore, dolje, lijevo, desno}.

Početna pozicija zmije je određena koordinatom $T_0(x, y) = (0, 0)$ pri čemu su x i y cijeli brojevi. Koordinatni sustav prikazan je na slici.



Nakon učitavanja valjanog znakovnog niza potrebno je pomaknuti koordinatu zmije za vrijednost 5 u smjeru određenom učitanim znakovnim nizom.

Na kraju je potrebno ispisati konačnu poziciju (koordinatu) zmije u formatu kao u primjeru.

Primjer: ako se redom učitaju vrijednosti gore, desno, gore, desno, dolje, lijevo, kraj potrebno je ispisati: „Zmija se na kraju nalazi na poziciji (5,5)!“.



```

int x = 0;
int y = 0;
int pomak = 5;
bool ispravanUnos = true;

do
{
    Console.WriteLine("Unesite smjer gibanja zmije: ");
    string smjer = Console.ReadLine();

    switch (smjer)
    {
        case "gore":
            y += pomak;
            break;
        case "dolje":
            y -= pomak;
            break;
        case "desno":
            x += pomak;
            break;
        case "lijevo":
            x -= pomak;
            break;
        default:
            ispravanUnos = false;
            break;
    }
} while (ispravanUnos);

Console.WriteLine("Zmija se na kraju nalazi na poziciji ("
+ x + ", " + y + ")!");

Console.ReadKey();

```



Zadatak 5

Napišite konzolnu aplikaciju koja učitava prirodni broj n s konzole te potom kreira polje veličine n u koje će se spremati studenti na kolegiju Algoritmi i programiranje (klasa Student).

Potrebno je popuniti polje s instancama klase Student čiji atributi ime i bodovi imaju vrijednosti koje se unose preko konzole.

```
public class Student
{
    private string ime;
    public double bodovi;

    public Student(string IME, double BODOVI)
    {
        ime = IME;
        bodovi = BODOVI;
    }
}

class Program
{
    static void Main(string[] args)
    {
        //Nadopuni
        Console.Write("Unesite broj studenata: ");
        int n = Convert.ToInt32(Console.ReadLine());

        Student[] studenti = new Student[n];

        for (int i = 0; i < studenti.Length; i++)
        {
            Console.Write("Unesite ime " + (i + 1) + ". studenta: ");
            string ime = Console.ReadLine();

            Console.Write("Unesite bodove " + (i + 1) + ". studenta: ");
            double bodovi = Convert.ToDouble(Console.ReadLine());

            studenti[i] = new Student(ime, bodovi);
        }

        Console.ReadKey();
    }
}
```



Ispit #34 (11.07.19. - grupa B)

Zadatak 1

Napišite kod klase Dostava koja sadrži javne atribute naziv, x i y (realni brojevi) koji predstavljaju geografsku dužinu, odnosno širinu te privatni atribut obavljenaDostava (binarna vrijednost istina ili laž).

Konstruktor klase prima argumente naziva, geografske dužine x i geografske širine y, a atribut obavljenaDostava postavlja na neistinu, jer još nije obavljena dostava.

Potrebno je napisati i javnu metodu ObaviDostavu koja postavlja atribut obavljene dostave na istinu i ispisuje poruku: „Dostava za naziv_dostave je obavljena!“.

```
class Dostava
{
    public string naziv;
    public double x;
    public double y;
    private bool obavljenaDostava;

    public Dostava(string NAZIV, double X, double Y)
    {
        naziv = NAZIV;
        x = X;
        y = Y;
        obavljenaDostava = false;
    }

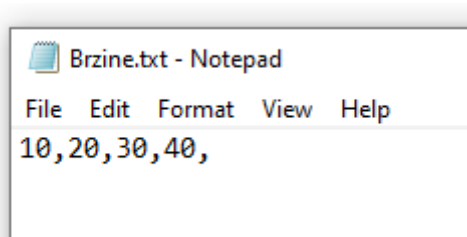
    public void ObaviDostavu()
    {
        obavljenaDostava = true;
        Console.WriteLine("Dostava za " + naziv + " je obavljena!");
    }
}
```



Zadatak 2

Napišite metodu `Ispis` koja kao argument prima niz znakova `s` i broj simulacija brzina `n` (cijeli broj). Metoda ne vraća rezultat.

Niz znakova `s` predstavlja naziv datoteke u koju će se spremiti vrijednosti, npr. `Brzine.txt`. Potrebno je iz konzole učitati `n` vrijednosti brzina (realni brojevi) i zapisivati ih u datoteku sve skupa u jednom retku, odvojene zarezom (kao u primjeru na slici).



Napomena: potrebno je koristiti try-catch blok, te nije potrebno dodati imenski prostor `System.IO`.

```
static void Ispis(string s, int n)
{
    try
    {
        StreamWriter writer = new StreamWriter(s);

        for (int i = 1; i <= n; i++)
        {
            Console.Write("Unesite " + i + ". brzinu: ");
            double brzina = Convert.ToDouble(Console.ReadLine());

            writer.Write(brzina + ",");
        }

        writer.Flush();
        writer.Close();
    }
    catch (Exception ex)
    {
        Console.WriteLine("Greška: " + ex.Message);
    }
}
```



Zadatak 3

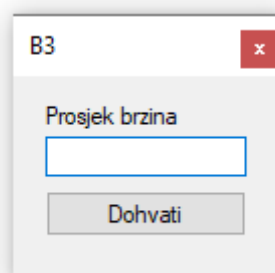
Nadopunite sljedeći kod unutar odgovarajućih metoda na način da se klikom na gumb Dohvati dohвате 2 (dvije) slučajne brzine zapisane u listi brzine te se prosjek vrijednosti dohvaćenih brzina zapiše u TextBox txtProsjeKBrzina.

```
//Nadopuni
List<int> brzine = new List<int>();

private void BtnDohvati_Click(object sender, EventArgs e)
{

}
```

Pretpostavka je da je u listi već zapisano 100 brzina. Sučelje je prikazano slikom.



Nije potrebno paziti da budu dohvaćene različite vrijednosti brzina iz liste.

```
//Nadopuni
List<int> brzine = new List<int>();
Random rnd = new Random();
private void BtnDohvati_Click(object sender, EventArgs e)
{
    int slucajniIndeks1 = rnd.Next(0, brzine.Count);
    int slucajniIndeks2 = rnd.Next(0, brzine.Count);

    double prosjek = (brzine[slucajniIndeks1] + brzine[slucajniIndeks2])/2.0;

    txtProsjeKBrzina.Text = Convert.ToString(prosjek);
}
```



Zadatak 4

Napišite program koji simulira vožnju automobila po trkaćoj stazi. Program treba učitavati niz znakova s konzole koji predstavljaju kretanje automobila, sve dok automobil ne prođe udaljenost od 300 m koliko iznosi staza.

Pretpostavka je da korisnik učitava novi znakovni niz s konzole svakih $t = 5$ sekundi te da se neće učitati krivi znakovni niz.

Moguće je učitati niz znakova iz skupa ubrzaj, uspori, koji odgovaraju ubrzanju odnosno usporenju automobila. Svaki put kada se učitava novi znakovni niz potrebno je izračunati brzinu i udaljenost, ako je učitani znakovni niz uspori primjenjuju se izrazi koji se odnose na usporenje, a ako je učitani znakovni niz ubrzaj primjenjuju se izrazi koji se odnose na ubrzanje.

Prijeđeni put se računa prema izrazu:

$$s_k = \begin{cases} s_p + v_p t + \frac{a}{2} t^2, & \text{ubrzanje} \\ s_p + v_p t - \frac{a}{2} t^2, & \text{usporenje} \end{cases}$$

Brzina se računa prema izrazu:

$$v_k = \begin{cases} v_p + at, & \text{ubrzanje} \\ v_p - at, & \text{usporenje} \end{cases}$$

Početna brzina, akceleracija i udaljenost su postavljeni na vrijednost 0 (realni brojevi). v_p i s_p su početna brzina i prijeđena udaljenost na početku intervala, a v_k i s_k su krajnja brzina i prijeđena udaljenost na kraju intervala.

Na kraju je potrebno ispisati vrijeme prolaska staze u formatu kao u primjeru.

Vrijeme se računa prema vremenu učitavanja, odnosno ako je primjerice već nakon prvog učitavanja prijeđena udaljenost veća od 300 m, vrijeme prolaska staze iznosi 5 sekundi (jer je "prošlo" 5 sekundi). Akceleracija za ubrzanje i usporenje iznosi $a = 2.78 \text{ m/s}^2$.

Primjer: ako se redom učitaju vrijednosti ubrzaj, ubrzaj, uspori, ubrzaj potrebno je ispisati: „Vrijeme prolaska staze iznosi 20 sekundi!“ (opaska: jer je prošlo 4 učitavanja).



```

double vp = 0.0, vk = 0.0, sp = 0.0, sk = 0.0;
double a = 2.78;
int t = 5;
int ukupnoVrijeme = 0;
bool stazaJePrijedjena = false;

do
{
    Console.WriteLine("Unesite kretanje vozila (ubrzaj/uspori): ");
    string kretanje = Console.ReadLine();

    if (kretanje == "ubrzaj")
    {
        vk = vp + a * t;
        sk = sp + (vp * t) + (a / 2 * Math.Pow(t, 2));
    }
    else if (kretanje == "uspori")
    {
        vk = vp - a * t;
        sk = sp + (vp * t) - (a / 2 * Math.Pow(t, 2));
    }

    ukupnoVrijeme += 5;
    vp = vk;
    sp = sk;

    if (sk > 300)
    {
        stazaJePrijedjena = true;
    }
} while (!stazaJePrijedjena);

Console.WriteLine("Vrijeme prolaska staze iznosi " + ukupnoVrijeme
+ " sekundi!");

Console.ReadKey();

```



Zadatak 5

Napišite konzolnu aplikaciju koja učitava prirodni broj n s konzole te potom kreira polje veličine n u koje će se spremati točke (klasa Točka).

Potrebno je popuniti polje s instancama klase Točka čiji atributi x i y imaju vrijednosti koje se unose preko konzole.

```
public class Točka
{
    public int x;
    public int y;

    public Točka(int X, int Y)
    {
        x = X;
        y = Y;
    }
}

class Program
{
    static void Main(string[] args)
    {
        //Nadopuni
        Console.Write("Unesite željeni broj točaka n: ");
        int n = Convert.ToInt32(Console.ReadLine());

        Točka[] tocke = new Točka[n];

        for (int i = 0; i < tocke.Length; i++)
        {
            Console.Write("Unesite x koordinantu " + (i + 1) + ". točke: ");
            int x = Convert.ToInt32(Console.ReadLine());

            Console.Write("Unesite y koordinantu " + (i + 1) + ". točke: ");
            int y = Convert.ToInt32(Console.ReadLine());

            tocke[i] = new Točka(x, y);
        }

        Console.ReadKey();
    }
}
```



Ispit #35 (29.08.19. - grupa A)

Zadatak 1

Nadopunite sljedeći kod unutar Main metode na način da stvorite novi auto naziva Insignia, kojem je vlasnik Marko, te koji se trenutno nalazi na GPS lokaciji $x = 15.9773$, $y = 45.8129$.

Potom u konzolu ispišite vrijednost x GPS lokacije vozila i naziv automobila (vrijednosti atributa unutar instance).

```
static void Main(string[] args)
{
    //Nadopuni

    GPSLokacija lokacija = new GPSLokacija(45.8129, 15.9773);
    Auto auto = new Auto(lokacija, "Marko", "Insignia");

    Console.WriteLine(auto.lokacija.x);
    Console.WriteLine(auto.imeAuto);

    Console.ReadKey();
}

public class GPSLokacija
{
    public double x;
    public double y;

    public GPSLokacija(double Y, double X)
    {
        x = X;
        y = Y;
    }
}

public class Auto
{
    public string imeAuto;
    public string imeVlasnika;
    public GPSLokacija lokacija;

    public Auto(GPSLokacija LOKACIJA, string IMEVLASNIKA, string IMEAUTA)
    {
        imeAuto = IMEAUTA;
        lokacija = LOKACIJA;
        imeVlasnika = IMEVLASNIKA;
    }
}
```



Zadatak 2

Napišite konzolnu aplikaciju koja učitava sadržaj datoteke `potrosnjaInterneta.txt` kojoj se pristupa preko relativne putanje.

U datoteci je zapisano 12 realnih brojeva svaki u zasebnom retku koji predstavljaju izmjerenu mjesečnu potrošnju mobilnog interneta u GB.

Potrebno je ispisati prosječnu mjesečnu potrošnju korisnika u sljedećem formatu: „Prosječna mjesečna potrošnja korisnika je x GB!“.

Napomena: potrebno je koristiti try-catch blok, te nije potrebno dodati imenski prostor `System.IO`.

```
try
{
    StreamReader reader = new StreamReader("potrosnjaInterneta.txt");

    double sumaPotrosnje = 0.0;

    while (!reader.EndOfStream)
    {
        double mjesečnaPotrosnja = Convert.ToDouble(reader.ReadLine());
        sumaPotrosnje += mjesečnaPotrosnja;
    }

    reader.Close();

    double prosjek = sumaPotrosnje / 12;

    Console.WriteLine("Prosječna mjesečna potrošnja korisnika je "
        + prosjek + " GB!");
}
catch (Exception ex)
{
    Console.WriteLine("Greška: " + ex);
}

Console.ReadKey();
```



Zadatak 3

Napišite program koji učitava tri realna broja s tipkovnice a, b i c koji redom predstavljaju stranice kvadra. Potrebno je napisati metodu IzracunajDijagonalu koja kao argumente prima stranice kvadra, a vraća izračunatu duljinu dijagonale kvadra.

U glavnom dijelu programa potrebno je ispisati duljinu dijagonale samo ako su sve stranice kvadra veće od 0. Format ispisa mora biti kao u primjeru.

Primjer: ako se redom učitaju vrijednosti 2.5, 3 i 4 potrebno je ispisati: „Duljina dijagonale kvadra određenog stranicama a = 2.5, b = 3 i c = 4 je 5.59!“.

Izraz za izračun dijagonale kvadra dan je izrazom:

$$D = \sqrt{a^2 + b^2 + c^2}$$

```
static void Main(string[] args)
{
    Console.Write("Unesite stranicu kvadra a: ");
    double a = Convert.ToDouble(Console.ReadLine());

    Console.Write("Unesite stranicu kvadra b: ");
    double b = Convert.ToDouble(Console.ReadLine());

    Console.Write("Unesite stranicu kvadra c: ");
    double c = Convert.ToDouble(Console.ReadLine());

    if (a > 0 && b > 0 && c > 0)
    {
        double D = IzracunajDijagonalu(a, b, c);

        Console.WriteLine("Duljina dijagonale kvadra određenog stranicama a = "
            + a + ", b = " + b + " i c = " + c + " je " + D + "!");
    }

    Console.ReadKey();
}

static double IzracunajDijagonalu(double a, double b, double c)
{
    double D = Math.Sqrt(Math.Pow(a, 2) + Math.Pow(b, 2) + Math.Pow(c, 2));

    return D;
}
```



Zadatak 4

Napišite program koji učitava podatke o nazivima smjerova na fakultetu i broju predmeta na smjeru u dva polja.

U prvo polje se upisuju nazivi smjerova (niz znakova), a u drugo polje se upisuju cijeli brojevi koji predstavljaju broj predmeta na odgovarajućem smjeru. Na Fakultetu postoji 10 različitih smjerova.

Na kraju učitavanja u proizvoljnom formatu potrebno je ispisati naziv smjera koji ima najviše predmeta.

```
static void Main(string[] args)
{
    string[] smjerovi = new string[10];
    int[] brojPredmeta = new int[10];

    int najvećiBrojPredmeta = int.MinValue;
    string smjerSNajvecimBrojemPredmeta = "";

    for (int i = 0; i < smjerovi.Length; i++)
    {
        Console.Write("Unesite naziv " + (i + 1) + ". smjera: ");
        smjerovi[i] = Console.ReadLine();

        Console.Write("Unesite broj predmeta na smjeru " + smjerovi[i]
            + ": ");
        brojPredmeta[i] = Convert.ToInt32(Console.ReadLine());

        if (brojPredmeta[i] > najvećiBrojPredmeta)
        {
            najvećiBrojPredmeta = brojPredmeta[i];
            smjerSNajvecimBrojemPredmeta = smjerovi[i];
        }
    }

    Console.WriteLine("Smjer s najvećim brojem predmeta je "
        + smjerSNajvecimBrojemPredmeta + ".");

    Console.ReadKey();
}
```



Zadatak 5

Napišite konzolnu aplikaciju koja učitava 1000 cijelih slučajnih brojeva unutar intervala <0, 100>.

U dvije liste spremaju se brojevi ovisno o njihovoj vrijednosti. Ako je slučajan broj djeljiv i s 2 i s 3 te nije djeljiv s 5 dodaje ga se u jednu listu, inače ga se dodaje u drugu listu.

Na kraju je potrebno ispisati u konzolu koliko ima elemenata u pojedinoj listi u proizvoljnom formatu.

```
static void Main(string[] args)
{
    Random rnd = new Random();

    List<int> lista1 = new List<int>();
    List<int> lista2 = new List<int>();

    for (int i = 1; i <= 1000; i++)
    {
        int x = rnd.Next(1, 100);

        if (x % 2 == 0 && x % 3 == 0 && x % 5 != 0)
        {
            lista1.Add(x);
        }
        else
        {
            lista2.Add(x);
        }
    }

    Console.WriteLine("Broj brojeva u prvoj listi: " + lista1.Count);
    Console.WriteLine("Broj brojeva u drugoj listi: " + lista2.Count);

    Console.ReadKey();
}
```



Ispit #36 (29.08.19. - grupa B)

Zadatak 1

Nadopunite sljedeći kod unutar Main metode na način da stvorite novi grad Zagreb s koordinatama $x = 15.9773$, $y = 45.8129$ i brojem stanovnika 800000.

Potom u konzolu ispišite x i y koordinatu grada (vrijednost atributa unutar instance).

```
static void Main(string[] args)
{
    //Nadopuni
    Tocka t = new Tocka(45.8129, 15.9773);
    Grad g = new Grad(800000, "Zagreb", t);

    Console.WriteLine(g.t.x);
    Console.WriteLine(g.t.y);

    Console.ReadKey();
}

public class Grad
{
    private string naziv;
    public Tocka t;
    private int brojStanovnika;

    public Grad(int BROJSTANOVNIKA, string NAZIV, Tocka T)
    {
        t = T;
        naziv = NAZIV;
        brojStanovnika = BROJSTANOVNIKA;
    }
}

public class Tocka
{
    public double x;
    public double y;

    public Tocka(double Y, double X)
    {
        x = X;
        y = Y;
    }
}
```



Zadatak 2

Napišite konzolnu aplikaciju koja učitava cjelobrojnu vrijednosti n s tipkovnice koja predstavlja broj redaka i stupaca kvadratne matrice $M = n \times n$.

Potrebno je popuniti matricu M s rezultatima množenja prirodnih brojeva počevši od vrijednosti 1, odnosno kao tablicu množenja.

Matricu je potrebno ispisati u formatu kao u primjeru. Primjerice, ako se za n učitava vrijednost 4, potrebno je ispisati matricu:

```
1,2,3,4
2,4,6,8
3,6,9,12
4,8,12,16
```

```
static void Main(string[] args)
{
    Console.WriteLine("Unesite n: ");
    int n = Convert.ToInt32(Console.ReadLine());

    int[,] M = new int[n, n];

    for (int i = 1; i <= M.GetLength(0); i++)
    {
        for (int j = 1; j <= M.GetLength(1); j++)
        {
            M[i - 1, j - 1] = i * j;

            Console.Write(M[i - 1, j - 1]);
            if (j < M.GetLength(1))
            {
                Console.Write(",");
            }
        }
        Console.WriteLine();
    }

    Console.ReadKey();
}
```



Zadatak 3

Napišite konzolnu aplikaciju koji učitava tri realna broja s tipkovnice a, b i c koji predstavljaju stranice trokuta.

Potrebno je provjeriti da li su sve unesene vrijednosti veće od nule, te ako jesu potrebno je izračunati i ispisati površinu trokuta prema formatu u primjeru koristeći Heronovu formulu danu jednadžbama 1 i 2, a ako nisu potrebno je ispisati „Unesene su krive vrijednosti stranica trokuta!“.

Provjeru da li su učitane ispravne vrijednosti potrebno je napisati u zasebnoj metodi Provjera koja prima tri argumenta (stranice trokuta) i vraća rezultat provjere.

Ako se primjerice redom unesu vrijednosti 3, 4 i 5 potrebno je ispisati: „Površina trokuta određenog stranicama a = 3, b = 4 i c = 5 iznosi: 6!“.

$$(1) P = \sqrt{s(s-a)(s-b)(s-c)}$$

$$(2) s = \frac{a+b+c}{2}$$

```
static void Main(string[] args)
{
    Console.Write("Unesite stranicu trokuta a: ");
    double a = Convert.ToDouble(Console.ReadLine());

    Console.Write("Unesite stranicu trokuta b: ");
    double b = Convert.ToDouble(Console.ReadLine());

    Console.Write("Unesite stranicu trokuta c: ");
    double c = Convert.ToDouble(Console.ReadLine());

    if (Provjera(a, b, c))
    {
        double s = (a + b + c) / 2;
        double P = Math.Sqrt(s * (s - a) * (s - b) * (s - c));

        Console.WriteLine("Površina trokuta omeđenog stranicama a = " + a
            + ", b = " + b + " i c = " + c + " iznosi: " + P + "!");
    }
    else
    {
        Console.WriteLine("Unesene su krive vrijednosti stranica trokuta!");
    }

    Console.ReadKey();
}

static bool Provjera(double a, double b, double c)
{
    if (a > 0 && b > 0 && c > 0)
    {
        return true;
    }
    return false;
}
```



Zadatak 4

Napišite konzolnu aplikaciju koja učitava 100 imena i prezimena korisnika (niz znakova) i pripadajuće mjesečne potrošnje interneta (realan broj).

U datoteku korisnici.txt potrebno je zapisivati imena i prezimena korisnika. Svaki korisnik u zasebnom retku. U odgovarajućem retku u datoteci potrosnja.txt potrebno je zapisivati odgovarajuću mjesečnu potrošnju interneta korisnika. Datotekama se pristupa preko relativne putanje.

Napomena: potrebno je koristiti try-catch blok, nije potrebno dodati imenski prostor System.IO te se ime i prezime može učitati kao jedan znakovni niz.

```
static void Main(string[] args)
{
    try
    {
        StreamWriter korisnici = new StreamWriter("korisnici.txt");
        StreamWriter potrosnja = new StreamWriter("potrosnja.txt");

        for (int i = 1; i <= 100; i++)
        {
            Console.Write("Unesite ime i prezime " + i + ". korisnika: ");
            string korisnik = Console.ReadLine();
            korisnici.WriteLine(korisnik);

            Console.Write("Unesite potrošnju za " + i + ". korisnika: ");
            double potrosnjaKorisnika = Convert.ToDouble(Console.ReadLine());
        }

        korisnici.Flush();
        korisnici.Close();

        potrosnja.Flush();
        potrosnja.Close();
    }
    catch (Exception ex)
    {
        Console.WriteLine("Greška: " + ex.Message);
    }

    Console.ReadKey();
}
```



Zadatak 5

Napišite konzolnu aplikaciju koja učitava 1000 cijelih slučajnih brojeva unutar intervala [0, 1000>.

U tri liste spremaju se brojevi ovisno o njihovoj vrijednosti. Ako je broj unutar intervala [0, 10> dodaje ga se u listu jednaZnamenka, ako je broj unutar intervala [10, 100>, dodaje ga se u listu dvijeZnamenke, inače ga se dodaje u listu triZnamenke.

Na kraju je potrebno ispisati koliko ima kojih brojeva, primjerice:

„Jednoznamenkastih brojeva ima 10!

Dvoznamenkastih brojeva ima 90!

Troznamenkastih brojeva ima 900!“.

```
static void Main(string[] args)
{
    Random rnd = new Random();

    List<int> jednaZnamenka = new List<int>();
    List<int> dvijeZnamenke = new List<int>();
    List<int> triZnamenke = new List<int>();

    for (int i = 1; i <= 1000; i++)
    {
        int x = rnd.Next(0, 1000);

        if (x >= 0 && x < 10)
        {
            jednaZnamenka.Add(x);
        }
        else if (x >= 10 && x < 100)
        {
            dvijeZnamenke.Add(x);
        }
        else
        {
            triZnamenke.Add(x);
        }
    }

    Console.WriteLine("Jednoznamenkastih brojeva ima "
        + jednaZnamenka.Count + "!");
    Console.WriteLine("Dvoznamenkastih brojeva ima "
        + dvijeZnamenke.Count + "!");
    Console.WriteLine("Troznamenkastih brojeva ima "
        + triZnamenke.Count + "!");

    Console.ReadKey();
}
```



Ispit #38 (20.02.20. - grupa A)

Zadatak 1

Što će se ispisati u konzolu nakon izvršavanja sljedećeg programskog odsječka? Potrebno je naznačiti redoslijed ispisivanja.

```
static void Main(string[] args)
{
    PravokutanTrokut p = new PravokutanTrokut(4, 6);
    p.izracunajPovrsinu();
    Console.WriteLine(p.izracunajPovrsinu(2, 3));
    p.ispis();

    Console.ReadKey();
}

class PravokutanTrokut
{
    private double A;
    private double B;
    private double povrsina;

    public PravokutanTrokut(double a, double b)
    {
        A = a;
        B = b;
        povrsina = 0;
        ispis();
    }

    public double izracunajPovrsinu(int a, int b)
    {
        double POVRSINA = (b * a) / 2;
        ispis();
        return POVRSINA;
    }

    public void izracunajPovrsinu()
    {
        povrsina = (A * B) / 2;
        ispis();
    }

    public void ispis()
    {
        Console.WriteLine(povrsina);
    }
}
```

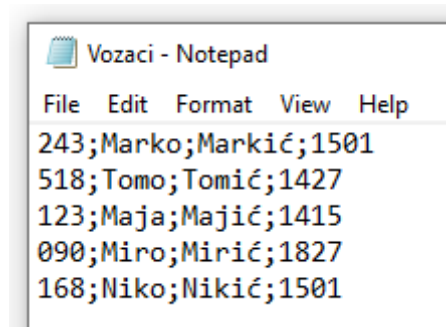
Redoslijed	Ispis
1	0
2	12
3	12
4	3
5	12



Zadatak 2

Napišite C# kod metode PronadiNajbrzeg koja kao argument prima cijelu putanju do datoteke koja sadrži zapise o vozačima utrke i njihovim vremenima (u sekundama).

U jednom retku datoteke zapisana je informacija o vozaču i njegovom vremenu u formatu: ID_vozac;Ime;Prezime;Vrijeme kao što je prikazano na slici.



Metoda kao rezultat vraća ID vozača (cijeli broj) koji je ostvario najbolje vrijeme na utrci, a u konzolu u proizvoljnom formatu ispisuje ime, prezime i vrijeme vozača. Potrebno je koristiti try-catch blok te nije potrebno dodati imenski prostor System.IO.

Prilikom učitavanja za dijeljenje podataka u retku potrebno je koristiti ugrađenu metodu Split nad tipom podatka string. Primjer poziva metode Split nad retkom "Ovo-je-primjer!", s crticom '-' kao znakom za dijeljenje retka je sljedeći:

```
string linija = "Ovo-je-primjer!";  
string[] dijeloviLinije = linija.Split('-');
```



```

static int PronadiNajbrzeg(string putanja)
{
    int IDNajbrzeg = -1;

    try
    {
        StreamReader reader = new StreamReader(putanja);

        int vrijemeNajbrzeg = int.MaxValue;
        string imeNajbrzeg = "";
        string prezimeNajbrzeg = "";

        while (!reader.EndOfStream)
        {
            string linija = reader.ReadLine();
            string[] dijeloviLinije = linija.Split(';');
            int trenutnoVrijeme = Convert.ToInt32(dijeloviLinije[3]);
            if (trenutnoVrijeme < vrijemeNajbrzeg)
            {
                vrijemeNajbrzeg = trenutnoVrijeme;
                IDNajbrzeg = Convert.ToInt32(dijeloviLinije[0]);
                imeNajbrzeg = dijeloviLinije[1];
                prezimeNajbrzeg = dijeloviLinije[2];
            }
        }

        reader.Close();

        Console.WriteLine("Vozač " + imeNajbrzeg + " " + prezimeNajbrzeg
            + " je s vremenom od " + vrijemeNajbrzeg
            + " [s] ostvario/la najbolje vrijeme na utrci.");
    }
    catch (Exception ex)
    {
        Console.WriteLine("Greška: " + ex.Message);
    }

    return IDNajbrzeg;
}

```



Zadatak 3

Napišite kod C# konzolne aplikacije koja predstavlja igru pogađanja cijelog broja. Prvo je potrebno dohvatiti slučajan cijeli broj x unutar intervala $[0, 100>$. Potom je preko tipkovnice potrebno učitavati cijeli broj z sve dok se ne pogodi traženi slučajan broj x .

Ako je uspješno pogođen traženi broj x program se zaustavlja i potrebno je u konzolu ispisati poruku „Nakon y pokušaja uspješno je pogođen broj $x!$ “, pri čemu x predstavlja vrijednost broja koji se pogađa, a y vrijednost broja pokušaja.

Ako nije pogođen broj x potrebno je ispisati sugestiju korisniku koju vrijednost broja da upiše sljedeću i to: ukoliko je učitani broj z manji od x potrebno je ispisati poruku „Traženi broj je veći od broja $z!$ “, inače ako je z veći od x potrebno je ispisati poruku „Traženi broj je manji od broja $z!$ “ (z predstavlja vrijednost učitanoog broja).

```
static void Main(string[] args)
{
    Random rnd = new Random();
    int x = rnd.Next(0, 100);
    int y = 0;
    int z = -1;

    do
    {
        Console.Write("Unesite cijeli broj u rasponu [0,100>: ");
        z = Convert.ToInt32(Console.ReadLine());

        y++;

        if (z == x)
        {
            Console.WriteLine("Nakon " + y
                + " pokušaja uspješno je pogođen broj " + x + "!");
        }
        else if (z > x)
        {
            Console.WriteLine("Traženi broj je manji od broja " + z + "!");
        }
        else
        {
            Console.WriteLine("Traženi broj je veći od broja " + z + "!");
        }
    } while (x != z);

    Console.ReadKey();
}
```



Zadatak 4

Napišite kod C# programa koji u dvije liste učitava JMBAG studenata i broj bodova na ispitu (realan broj) studenata sve dok se ne učitava "prazan" JMBAG (pojašnjenje: JMBAG u listu JMBAG-ova, a broj bodova u listu bodova).

Potom program u konzolu treba ispisati JMBAG-ove i bodove onih studenata čiji je broj bodova veći od prosjeka. Format ispisa je proizvoljan.

```
static void Main(string[] args)
{
    List<string> listaJMBAG = new List<string>();
    List<double> listaBodova = new List<double>();

    string JMBAG = "";

    do
    {
        Console.Write("Unesite " + (listaJMBAG.Count + 1) + ". JMBAG: ");
        JMBAG = Console.ReadLine();

        if (JMBAG != "")
        {
            listaJMBAG.Add(JMBAG);

            Console.Write("Za " + listaJMBAG.Count + ". studenta (" + JMBAG
                + ") unesite broj bodova na ispitu: ");
            double bodoviNaIspitu = Convert.ToDouble(Console.ReadLine());

            listaBodova.Add(bodoviNaIspitu);
        }
    } while (JMBAG != "");

    double prosjecanBrojBodova = listaBodova.Average();

    Console.WriteLine("Studenti koji imaju veći broj bodova od prosjeka: ");

    for (int i = 0; i < listaBodova.Count; i++)
    {
        if (listaBodova[i] > prosjecanBrojBodova)
        {
            Console.WriteLine("JMBAG: " + listaJMBAG[i] + ", bodovi: "
                + listaBodova[i]);
        }
    }

    Console.ReadKey();
}
```



Zadatak 5

Što će se ispisati u konzolu nakon izvršavanja sljedećeg programskog odsječka? Potrebno je naznačiti redoslijed ispisivanja. Kao točno rješenje priznaje se samo ono u kojem je vidljiv postupak, odnosno prikazane su vrijednosti varijabli u svakoj iteraciji.

```
int[,] polje = new int[2, 3];
polje[0, 0] = polje.Length;
polje[0, 1] = polje.GetLength(0);
polje[1, 0] = polje.GetLength(1);
polje[1, 1] = polje.GetLength(1) * polje.Length;

for (int j = polje.GetLength(1) - 1; j > 0; j--)
{
    for (int i = polje.GetLength(0) - 1; i >= 0; i--)
    {
        Console.WriteLine(polje[i, j]);
    }
}
```

Izgled polja:

$$\begin{bmatrix} 6 & 2 & 0 \\ 3 & 18 & 0 \end{bmatrix}$$

Korak	j	i	Ispis
1	2	1	0
2	2	0	0
3	2	-1	
4	1	1	18
5	1	0	2
6	1	-1	
7	0		



Ispit #39 (02.06.20. - grupa A)

Zadatak 1

Napišite kod metode provjeraSifre koja kao argument prima šifru prometne nesreće (cijeli broj), a kao rezultat vraća istinu ili laž ovisno o tome da li je šifra ispravna ili ne. Šifra je ispravna ako je šifra peteroznamenkasti broj i ako je suma znamenki šifre veća od 10.

Primjeri:

243 - neispravna šifra jer nije peteroznamenkasti broj

15002 - neispravna šifra jer je suma znamenki 8

83010 - ispravna šifra jer je suma znamenki 12 i peteroznamenkasti je broj

```
static bool provjeraSifre(int sifra)
{
    int zbrojZnamenki = 0;
    int brojZnamenki = 0;

    while (sifra > 0)
    {
        int zadnjaZnamenka = sifra % 10;
        zbrojZnamenki += zadnjaZnamenka;
        brojZnamenki++;
        sifra /= 10;
    }

    if (zbrojZnamenki > 10 && brojZnamenki == 5)
    {
        return true;
    }

    return false;
}
```



Zadatak 2

Napišite kod dvije klase Lokacija i PrometnaNesreca.

Klasa Lokacija sadrži javne atribute x i y (realni brojevi) koji predstavljaju geografsku dužinu, odnosno širinu. Konstruktor klase Lokacija prima argumente za sve atribute klase.

Klasa PrometnaNesreca sadrži javne atribute Sifra (cijeli broj) koji predstavlja jedinstveni identifikator nesreće, Postaja (niz znakova) koji predstavlja naziv policijske postaje koja je evidentirala nesreću, DatumVrijeme (niz znakova) koji predstavlja datum i vrijeme kada se dogodila prometna nesreća i atribut LokacijaNesrece (tip klase Lokacija) koji predstavlja mjesto gdje se nesreća dogodila.

Unutar klase potrebno je napisati i javnu metodu Ispis koja ispisuje opis prometne nesreće u formatu: „Prometnu nesreću pod šifrom Sifra koja se dogodila DatumVrijeme na lokaciji (X, Y) evidentirala je policijska postaja Postaja!“. Vrijednosti Sifra, DatumVrijeme, X, Y i Postaja potrebno je dohvatiti iz odgovarajućih atributa klase.

Primjer ispisa: Prometnu nesreću pod šifrom 21810 koja se dogodila 25-05-2020 08:30 na lokaciji (16.106822, 45.827615) evidentirala je policijska postaja PP SESVETE!

```
class Lokacija
{
    public double x;
    public double y;

    public Lokacija(double X, double Y)
    {
        x = X;
        y = Y;
    }
}

class PrometnaNesreca
{
    public int Sifra;
    public string Postaja;
    public string DatumVrijeme;
    public Lokacija LokacijaNesrece;

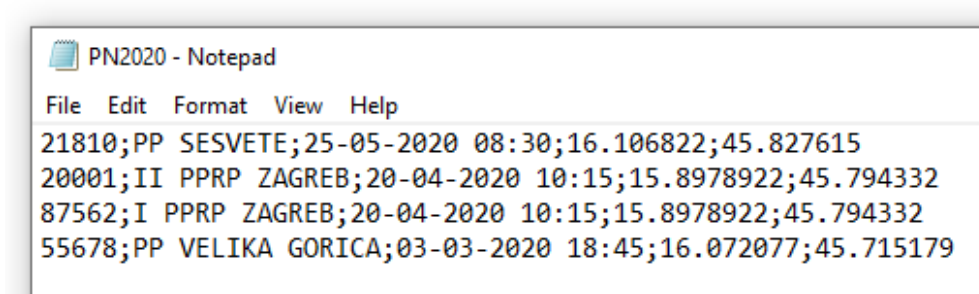
    public void Ispis()
    {
        Console.WriteLine("Prometnu nesreću pod šifrom " + Sifra
            + " koja se dogodila " + DatumVrijeme
            + " na lokaciji(" + LokacijaNesrece.x + ", "
            + LokacijaNesrece.y + ") evidentirala je policijska postaja "
            + Postaja + "!");
    }
}
```



Zadatak 3

Napišite C# program koji učitava datoteku o prometnim nesrećama preko apsolutne putanje. Datoteka naziva PN2020.txt spremljena je u direktorij E:\PN.

U jednom retku datoteke zapisana je informacija o šifri prometne nesreće (cijeli broj), policijskoj postaji (niz znakova), datumu-vremenu (niz znakova), geografskoj dužini X (realan broj) i geografskoj širini Y (realan broj), međusobno odijeljenih znakom ";" kao što je prikazano na slici.



Podatke je potrebno učitati u listu listaNesreca, na način da se u listu dodaje kreirana instanca klase PrometnaNesreca (zadatak 2) na odgovarajući način. U listu se dodaju samo one prometne nesreće koje imaju VALJANU šifru. Za provjeru valjanosti šifre koristiti poziv metode provjeraSifre (zadatak 1).

Na kraju je u konzolu potrebno ispisati koliko ima učitanih valjanih prometnih nesreća u proizvoljnom formatu.

Potrebno je koristiti try-catch blok te nije potrebno dodati imenski prostor System.IO.

Prilikom učitavanja za dijeljenje podataka u retku potrebno je koristiti ugrađenu metodu Split nad tipom podatka string. Primjer poziva metode Split nad retkom "Ovo-je-primjer!", s crticom '-' kao znakom za dijeljenje retka je sljedeći:

```
string linija = "Ovo-je-primjer!";
string[] dijeloviLinije = linija.Split('-');
```



```

using System.IO;

static void Main(string[] args)
{
    try
    {
        StreamReader reader = new StreamReader(@"E:\PN\PN2020.txt");

        List<PrometnaNesreca> listaNesreca = new List<PrometnaNesreca>();

        while (!reader.EndOfStream)
        {
            string linija = reader.ReadLine();
            string[] dijeloviLinije = linija.Split(';');

            if (provjeraSifre(Convert.ToInt32(dijeloviLinije[0])))
            {
                PrometnaNesreca p = new PrometnaNesreca();
                Lokacija l = new Lokacija(Convert.ToDouble(dijeloviLinije[3]),
                    Convert.ToDouble(dijeloviLinije[4]));
                p.Sifra = Convert.ToInt32(dijeloviLinije[0]);
                p.Postaja = dijeloviLinije[1];
                p.DatumVrijeme = dijeloviLinije[2];
                p.LokacijaNesrece = l;
                listaNesreca.Add(p);
            }
        }
        reader.Close();
        Console.WriteLine("Broj validnih prometnih nesreća: "
            + listaNesreca.Count());
    }
    catch (Exception ex)
    {
        Console.WriteLine(ex.Message);
    }

    Console.ReadKey();
}

```



Zadatak 4

Napišite kod konzolne aplikacije koja služi za ispis koliko je prometna nesreća PN1 na lokaciji (16.106822, 45.827615) daleko od lokacije koja se učitava preko konzole. Potrebno je s konzole učitati geografsku dužinu x (realan broj) i geografsku širinu y (realan broj) koji predstavljaju željenu lokaciju i ispisati koliko je PN1 udaljena od učitane lokacije. Za izračun udaljenosti potrebno je koristiti Euklidsku udaljenost prema formuli:

$$d(T_1(x_1, y_1), T_2(x_2, y_2)) = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

Za ispis udaljenosti potrebno je koristiti kategorizaciju danu tablicom:

Udaljenost	Tekst
[0, 0.025]	blizu
<0.025, 0.2]	daleko
<0.2, ∞]	jako daleko

Ispis formatirati na sljedeći način: „Učitana lokacija (x, y) je tekst zadanoj lokaciji prometne nesreće PN1!“, pri čemu je potrebno ispisati odgovarajuće vrijednosti x, y i tekst, primjerice: „Učitana lokacija (16.106802,45.827625) je blizu zadanoj lokaciji prometne nesreće PN1!“.

```
static void Main(string[] args)
{
    double x1 = 16.106822;
    double y1 = 45.827615;

    Console.Write("Unesite x koordinatu željene lokacije: ");
    double x2 = Convert.ToDouble(Console.ReadLine());

    Console.Write("Unesite y koordinatu željene lokacije: ");
    double y2 = Convert.ToDouble(Console.ReadLine());

    double d = Math.Sqrt(Math.Pow(x2 - x1, 2) + Math.Pow(y2 - y1, 2));

    string tekst = "";
    if (d >= 0 && d <= 0.025)
    {
        tekst = "blizu";
    }
    else if (d > 0.025 && d <= 0.2)
    {
        tekst = "daleko";
    }
    else
    {
        tekst = "jako daleko";
    }

    Console.WriteLine("Učitana lokacija (" + x2 + ", " + y2 + ") je " + tekst
        + " zadanoj lokaciji prometne nesreće PN1!");

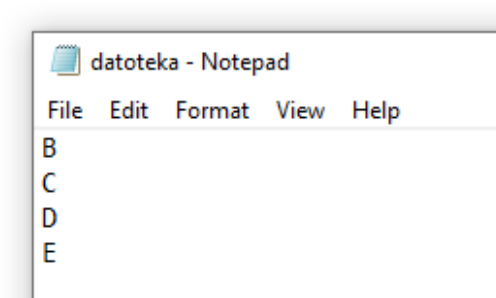
    Console.ReadKey();
}
```



Zadatak 5

Što će se ispisati nakon izvršavanja sljedećeg programskog odsječka? Potrebno je naznačiti redoslijed ispisivanja.

Pretpostavka je da je datoteka „datoteka.txt“ spremljena u projektni folder kako bi joj se moglo pristupiti preko relativne putanje, a njezin sadržaj je prikazan sljedećom slikom:



```
try
{
    StreamReader citanje = new StreamReader("datoteka.txt");
    string s = "A";

    while (!citanje.EndOfStream)
    {
        Console.WriteLine(s);
        s = citanje.ReadLine();
        s = citanje.ReadLine();
        Console.WriteLine(s);
    }
    citanje.Close();
    Console.WriteLine("F");
}
catch (Exception ex)
{
    Console.WriteLine("G");
}
Console.WriteLine("H");
```

Redoslijed	Ispis
1	A
2	C
3	C
4	E
5	F
6	H



6. Što će se ispisati u konzolu nakon izvršavanja sljedećeg programskog odsječka?

Potrebno je naznačiti redoslijed ispisivanja. Kao točno rješenje priznaje se samo ono u kojem je vidljiv postupak i sadržaj varijabli u svakoj iteraciji.

```
static void Main(string[] args)
{
    int[] p = new int[3];
    p[0] = p.Length;
    p[1] = -1;
    p[p.Length - 1] = p.Length / 2;
    Metoda(p);
    for (int i = 0; i < p.Length; i++)
    {
        Console.WriteLine(p[i]);
    }
}
private static void Metoda(int[] q)
{
    for (int i = 0; i < 2; i++)
    {
        q[i] = i;
    }
}
```

Izgled polja p unutar Main metode prije poziva metode Metoda:

[3 -1 1]

Izvršavanje metode Metoda:

Korak	i	Polje p
1	0	[0 -1 1]
2	1	[0 1 1]
3	2	[0 1 1]

Izgled polja p nakon izvršenja metode Metoda:

[0 1 1]

Izvršavanje petlje u Main metodi (nakon izvršenja metode Metoda):

Korak	i	Polje p	Ispis
1	0	[0 1 1]	0
2	1	[0 1 1]	1
3	2	[0 1 1]	1
3	3	[0 1 1]	



Ispit #40 (15.06.20. - grupa A)

Zadatak 1

Napišite C# kod metode ObradaDatuma koja kao argumente prima mjesec (cijeli broj) i dan (cijeli broj) koji predstavljaju datum u 2020. godini. Kao rezultat metoda vraća polje cijelih brojeva od 2 elementa.

Prvi element predstavlja oznaku dana u tjednu i treba poprimiti vrijednost: 1 - ukoliko se radi o radnom danu u tjednu (ponedjeljak, . . . , petak) ili 2 - ukoliko se radi o neradnom danu u tjednu (subota, nedjelja).

Drugi element predstavlja broj dana od početka 2020. godine. Pretpostavka je da svaki mjesec ima točno 30 dana i da je 01.01.2020. bila srijeda.

Primjeri:

Dan - 1, Mjesec - 1 metoda treba vratiti vrijednosti 1 i 1

Dan - 1, Mjesec - 2 metoda treba vratiti vrijednosti 1 i 31
Dan - 1, Mjesec - 3 metoda treba vratiti vrijednosti 2 i 61

Rješenje:

```
static int[] ObradaDatuma(int mjesec, int dan)
{
    int brojDanaUGodini = (mjesec - 1) * 30 + dan;

    int oznakaDana = brojDanaUGodini % 7;
    int danJeRadni = 0;

    if (oznakaDana != 4 && oznakaDana != 5)
    {
        danJeRadni = 1;
    }
    else
    {
        danJeRadni = 2;
    }

    int[] polje = new int[2];
    polje[0] = danJeRadni;
    polje[1] = brojDanaUGodini;

    return polje;
}
```



Zadatak 2

Napišite C# kod za učitavanje podataka o zapisima radara preko konzole. Za jedan zapis s radara potrebno je učitati vrijednost dana (cijeli broj), mjeseca (cijeli broj) i brzine (realan broj).

U jednodimenzionalno polje naziva brzine potrebno je učitati 100 VALJANIH zapisa brzina (ne svih nego samo valjanih). Zapis je valjan ukoliko je dan na koji se dogodio zapis bio radni (ponedjeljak, . . . , petak) i ukoliko je broj dana proteklih od početka 2020. godine veći od 100.

Za dohvat navedenih vrijednosti potrebno je koristiti metodu `ObradaDatuma` iz Zadatka 1.

Na kraju je potrebno ispisati vrijednost maksimalne valjane brzine i pripadajućeg dana i mjeseca, te vrijednost prosječne valjane brzine u formatu kao u primjeru.

Primjer: „Maksimalna valjana brzina iznosi 93.2 km/h i zabilježena je 15.06.2020! Prosječna valjana brzina iznosi 35,72856666 km/h!“.



Rješenje:

```
static void Main(string[] args)
{
    double[] brzine = new double[100];

    int indeksPolja = 0;
    double maksimalnaBrzina = double.MinValue;
    int danMaksimalneBrzine = -1;
    int mjesecMaksimalneBrzine = -1;
    double sumaBrzina = 0.0;

    while (indeksPolja < brzine.Length)
    {
        Console.WriteLine("Unesite mjesec iz 2020. godine: ");
        int mjesec = Convert.ToInt32(Console.ReadLine());

        Console.WriteLine("Unesite dan iz " + mjesec
            + ". mjeseca 2020. godine: ");
        int dan = Convert.ToInt32(Console.ReadLine());

        Console.WriteLine("Unesite brzinu za datum " + dan + ". "
            + mjesec + ". 2020.: ");
        double brzina = Convert.ToDouble(Console.ReadLine());

        int[] provjeraValjanostiDatuma = ObradaDatuma(mjesec, dan);

        if (provjeraValjanostiDatuma[0]==1 && provjeraValjanostiDatuma[1]>100)
        {
            brzine[indeksPolja] = brzina;
            indeksPolja++;
            sumaBrzina += brzina;
            if (brzina > maksimalnaBrzina)
            {
                maksimalnaBrzina = brzina;
                danMaksimalneBrzine = dan;
                mjesecMaksimalneBrzine = mjesec;
            }
        }
    }

    double prosjecnaBrzina = sumaBrzina / brzine.Length;

    Console.WriteLine("Maksimalna valjana brzina iznosi " + maksimalnaBrzina
        + " km/h i zabilježena je " + danMaksimalneBrzine
        + "." + mjesecMaksimalneBrzine + "."
        + "2020.! Prosječna valjana brzina iznosi " + prosjecnaBrzina + " km/h!");

    Console.ReadKey();
}
```

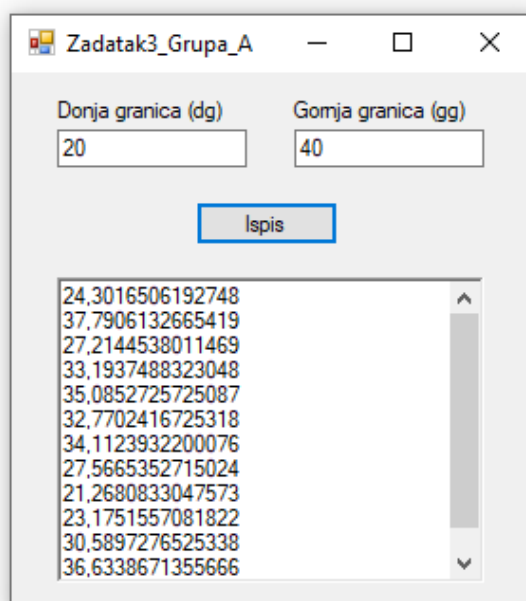


Zadatak 3

Nadopunite C# kod Windows forme koja služi za vizualizaciju podataka učitanih u jednodimenzionalno polje realnih brojeva naziva brzine (zadatak 2).

Pritiskom na gumb Ispis naziva btnIspis potrebno je u kontrolu tipa RichTextBox naziva txtIspis ispisati sve brzine iz polja brzine koje se nalaze unutar intervala <dg, gg>, pri čemu se granice dg i gg (realni brojevi) unose preko kontrola tipa TextBox naziva txtDonja i txtGornja.

Svaka brzina mora biti u zasebnom retku, a oznaka za prelazak u novi red je "\n". Sučelje je prikazano na sljedećoj slici:



```
//Brzine su spremljene u polje naziva brzine (zadatak 2)
public Form1()
{
    InitializeComponent();
}

private void btnIspis_Click(object sender, EventArgs e)
{
    //Nadopuni
}
```



Rješenje:

```
//Brzine su spremljene u polje naziva brzine (zadatak 2)
public Form1()
{
    InitializeComponent();
}

private void btnIspis_Click(object sender, EventArgs e)
{
    double dg = Convert.ToDouble(txtDonja.Text);
    double gg = Convert.ToDouble(txtGornja.Text);

    string porukaZaIspis = "";
    for (int i = 0; i < brzine.Length; i++)
    {
        if (brzine[i] > dg && brzine[i] < gg)
        {
            porukaZaIspis += brzine[i] + "\n";
        }
    }

    txtIspis.Text = porukaZaIspis;
}
```

Rješenje 2:

```
//Brzine su spremljene u polje naziva brzine (zadatak 2)
public Form1()
{
    InitializeComponent();
}

private void btnIspis_Click(object sender, EventArgs e)
{
    double dg = Convert.ToDouble(txtDonja.Text);
    double gg = Convert.ToDouble(txtGornja.Text);

    txtIspis.Text = "";
    for (int i = 0; i < brzine.Length; i++)
    {
        if (brzine[i] > dg && brzine[i] < gg)
        {
            txtIspis.Text += brzine[i] + "\n";
        }
    }
}
```



Zadatak 4

Što će se ispisati u konzolu nakon izvršavanja sljedećeg programskog odsječka. Potrebno je naznačiti redoslijed ispisivanja i vrijednosti varijabli u svakom pozivu metoda.

```
static void Main(string[] args)
{
    double a = 4;
    Metoda1(a);
    Console.WriteLine(a);
}

static void Metoda1(double a)
{
    a *= 2;
    Console.WriteLine(a);
    double b = Metoda3(a);
    Console.WriteLine(b);
}

static void Metoda2()
{
    Console.WriteLine("Test1");
}

static double Metoda3(double a)
{
    Console.WriteLine("Test1");
    return Math.Pow(a, 2);
}
```



Rješenje:

Redoslijed ispisa	Ispis	a	Dodatni komentari
		Main(a=4)	Program kreće iz Main metode i vrijednost varijable a = 4
		Main(a=4), Metoda1(a=4)	Poziva se Metoda1(a=4)
		Main(a=4), Metoda1(a=8)	U Metoda1 $a=a*2=8$
1	8	Main(a=4), Metoda1(a=8)	Ispis a=8 u Metoda1
		Main(a=4), Metoda1(a=8), Metoda3(a=8)	Poziva se Metoda3(a=8)
2	Test1	Main(a=4), Metoda1(a=8), Metoda3(a=8)	Ispis u Metoda3(a=8)
		Main(a=4), Metoda1(a=8, b=64)	Vraća se Math.Pow(8,2)=64 iz Metoda3 na mjesto gdje je Metoda3 bila pozvana, tj. u Metoda1 i sada je u Metoda 1 varijabla b=64
3	64	Main(a=4), Metoda1(a=8, b=64)	Ispis v=64 u Metoda1
		Main(a=4)	Nakon zadnjeg ispisa završava se Metoda1 i program se vraća na mjesto gdje je Metoda1 bila pozvana, tj. u Main metodu
4	4		Zadnji ispis u Main metodu za a=4, nakon toga se završava program



Zadatak 5

Što će se ispisati u konzolu nakon izvršavanja sljedećeg programskog odsječka. Potrebno je naznačiti redoslijed ispitivanja. Kao točno rješenje priznaje se samo ono u kojem je vidljiv postupak i sadržaj varijabli u svakoj iteraciji.

```
List<int> lista = new List<int>();
lista.Add(-1);
lista.AddRange(new int[] { -1, 0, 1 });
lista.RemoveAt(lista.Count - 2);
lista.Insert(lista.Count - 2, 2);
for (int i = 0; i < lista.Count; i++)
{
    Console.WriteLine(lista[i]);
}
```

Rješenje:

Redoslijed ispisa	Ispis	lista	Dodatni komentari
		[]	List<int> lista = new List<int>() kreira se prazna lista
		[-1]	lista.Add(-1); na zadnje mjesto u listi dodaje se -1
		[-1 -1 0 1]	lista.AddRange(new int[] { -1, 0, 1 }); na kraj liste dodaju se brojevi -1,0,1
		[-1 -1 1]	lista.RemoveAt(lista.Count - 2); lista.Count = 4 pa se iz liste briše element na indeksu 2, tj. treći broj, a to je 0
		[-1 2 -1 1]	lista.Insert(lista.Count - 2, 2); lista.Count=3 pa se broj 2 ubacuje na indeks 1
1	-1	[-1 2 -1 1]	Ulazak u petlju, za i=0 se ispisuje broj iz liste na indeksu 0
2	2	[-1 2 -1 1]	za i=1 se ispisuje broj iz liste na indeksu 1
3	-1	[-1 2 -1 1]	za i=2 se ispisuje broj iz liste na indeksu 2
4	1	[-1 2 -1 1]	za i=3 se ispisuje broj iz liste na indeksu 3
			Uvjet ponavljanja petlje za i=4 (4<4) nije istinit pa se izlazi iz for petlje i program se završava



Ispit #41 (26.06.20. - grupa A)

Zadatak 1

Što će se ispisati u konzolu nakon izvršavanja sljedećeg programskog odsječka. Potrebno je naznačiti redoslijed ispisivanja. Kao točno rješenje priznaje se samo ono u kojem je vidljiv postupak i sadržaj varijabli.

```
int a = 16 % 7;
int b = 5 / a;
int c, d;

if (a >= 2 && (a != 3 ^ a == 2))
{
    c = ++a;
    d = b % 2;
}
else
{
    c = a++;
    d = a * 3 - 2;
}

Console.WriteLine(a + ", " + b + ", " + c + ", " + d);
```



Rješenje:

Ispis	a	b	c	d	Dodatni komentari
	2				% je ostatak cjelobrojnog dijeljenja $a=16\%7=2$
	2	2			dijeljenje int-ova (cijelih brojeva) uvijek "gubi" decimale, tj. daje cijeli broj $b=5/2=5/2=2$
					if uvjet: $a \geq 2 \ \&\& \ (a \neq 3 \wedge a == 2)$ $2 \geq 2 \ \&\& \ (2 \neq 3 \wedge 2 == 2)$ $true \ \&\& \ (true \wedge true)$ $true \ \&\& \ false$ $false$ Obzirom da je uvjet if-a bio false izvršava se else dio uvjetnog granjanja
	3	2	2		$c=a++;$ prvo se u varijablu c sprema vrijednost od a (2) na kraju tog reda se i vrijednost od a mijenja, tj. povećava se za 1
	3	2	2	7	$d=a*3-2=3*3-2=9-2=7$
3, 2, 2, 7	3	2	2	7	Console.WriteLine() ispisuje poruku



Zadatak 2

Što će se ispisati u konzolu nakon izvršavanja sljedećeg programskog odsječka. Potrebno je naznačiti redoslijed ispisivanja. Kao točno rješenje priznaje se samo ono u kojem je vidljiv postupak i sadržaj varijabli u svakoj iteraciji.

```
int i, j;

for (i = 1; i <= 3; i++)
{
    for (j = 1; j <= 4; j++)
    {
        if (j % 2 != 0)
            continue;
        Console.WriteLine(j);
        if (j != i)
            break;
    }
}
```

Rješenje:

Ispis	i	j	Dodatni komentari
			u prvom redu deklariraju se varijable i, j još nemaju nikakvu vrijednost
	1		ulaskom u vanjsku petlju definira se i = 1
		1	ulaskom u unutarnju petlju definira se j = 1
			uvjet if-a je istinit ako je vrijednost varijable j neparna obzirom da je j = 1, tj. neparan je, uvjet je istina i izvršava se naredba continue
		2	naredba continue završava trenutnu iteraciju unutarnje petlje na kraju iteracije j se povećava za 1 (j++)



Ispis	i	j	Dodatni komentari
2			u prolasku unutarnje petlje za j = 2 prvi uvjet nije istina potom dolazimo do Console.WriteLine(j) koji ispisuje vrijednost varijable j
			uvjet j != i je istina i izvršava se naredba break naredba break bezuvjetno završava petlju u kojoj se nalazi (unutarnja petlja)
	2		obzirom da je unutarnja petlja gotova, dolazimo do kraja vanjske petlje na kraju iteracije i se povećava za 1 (i++)
		1	vanjska petlja se ponavlja za i = 2 ulaskom u unutarnju petlju definira se j = 1
			uvjet if-a je istinit ako je vrijednost varijable j neparna obzirom da je j=1, tj. neparan je, uvjet je istina i izvršava se naredba continue
		2	naredba continue završava trenutnu iteraciju unutarnje petlje na kraju iteracije j se povećava za 1 (j++)
2			u prolasku unutarnje petlje za j = 2 prvi uvjet nije istina potom dolazimo do Console.WriteLine(j) koji ispisuje vrijednost varijable j
		3	uvjet j != i nije istina pa se preskače naredba break na kraju iteracije j se povećava za 1 (j++)



Ispis	i	j	Dodatni komentari
			unutarnja petlja se ponavlja za $j = 3$ obzirom da je $j = 3$, tj. neparan je, uvjet je istina i izvršava se naredba continue
		4	naredba continue završava trenutnu iteraciju unutarnje petlje na kraju iteracije j se povećava za 1 ($j++$)
4			u prolasku unutarnje petlje za $j = 4$ prvi uvjet nije istina potom dolazimo do <code>Console.WriteLine(j)</code> koji ispisuje vrijednost varijable j
			uvjet $j != i$ je istina i izvršava se naredba break naredba break bezuvjetno završava petlju u kojoj se nalazi (unutarnja petlja)
	3		obzirom da je unutarnja petlja gotova, dolazimo do kraja vanjske petlje na kraju iteracije i se povećava za 1 ($i++$)
		1	vanjska petlja se ponavlja za $i = 3$ ulaskom u unutarnju petlju definira se $j = 1$
			uvjet if-a je istinit ako je vrijednost varijable j neparna obzirom da je $j=1$, tj. neparan je, uvjet je istina i izvršava se naredba continue
		2	naredba continue završava trenutnu iteraciju unutarnje petlje na kraju iteracije j se povećava za 1 ($j++$)



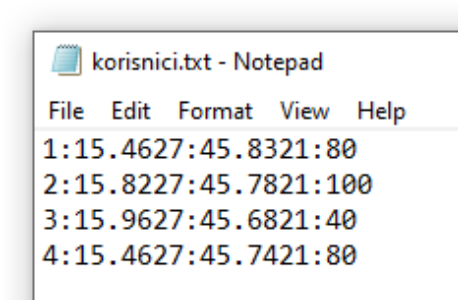
Ispis	i	j	Dodatni komentari
2			u prolasku unutarnje petlje za $j = 2$ prvi uvjet nije istina potom dolazimo do <code>Console.WriteLine(j)</code> koji ispisuje vrijednost varijable <code>j</code>
			uvjet $j != i$ je istina i izvršava se naredba <code>break</code> naredba <code>break</code> bezuvjetno završava petlju u kojoj se nalazi (unutarnja petlja)
	4		obzirom da je unutarnja petlja gotova, dolazimo do kraja vanjske petlje na kraju iteracije <code>i</code> se povećava za 1 (<code>i++</code>)
			uvjet ponavljanja vanjske petlje $4 \leq 3$ nije istina i petlja se više ne ponavlja



Zadatak 3

Napišite C# kod za učitavanje podataka o 100 korisnika dostave koji se nalaze unutar datoteke korisnici.txt, kojoj se treba pristupiti preko relativne putanje.

U jednom retku datoteke zapisana je informacija o šifri korisnika (cijeli broj), geografskoj dužini X (realan broj), geografskoj širini Y (realan broj) i količini (cijeli broj) paketa koji je potrebno dostaviti tom korisniku, međusobno odijeljenih znakom ":", kao što je prikazano na slici.



Podatke je potrebno učitati u polje korisnici koje treba sadržavati instance klase Korisnik prikazane u nastavku.

```
public class Korisnik
{
    public int sifra;
    public double x;
    public double y;
    public int kolicina;

    public Korisnik(int KOLICINA, int SIFRA, double X, double Y)
    {
        sifra = SIFRA;
        x = X;
        y = Y;
        kolicina = KOLICINA;
    }
}
```

Na kraju je u konzolu u proizvoljnom formatu potrebno ispisati kolika je ukupna količina paketa koju je potrebno prevesti, te koliki je minimalan broj vozila potreban za prijevoz ako je poznato da je kapacitet vozila 200.

Za izračun gornje granice za broj vozila potrebno je koristiti ugrađenu metodu Math.Ceiling koja zaokružuje realan broj na najbliži veći cijeli broj, ali i dalje vraća realan broj.

Potrebno je koristiti try-catch blok te nije potrebno dodati imenski prostor System.IO. Prilikom učitavanja za dijeljenje podataka u retku potrebno je koristiti ugrađenu metodu Split nad tipom podatka string. Primjer poziva metode Split nad retkom "Ovo-je-primjer!", s crticom '-' kao znakom za dijeljenje retka je sljedeći:

```
string linija = "Ovo-je-primjer!";
string[] dijeloviLinije = linija.Split('-');
```



Rješenje:

```
using System.IO;

static void Main(string[] args)
{
    Korisnik[] korisnici = new Korisnik[100];

    try
    {
        StreamReader reader = new StreamReader("korisnici.txt");

        int ukupnaKolicinaPaketa = 0;

        for (int i = 0; i < korisnici.Length; i++)
        {
            string linija = reader.ReadLine();
            string[] dijeloviLinije = linija.Split(':');

            Korisnik k = new Korisnik(Convert.ToInt32(dijeloviLinije[3]),
                Convert.ToInt32(dijeloviLinije[0]),
                Convert.ToDouble(dijeloviLinije[1]),
                Convert.ToDouble(dijeloviLinije[2]));
            korisnici[i] = k;

            ukupnaKolicinaPaketa += k.kolicina;
        }

        reader.Close();

        int brojVozila = Convert.ToInt32(
            Math.Ceiling(ukupnaKolicinaPaketa / 200.0));

        Console.WriteLine("Količina paketa koju je potrebno prevesti: "
            + ukupnaKolicinaPaketa);
        Console.WriteLine("Minimalan broj vozila za prijevoz paketa: "
            + brojVozila);
    }
    catch (Exception ex)
    {
        Console.WriteLine("Greška: " + ex.Message);
    }

    Console.ReadKey();
}
```



Zadatak 4

Napišite C# kod za raspoređivanje korisnika iz polja korisnici (Zadatak 3) po vozilima. Raspoređivanje po vozilima vršit će se na način da se počevši od početka polja korisnici redom dodaju u vozilo dok god nije premašen kapacitet vozila u iznosi od 200 (niti jedan korisnik nema količinu 200).

Nakon prekoračenja kapaciteta vozila potrebno je trenutni raspored korisnika po vozilu zapisati u datoteku raspored.txt (svaki raspored korisnika u zasebnom retku) te potom "otvoriti" novo vozilo, dodati trenutnog korisnika u to novo vozilo i ponoviti postupak sve dok se ne obrade svi korisnici iz polja.

Raspored korisnika unutar vozila potrebno je bilježiti kao niz odgovarajućih šifri korisnika odvojenih znakom ';'. Za primjer sa slike 1 prvi redak u datoteci raspored.txt trebao bi biti 1;2; (jer je zbroj korisnika pod šiframa 1 i 2 180, a dodavanje sljedećeg korisnika pod šifrom 3 premašilo bi kapacitet 200). Prema tome drugi redak u datoteci započeo bi s 3;4; itd.

Nije potrebno koristiti try-catch blok te nije potrebno dodati imenski prostor System.IO, a datoteci se pristupa preko relativne putanje.

```
StreamWriter writer = new StreamWriter("raspored.txt");

int trenutnaSumaKolicina = 0;

for (int i = 0; i < korisnici.Length; i++)
{
    int trenutnaKolicina = korisnici[i].kolicina;

    if (trenutnaSumaKolicina + trenutnaKolicina <= 200)
    {
        trenutnaSumaKolicina += trenutnaKolicina;
    }
    else
    {
        writer.WriteLine();
        trenutnaSumaKolicina = trenutnaKolicina;
    }
    writer.Write(korisnici[i].sifra + ";");
}

writer.Flush();
writer.Close();

Console.ReadKey();
```



Zadatak 5

Napišite C# kod koji s konzole učitava geografsku točku T_1 s dužinom x_1 (realan broj) i širinom y_1 (realan broj), te potom računa udaljenost do slučajne geografske točke T_2 s dužinom x_2 unutar intervala $[0, 20.18>$ i širinom y_2 unutar intervala $[0, 51.28>$.

Učitavanje koordinata točke T_1 i dohvat slučajnih koordinata točke T_2 potrebno je ponavljati sve dok udaljenost između točaka nije manja od 10 km.

Za izračun udaljenosti u kilometrima potrebno je koristiti sljedeću jednadžbu:

$$d(T_1(x_1, y_1), T_2(x_2, y_2)) = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} * 112.32$$

U svakoj iteraciji potrebno je ispisati koliko je učitana točka daleko od slučajno generirane točke prema tablici:

Udaljenost	Tekst
< 50	jako blizu
[50, 150>	blizu
> 150	daleko

Format ispisa treba biti kao u primjeru. Primjerice neka se u jednoj iteraciji petlje učitava $T_1 (x_1, y_1) = (12.789, 45.789)$ i slučajno dohvati $T_2 (x_2, y_2) = (15.7401, 38.306)$ potrebno je ispisati: „Učitana točka T1 (12.789, 45.789) je daleko slučajnoj točki T2 (15.7401, 38.306)!“.



Rješenje:

```
static void Main(string[] args)
{
    double trenutnaUdaljenost = 0.0;

    do
    {
        Console.Write("Unesite x1: ");
        double x1 = Convert.ToInt32(Console.ReadLine());

        Console.Write("Unesite y1: ");
        double y1 = Convert.ToInt32(Console.ReadLine());

        Random rnd = new Random();

        double x2 = rnd.NextDouble() * 20.18;
        double y2 = rnd.NextDouble() * 51.28;

        trenutnaUdaljenost = Math.Sqrt(Math.Pow(x2 - x1, 2)
            + Math.Pow(y2 - y1, 2)) * 112.32;

        string tekst = "";

        if (trenutnaUdaljenost < 50)
        {
            tekst = "jako blizu";
        }
        else if (trenutnaUdaljenost < 150)
        {
            tekst = "blizu";
        }
        else
        {
            tekst = "daleko";
        }

        Console.WriteLine("Učitana točka T1(" + x1 + ", " + y1
            + ") je " + tekst + " slučajnoj točki T2(" + x2 + ", " + y2 + ")!");
    } while (trenutnaUdaljenost >= 10);

    Console.ReadKey();
}
```



Ispit #42 (09.07.20. - grupa A)

Zadatak 1

Što će se ispisati u konzolu nakon izvršavanja sljedećeg programskog odsječka. Potrebno je naznačiti redoslijed ispisivanja. Kao točno rješenje priznaje se samo ono u kojem je vidljiv postupak i sadržaj varijabli u svakom pozivu metode.

```
static void Main(string[] args)
{
    Ispis(4);
    Console.ReadKey();
}

private static void Ispis(int x)
{
    Console.WriteLine(x);
    if (x < 2)
    {
        return;
    }
    else if (x <= 4)
    {
        Ispis(x - 2);
    }
    else
    {
        Ispis(x - 1);
    }
    Console.WriteLine(x);
}
```



Rješenje:

Redoslijed ispisa	Ispis	Dodatni komentari
		Poziv metode Ispis (x=4)
1	4	Prvi Console.WriteLine() u metodi Ispis(x=4)
		Uvjet $x \leq 4$ je istina pa se poziva metoda Ispis (x=2)
2	2	Prvi Console.WriteLine() u metodi Ispis(x=2)
		Uvjet $x \leq 4$ je istina pa se poziva metoda Ispis (x=0)
3	0	Prvi Console.WriteLine() u metodi Ispis(x=0)
		Uvjet $x < 2$ je istina pa se izvršava naredba return, tj. metoda Ispis(x=0) se završava
		Program se vraća na mjesto gdje je bila pozvana metoda Ispis(x=0), tj. u metodu Ispis(x=2)
4	2	Poslijednji Console.WriteLine() u metodi Ispis(x=2)
		Završava se metoda Ispis(x=2) i program se vraća na mjesto gdje je bila pozvana, tj. u metodu Ispis(x=4)
5	4	Poslijednji Console.WriteLine() u metodi Ispis(x=4)
		Završava se metoda Ispis(x=4) i program se vraća na mjesto gdje je bila pozvana, tj. u Main metodu
		U Main metodi pokreće se Console.ReadKey() i nakon toga se završava program



Zadatak 2

Što će se ispisati u konzolu nakon izvršavanja sljedećeg programskog odsječka. Potrebno je naznačiti redoslijed ispisivanja. Kao točno rješenje priznaje se samo ono u kojem je vidljiv postupak i sadržaj varijabli u svakoj iteraciji.

```
string tekst = "Programiranje";
int i = 0;

try
{
    for (i = 1; i < tekst.Length; i *= 2)
    {
        Console.WriteLine(tekst[i]);
    }
}
catch (Exception ex)
{
    Console.WriteLine(tekst[tekst.Length / 3]);
}
Console.WriteLine(tekst[tekst.Length / 3]);
```

Rješenje:

Kako bi mogli lakše pratiti na kojem indeksu se nalazi koje slovo unutar riječi možete si unaprijed napraviti sljedeću tablicu:

Slova	P	r	o	g	r	a	m	i	r	a	n	j	e
Indeks slova	0	1	2	3	4	5	6	7	8	9	10	11	12

Redoslijed ispisa	Ispis	i	Dodatni komentari
		0	Deklaracija i inicijalizacija varijable i
1	r	1	Ulazak u petlju, i se povećava tako da se množi s dva svakim prolaskom kroz petlju, a text.Length=13
2	o	2	
3	r	4	
4	r	8	
		16	Uvjet ponavljanja petlje (16<13) više nije istinit pa se ne ulazi u petlju
			U try dijelu nije bilo greške pa se catch dio nikada ne pokreće
5	r		Posljednji Console.WriteLine() u metodi, tekst.Length/3=4



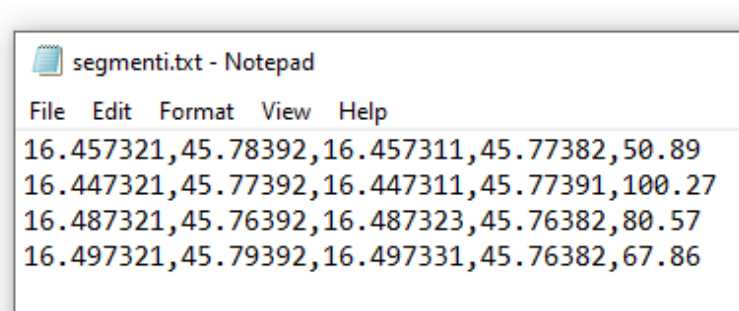
Zadatak 3

Napišite C# kod za učitavanje podataka o cestovnim segmentima u digitalnoj cestovnoj karti RH. Pojedini cestovni segment sadrži informacije o geografskoj dužini i širini početka (x_1, y_1) i kraja (x_2, y_2) (realni brojevi) i prosječnoj brzini v (realan broj) na cestovnom segmentu.

Podaci se trebaju učitavati iz datoteke segmenti.txt u listu segmenti, koja treba sadržavati instance klase Segment prikazane u nastavku.

```
public class Segment
{
    public double x1;
    public double y1;
    public double x2;
    public double y2;
    public double v;
}
```

U jednom retku datoteke zapisana je informacija o cestovnom segmentu u formatu: x_1, y_1, x_2, y_2, v kao što je prikazano na slici:



Instancu cestovnog segmenta potrebno je dodati u listu segmenti samo ako je brzina segmenta unutar intervala $<40, 60>$ ili $[80, 100>$. Na kraju je u konzolu u proizvoljnom formatu potrebno ispisati kolika je suma duljina svih valjanih cestovnih segmenata. Za izračun duljine pojedinog cestovnog segmenta potrebno je koristiti izraz:

$$d(T_1(x_1, y_1), T_2(x_2, y_2)) = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} * 111.32$$

Potrebno je koristiti try-catch blok te nije potrebno dodati imenski prostor System.IO. Prilikom učitavanja za dijeljenje podataka u retku potrebno je koristiti ugrađenu metodu Split nad tipom podataka string. Primjer poziva metode Split nad retkom „Ovo je primjer!“, s crticom '-' kao znakom za dijeljenje retka, je sljedeći:

```
string linija = "Ovo-je-primjer!";
string[] dijeloviLinije = linija.Split('-');
```



Rješenje:

```
static void Main(string[] args)
{
    try
    {
        StreamReader reader = new StreamReader("segmenti.txt");

        List<Segment> segmenti = new List<Segment>();
        double sumaDuljinaSegmenata = 0;

        while (!reader.EndOfStream)
        {
            string linija = reader.ReadLine();
            string[] dijeloviLinije = linija.Split(',');

            Segment s = new Segment();
            s.x1 = Convert.ToDouble(dijeloviLinije[0]);
            s.y1 = Convert.ToDouble(dijeloviLinije[1]);
            s.x2 = Convert.ToDouble(dijeloviLinije[2]);
            s.y2 = Convert.ToDouble(dijeloviLinije[3]);
            s.v = Convert.ToDouble(dijeloviLinije[4]);

            if ((s.v > 40 && s.v < 60) || (s.v >= 80 && s.v < 100))
            {
                segmenti.Add(s);
                sumaDuljinaSegmenata += Math.Sqrt(Math.Pow(s.x2 - s.x1, 2)
                    + Math.Pow(s.y2 - s.y1, 2)) * 111.32;
            }

            reader.Close();

            Console.WriteLine("Suma duljina valjanih cestovnih segmenata je "
                + sumaDuljinaSegmenata + " km.");
        }
        catch (Exception ex)
        {
            Console.WriteLine("Greška: " + ex.Message);
        }

        Console.ReadKey();
    }
}
```

Napomena: ako ćete provjeravati zadatak moguće je da će Vam zadatak izračunati krivu sumu valjanih duljina, tj. ispisat će se „0“ umjesto pravog rezultata. Do problema dolazi zbog formata brojeva u Windowsima. Hrvatski format ignorira točku unutar broja, pa je za decimalni broj potrebno staviti zarez umjesto točke koja je napisana u datoteci segmenti.txt. Problem se može popraviti naredbom Replace kod dodjeljivanja vrijednosti atributima klase Segmenti. Primjer popravljivanja rješenja:

```
red: s.x1 = Convert.ToDouble(dijeloviLinije[0]);
zamijeniti redom: s.x1 = Convert.ToDouble(dijeloviLinije[0].Replace('.', ','));
```



Zadatak 4

Napišite C# kod metode MapMatching koja kao argument prima 3 vrijednosti: geografsku dužinu x_1 , geografsku širinu y_1 i listu segmenata segmenti (zadatak 3).

Metoda za zadanu točku $T_1(x_1, y_1)$ treba pronaći najbliži cestovni segment prema kriteriju zadanom izrazom koji se nalazi na kraju ovog zadatka, pri čemu S_i označava prijedeni cestovni segment iz liste, $P_{S_i}(x_p, y_p)$ točku polovišta segmenta S_i određenu koordinatama x_p i y_p (koju je potrebno izračunati), $d(T_1(x_1, y_1), P_{S_i}(x_p, y_p))$ udaljenost između dvije točke danu izrazom iz prethodnog zadatka (zadatak 3), a $KutIzmeđu(x_1, y_1, S_i)$ poziv metode KutIzmeđu koja vraća kut u stupnjevima između točke $T_1(x_1, y_1)$ i segmenta S_i .

Metodu KutIzmeđu nije potrebno implementirati, već se pretpostavlja da je ona već implementirana, ali je potrebno napisati točan poziv metode. Metoda MapMatching treba vratiti instancu cestovnog segmenta koji je najbliži zadanoj točki prema navedenom kriteriju.

$$F(T_1(x_1, y_1), S_i) = 1000 * d(T_1(x_1, y_1), P_{S_i}(x_p, y_p)) + 0.5 * KutIzmeđu(x_1, y_1, S_i)$$

Rješenje:

```
static Segment MapMatching(double x1, double y1, List<Segment> segmenti)
{
    double najmanjaUdaljenost = double.MaxValue;
    int indeksSegmentaSNajmanjomUdaljenosti = 0;

    for (int i = 0; i < segmenti.Count; i++)
    {
        double xp = (segmenti[i].x1 + segmenti[i].x2) / 2;
        double yp = (segmenti[i].y1 + segmenti[i].y2) / 2;

        double d = Math.Sqrt(Math.Pow(xp - x1, 2) + Math.Pow(yp - y1, 2)) * 111.32;

        double F = 1000 * d + 0.5 * KutIzmeđu(x1, y1, segmenti[i]);

        if (F < najmanjaUdaljenost)
        {
            najmanjaUdaljenost = F;
            indeksSegmentaSNajmanjomUdaljenosti = i;
        }
    }

    return segmenti[indeksSegmentaSNajmanjomUdaljenosti];
}
```



Zadatak 5

Napišite C# kod koji s konzole učitava tri mase paketa u proizvodnom pogonu u kilogramima m_1 , m_2 i m_3 (cijeli brojevi).

Potom se na paletu maksimalne zapremnine 500 kg dodaje samo onaj paket koji od trenutno tri učitana ima najveću masu.

Učitavanje tri po tri paketa potrebno je ponavljati sve dok se najteži paket može dodati na paletu. U svakom koraku petlje potrebno je u formatu kao u primjeru ispisati kolika je masa paketa dodanog na paletu, te kolike su mase odbačenih paketa.

Primjerice: „Na paletu je stavljen paket mase 3 kg, a odbačeni su paketi mase 1 i 2 kg.“.

Na kraju je u formatu kao u primjeru potrebno ispisati koliko je paketa stavljeno na paletu i kolika je ukupna masa palete.

Primjer ispisa: „Na paletu je stavljeno ukupno 50 paketa, ukupne mase 490 kg!“.



Rješenje:

```
static void Main(string[] args)
{
    int[] masePaketa = new int[3];
    int brojPrihvacenihPaketa = 0;
    int ukupnaMasa = 0;

    do
    {
        int indeksNajveceMase = 0;
        int najvecMasa = int.MinValue;

        for (int i = 0; i < masePaketa.Length; i++)
        {
            Console.Write("Unesite masu " + (i + 1) + ". paketa: ");
            masePaketa[i] = Convert.ToInt32(Console.ReadLine());

            if (masePaketa[i] > najvecMasa)
            {
                najvecMasa = masePaketa[i];
                indeksNajveceMase = i;
            }
        }
        if ((ukupnaMasa + masePaketa[indeksNajveceMase]) <= 500)
        {
            ukupnaMasa += masePaketa[indeksNajveceMase];
            brojPrihvacenihPaketa++;

            Console.Write("Na paletu je stavljen paket mase "
                + masePaketa[indeksNajveceMase]
                + " kg, a odbačeni su paketi mase ");

            int brojNeprihvacenih = 0;

            for (int i = 0; i < masePaketa.Length; i++)
            {
                if (i != indeksNajveceMase)
                {
                    Console.Write(masePaketa[i]);
                    brojNeprihvacenih++;
                }
                if (brojNeprihvacenih == 1)
                {
                    Console.Write(" i ");
                }
            }
            Console.WriteLine(" kg.");
        }
        else
        {
            Console.WriteLine("Svi paketi su odbačeni jer bi dodavanje paketa
                od " + masePaketa[indeksNajveceMase]
                + " kg prekoračilo dozvoljenu masu na paleti!");
            break;
        }
    } while (true);
    Console.WriteLine("Na paletu je stavljeno ukupno "
        + brojPrihvacenihPaketa + " paketa, ukupne mase " + ukupnaMasa + " kg!");

    Console.ReadKey();
}
```



Ispit #43 (09.07.20. - grupa B)

Zadatak 1

Što će se ispisati u konzolu nakon izvršavanja sljedećeg programskog odsječka. Potrebno je naznačiti redoslijed ispisivanja. Kao točno rješenje priznaje se samo ono u kojem je vidljiv postupak i sadržaj varijabli u svakom pozivu metode.

```
static void Main(string[] args)
{
    Ispis(5);
}

private static void Ispis(int x)
{
    if (x < 2)
    {
        Console.WriteLine(x);
        return;
    }
    else if (x <= 4)
    {
        Ispis(x - 2);
        Console.WriteLine(x);
    }
    else
    {
        Console.WriteLine(x);
        Ispis(x - 2);
    }
    Console.WriteLine(x);
}
```



Rješenje:

Redoslijed ispisa	Ispis	Dodatni komentari
		Poziv metode Ispis(x=5)
		Niti jedan uvjet uvjetnog grananja nije istina pa se izvršava else grana
1	5	Ispis iz else grane metode Ispis(x=5)
		Poziv metode Ispis(x=3)
		Uvjet $x \leq 4$ je istina pa se izvršava else if grana uvjetnog grananja
		Poziv metode Ispis(x=1)
		Uvjet $x < 2$ je istina pa se izvršava if grana uvjetnog grananja
2	1	Ispis iz if grane metode Ispis(x=1)
		Izvršava se naredba return, tj. metoda Ispis(x=1) se završava
		Program se vraća na mjesto gdje je bila pozvana metoda Ispis(x=1), tj. u metodu Ispis(x=3)
3	3	Ispis iz else if grane metode Ispis(x=3)
4	3	Poslijednji Console.WriteLine() u metodi Ispis(x=3)
		Završava se metoda Ispis(x=3) i program se vraća na mjesto gdje je bila pozvana, tj. u metodu Ispis(x=5)
5	5	Poslijednji Console.WriteLine() u metodi Ispis(x=5)
		Završava se metoda Ispis(x=4) i program se vraća na mjesto gdje je bila pozvana, tj. u Main metodu
		Program se završava u Main metodi



Zadatak 2

Što će se ispisati u konzolu nakon izvršavanja sljedećeg programskog odsječka. Potrebno je naznačiti redoslijed ispisivanja. Kao točno rješenje priznaje se samo ono u kojem je vidljiv postupak i sadržaj varijabli u svakoj iteraciji.

```
string tekst = "Programiranje";
int i = 0;

try
{
    for (i = 1; i < 30; i *= 3)
    {
        Console.WriteLine(tekst[i]);
    }
}
catch (Exception ex)
{
    Console.WriteLine(tekst[i / 4]);
}
Console.WriteLine(tekst[i / 3]);
```

Rješenje:

Kako bi mogli lakše pratiti na kojem indeksu se nalazi koje slovo unutar riječi možete si unaprijed napraviti sljedeću tablicu:

Slova	P	r	o	g	r	a	m	i	r	a	n	j	e
Indeks slova	0	1	2	3	4	5	6	7	8	9	10	11	12

Redoslijed ispisa	Ispis	i	Dodatni komentari
		0	Deklaracija i inicijalizacija varijable i
1	r	1	Ulazak u petlju, i se povećava tako da se množi s tri svakim prolaskom kroz petlju
2	g	3	
3	a	9	
		27	Obzirom da je zadnji indeks u tekst=12, i=27 će izazvati grešku
			Obzirom na grešku prestaje kompletan try dio koda i skaćemo u catch dio
4	m		$i / 4 = 27 / 4 = 6$
5	a		$i / 3 = 27 / 3 = 9$



Zadatak 3

Napišite C# kod za učitavanje podataka o cestovnim segmentima u digitalnoj cestovnoj karti RH. Pojedini cestovni segment sadrži informacije o geografskoj dužini i širini početka (x1, y1) i kraja (x2, y2) (realni brojevi) i prosječnoj brzini v (realan broj) na cestovnom segmentu.

Podaci se trebaju učitavati s konzole u listu segmenti, koja treba sadržavati instance klase Segment prikazane u nastavku.

```
public class Segment
{
    public double x1;
    public double y1;
    public double x2;
    public double y2;
    public double v;

    public Segment(double X1, double Y1, double V, double X2, double Y2)
    {
        x1 = X1;
        x2 = X2;
        y1 = Y1;
        y2 = Y2;
        v = V;
    }
}
```

Instancu cestovnog segmenta potrebno je dodati u listu segmenti samo ako je brzina segmenta unutar intervala [0,30> ili [60,80]. Na kraju je u konzolu u proizvoljno formatu potrebno ispisati kolika je prosječna duljina svih valjanih cestovnih segmenata. Za izračun duljine cestovnog segmenta potrebno je koristiti sljedeći izraz:

$$d(T_1(x_1, y_1), T_2(x_2, y_2)) = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} * 111.32$$



Rješenje:

NAPOMENA: obzirom da je osoba koja je slagala ispit kopirala jedan zadatak i radila izmjene za sve grupe ovog roka, u zadatku je napravljena greška. Iako piše da se podaci učitavaju s konzole nikad nije rečeno koliko se segmenata mora učitati. Zbog toga sam napravio da korisnik sam odlučuje o broju unosa segmenata unošenjem varijable unos.

```
static void Main(string[] args)
{
    List<Segment> segmenti = new List<Segment>();
    double sumaDuljinaSegmenata = 0;
    string unos = "";

    do
    {
        Console.Write("Želite li unesti cestovni segment? (DA / NE) ");
        unos = Console.ReadLine();

        if (unos == "DA")
        {
            Console.Write("Unesite geografsku dužinu početka cestovnog segmenta x1: ");
            double x1 = Convert.ToDouble(Console.ReadLine());
            Console.Write("Unesite geografsku širinu početka cestovnog segmenta y1: ");
            double y1 = Convert.ToDouble(Console.ReadLine());

            Console.Write("Unesite geografsku dužinu kraja cestovnog segmenta x2: ");
            double x2 = Convert.ToDouble(Console.ReadLine());
            Console.Write("Unesite geografsku širinu kraja cestovnog segmenta y2: ");
            double y2 = Convert.ToDouble(Console.ReadLine());

            Console.Write("Unesite prosječnu brzinu na cestovnom segmentu v: ");
            double v = Convert.ToDouble(Console.ReadLine());

            Segment s = new Segment(x1, y1, v, x2, y2);

            if ((s.v >= 0 && s.v < 30) || (s.v >= 60 && s.v <= 80))
            {
                segmenti.Add(s);
                sumaDuljinaSegmenata += Math.Sqrt(Math.Pow(s.x2 - s.x1, 2) + Math.Pow(s.y2 - s.y1, 2)) * 111.32;
            }
        }
    } while (unos != "NE");

    double prosjecnaDuljina = sumaDuljinaSegmenata / segmenti.Count();

    Console.WriteLine("Prosječna duljina svih cestovnih segmenata je " + prosjecnaDuljina + " kn.");

    Console.ReadKey();
}
```



Zadatak 4

Napišite C# kod metode MapMatching koja kao argument prima 3 vrijednosti: geografsku dužinu x_1 , geografsku širinu y_1 i listu segmenata segmenti (zadatak 3).

Metoda za zadanu točku $T_1(x_1, y_1)$ treba pronaći najbliži cestovni segment prema kriteriju zadanom izrazom koji se nalazi na kraju ovog zadatka, pri čemu S_i označava prijedeni cestovni segment iz liste, $P_{S_i}(x_p, y_p)$ točku polovišta segmenta S_i određenu koordinatama x_p i y_p (koju je potrebno izračunati), $d(T_1(x_1, y_1), P_{S_i}(x_p, y_p))$ udaljenost između dvije točke danu izrazom iz prethodnog zadatka (zadatak 3).

Na kraju metoda treba u proizvoljnom formatu ispisati sadržaj segmenta koji je najbliži točki $T_1(x_1, y_1)$,

$$F(T_1(x_1, y_1), S_i) = d(T_1(x_1, y_1), P_{S_i}(x_p, y_p))$$

Rješenje:

```
static void MapMatching(double x1, double y1, List<Segment> segmenti)
{
    double najmanjaUdaljenost = double.MaxValue;
    int indeksSegmentaSNajmanjomUdaljenosti = 0;

    for (int i = 0; i < segmenti.Count; i++)
    {
        double xp = (segmenti[i].x1 + segmenti[i].x2) / 2;
        double yp = (segmenti[i].y1 + segmenti[i].y2) / 2;

        double d = Math.Sqrt(Math.Pow(xp - x1, 2) + Math.Pow(yp - y1, 2)) * 111.32;

        if (d < najmanjaUdaljenost)
        {
            najmanjaUdaljenost = d;
            indeksSegmentaSNajmanjomUdaljenosti = i;
        }
    }

    Console.WriteLine("Segment koji je najbliži točki T(" + x1 + ", " + y1
        + ") je segment koji je određen točkama T1("
        + segmenti[indeksSegmentaSNajmanjomUdaljenosti].x1 + ", "
        + segmenti[indeksSegmentaSNajmanjomUdaljenosti].y1 + ") i T2("
        + segmenti[indeksSegmentaSNajmanjomUdaljenosti].x2 + ", "
        + segmenti[indeksSegmentaSNajmanjomUdaljenosti].y2
        + ") te prosječnom brzinom od "
        + segmenti[indeksSegmentaSNajmanjomUdaljenosti].v + " km/h.");
}
```



Zadatak 5

Napišite C# kod koji s konzole učitava tri mase paketa u proizvodnom pogonu u kilogramima m_1 , m_2 i m_3 (realni brojevi).

Potom se na paletu maksimalne zapremnine 100 kg dodaje samo onaj paket koji od trenutno tri učitana ima najmanju masu.

Učitavanje tri po tri paketa potrebno je ponavljati sve dok se najlakši paket može dodati na paletu. U svakom koraku petlje potrebno je u formatu kao u primjeru ispisati kolika je masa paketa dodanog na paletu, te kolike su mase odbačenih paketa.

Primjerice: „Na paletu je stavljen paket mase 0.5 kg, a odbačeni su paketi mase 1.2 i 2.2 kg.“.

Na kraju je u formatu kao u primjeru potrebno ispisati koliko je paketa stavljeno na paletu i kolika je ukupna masa palete.

Primjer ispisa: „Na paletu je stavljeno ukupno 50 paketa, ukupne mase 99.38 kg!“.



Rješenje:

```
static void Main(string[] args)
{
    List<double> masePaketa = new List<double>();
    int brojPrihvacenihPaketa = 0;
    double ukupnaMasa = 0;

    do
    {
        int indeksNajmanjeMase = 0;
        double najmanjaMasa = double.MaxValue;

        for (int i = 0; i < 3; i++)
        {
            Console.Write("Unesite masu " + (i + 1) + ". paketa: ");
            masePaketa.Add(Convert.ToInt32(Console.ReadLine()));

            if (masePaketa[i] < najmanjaMasa)
            {
                najmanjaMasa = masePaketa[i];
                indeksNajmanjeMase = i;
            }
        }

        if ((ukupnaMasa + masePaketa[indeksNajmanjeMase]) <= 100)
        {
            ukupnaMasa += masePaketa[indeksNajmanjeMase];
            brojPrihvacenihPaketa++;

            Console.Write("Na paletu je stavljen paket mase " + najmanjaMasa
                + " kg, a odbačeni su paketi mase ");

            masePaketa.RemoveAt(indeksNajmanjeMase);

            Console.WriteLine(masePaketa[0] + " i " + masePaketa[1] + " kg.");

            masePaketa.Clear();
        }
        else
        {
            Console.WriteLine("Svi paketi su odbačeni jer bi dodavanje paketa
                od " + najmanjaMasa
                + " kg prekoračilo dozvoljenu masu na paleti!");
            break;
        }
    } while (true);

    Console.WriteLine("Na peletu je stavljeno ukupno " + brojPrihvacenihPaketa
        + " paketa, ukupne mase " + ukupnaMasa + " kg!");

    Console.ReadKey();
}
```



Ispit #44 (09.07.20. - grupa C)

Napomena: rok nije kompletan (nedostaju zadaci 3 i 4)

Zadatak 1

Što će se ispisati u konzolu nakon izvršavanja sljedećeg programskog odsječka. Potrebno je naznačiti redoslijed ispisivanja. Kao točno rješenje priznaje se samo ono u kojem je vidljiv postupak i sadržaj varijabli u svakom pozivu metode.

```
static void Main(string[] args)
{
    Ispis(4);
}

private static void Ispis(int x)
{
    Console.WriteLine(x);
    if (x > 5)
    {
        return;
    }
    else if (x > 2)
    {
        Ispis(x + 1);
    }
    else
    {
        Ispis(x + 2);
    }
    Console.WriteLine(x);
}
```



Rješenje:

Redoslijed ispisa	Ispis	Dodatni komentari
		Poziv metode Ispis (x=4)
1	4	Prvi Console.WriteLine() u metodi Ispis(x=4)
		Uvjet $x > 2$ je istina pa se poziva metoda Ispis(x=5)
2	5	Prvi Console.WriteLine() u metodi Ispis(x=5)
		Uvjet $x > 2$ je istina pa se poziva metoda Ispis(x=6)
3	6	Prvi Console.WriteLine() u metodi Ispis(x=6)
		Uvjet $x > 5$ je istina pa se izvršava naredba return, tj. metoda Ispis(x=6) se završava
		Program se vraća na mjesto gdje je bila pozvana metoda Ispis(x=6), tj. u metodu Ispis(x=5)
4	5	Posljednji Console.WriteLine() u metodi Ispis(x=5)
		Završava se metoda Ispis(x=5) i program se vraća na mjesto gdje je bila pozvana, tj. u metodu Ispis(4)
5	4	Posljednji Console.WriteLine() u metodi Ispis(x=4)
		Završava se metoda Ispis(x=4) i program se vraća na mjesto gdje je bila pozvana, tj. u Main metodu
		Program se završava u Main metodi



Zadatak 2

Što će se ispisati u konzolu nakon izvršavanja sljedećeg programskog odsječka. Potrebno je naznačiti redoslijed ispisivanja. Kao točno rješenje priznaje se samo ono u kojem je vidljiv postupak i sadržaj varijabli u svakoj iteraciji.

```
string tekst = "Programiranje";
int i = 0;

try
{
    for (i = tekst.Length - 1; i > 0; i -= 3)
    {
        Console.WriteLine(tekst[i]);
    }
}
catch (Exception ex)
{
    Console.WriteLine(tekst[tekst.Length / 2]);
}
Console.WriteLine(tekst[tekst.Length / 2]);
```

Rješenje:

Kako bi mogli lakše pratiti na kojem indeksu se nalazi koje slovo unutar riječi možete si unaprijed napraviti sljedeću tablicu:

Slova	P	r	o	g	r	a	m	i	r	a	n	j	e
Indeks slova	0	1	2	3	4	5	6	7	8	9	10	11	12

Redoslijed ispisa	Ispis	i	Dodatni komentari
		0	
1	e	12	Ulazak u petlju, i se smanjuje za 3 svakim prolaskom kroz petlju, a text.Length=13
2	a	9	
3	m	6	
4	g	3	
		0	Uvjet ponavljanja petlje (0>0) više nije istinit pa se ne ulazi u petlju
			u try dijelu nije bilo greške pa se catch dio nikada ne pokreće
5	m		Posljednji Console.WriteLine() u metodi, tekst.Length/2=6



Zadatak 5

Napišite C# kod koji s konzole učitava tri mase paketa u proizvodnom pogonu u kilogramima m_1 , m_2 i m_3 (cijeli brojevi).

Potom se na paletu maksimalne zapremnine 500 kg od trenutno tri učitana člana dodaje paket koji je najbliži masi od 5 kg.

Učitavanje tri po tri paketa potrebno je ponavljati sve dok se odabrani paket može dodati na paletu. U svakom koraku petlje potrebno je u formatu kao u primjeru ispisati kolika je masa paketa dodanog na paletu, te kolike su mase odbačenih paketa.

Primjerice: „Na paletu je stavljen paket mase 3 kg, a odbačeni su paketi mase 1 i 2 kg.“.

Na kraju je u formatu kao u primjeru potrebno ispisati kolika je prosječna masa paketa stavljenog na paletu.

Primjer ispisa: „Prosječna masa paketa na paleti iznosi 3.2 kg!“.



Rješenje:

```
static void Main(string[] args)
{
    List<int> masePaketa = new List<int>();
    int brojPrihvacenihPaketa = 0;
    int ukupnaMasa = 0;
    bool paletaJePuna = false;

    do
    {
        int indeksNajblizeg = 0;
        int najmanjaRazlikaMasa = int.MaxValue;

        for (int i = 0; i < 3; i++)
        {
            Console.Write("Unesite masu " + (i + 1) + ". paketa: ");
            masePaketa.Add(Convert.ToInt32(Console.ReadLine()));

            int trenutnaRazlikaMasa = Math.Abs(5 - masePaketa[i]);

            if (trenutnaRazlikaMasa < najmanjaRazlikaMasa)
            {
                najmanjaRazlikaMasa = trenutnaRazlikaMasa;
                indeksNajblizeg = i;
            }
        }

        if ((ukupnaMasa + masePaketa[indeksNajblizeg]) <= 500)
        {
            ukupnaMasa += masePaketa[indeksNajblizeg];
            brojPrihvacenihPaketa++;

            Console.Write("Na paletu je stavljen paket mase "
                + masePaketa[indeksNajblizeg] + " kg, a odbačeni su paketi mase ");

            masePaketa.RemoveAt(indeksNajblizeg);

            Console.WriteLine(masePaketa[0] + " i " + masePaketa[1] + " kg.");

            masePaketa.Clear();
        }
        else
        {
            Console.WriteLine("Svi paketi su odbačeni jer bi dodavanje paketa
                od " + masePaketa[indeksNajblizeg]
                + " kg prekoračilo dozvoljenu masu na paleti!");
            paletaJePuna = true;
        }
    } while (!paletaJePuna);

    double prosjecnaMasa = ukupnaMasa/Convert.ToDouble(brojPrihvacenihPaketa);

    Console.WriteLine("Prosječna masa paketa na paleti iznosi "
        + prosjecnaMasa + " kg!");

    Console.ReadKey();
}
```



Ispit #45 (31.08.20. - grupa A)

Zadatak 1

Što će se ispisati u konzolu nakon izvršavanja sljedećeg programskog odsječka. Potrebno je naznačiti redoslijed ispisivanja. Kao točno rješenje priznaje se samo ono u kojem je vidljiv postupak i sadržaj varijabli u svakoj iteraciji petlje.

```
int i = 1, j = 1;

do
{
    i += j;
    j = (j * 2) % 3;
    Console.WriteLine("(" + i + ", " + j + ")");
} while (i < 10 && (j + i) <= 7);
```



Zadatak 2

Što će se ispisati u konzolu nakon izvršavanja sljedećeg programskog odsječka, ukoliko se s konzole učitava niz „mark“. Potrebno je naznačiti redoslijed ispisivanja. Kao točno rješenje priznaje se samo ono u kojem je vidljiv postupak i sadržaj varijabli u svakoj iteraciji.

```
string tekst = Console.ReadLine();  
for (int i = tekst.Length-1; i >= 0; i--)  
{  
    switch (tekst[i])  
    {  
        case 'm':  
            Console.WriteLine('o');  
            break;  
        case 'a':  
            Console.WriteLine('t');  
            break;  
        case 'r':  
            Console.WriteLine('a');  
            break;  
        case 'k':  
            Console.WriteLine('m');  
            break;  
        default:  
            Console.WriteLine('e');  
            break;  
    }  
}
```



Zadatak 3

Napišite C# kod rekurzivne metode koja računa matematički izraz:

$$R(n) = R(n - 1) + R(n - 2) * R(n - 3), \forall n > 3$$

pri čemu je $R(0) = 1$, $R(1) = 2$, $R(2) = 3$.

Unutar Main metode potrebno je pozvati rekurzivnu metodu s argumentom $n = 6$ te ispisati rezultat.

```
static void Main(string[] args)
{
    Console.WriteLine("R(6) = " + R(6));

    Console.ReadKey();
}

static int R(int n)
{
    if (n == 0)
    {
        return 1;
    }
    else if (n == 1)
    {
        return 2;
    }
    else if (n == 2)
    {
        return 3;
    }

    return R(n - 1) + R(n - 2) * R(n - 3);
}
```



Zadatak 4

Napišite C# kod programa koji služi za simulaciju proizvodne trake u pogonu. S konzole je potrebno učitavati volumene dvaju paketa (cijeli brojevi) v_1 i v_2 sve dok je suma volumena tih dvaju paketa manja od 20.

U svakoj iteraciji učitavanja potrebno je volumene oba paketa zapisati u datoteku volumeni.txt.

Na kraju je u proizvoljnom formatu potrebno ispisati koliko je bilo ukupno paketa.

Nije potrebno koristiti try-catch blok te nije potrebno dodati imenski prostor System.IO.

```
using System.IO;

static void Main(string[] args)
{
    StreamWriter writer = new StreamWriter("volumeni.txt");

    int sumaVolumena = 0;
    int brojPrihvacenihPaketa = 0;

    do
    {
        Console.Write("Unesite volumen paketa v1: ");
        int v1 = Convert.ToInt32(Console.ReadLine());

        Console.Write("Unesite volumen paketa v2: ");
        int v2 = Convert.ToInt32(Console.ReadLine());

        sumaVolumena = v1 + v2;

        if (sumaVolumena < 20)
        {
            writer.WriteLine(v1 + "," + v2);
            brojPrihvacenihPaketa += 2;
        }
        else
        {
            Console.WriteLine("Volumen paketa je veći od 20 pa se paketi neće prihvatiti!");
        }
    } while (sumaVolumena < 20);

    writer.Flush();
    writer.Close();

    Console.WriteLine("Ukupan broj prihvaćenih paketa je "
        + brojPrihvacenihPaketa + ".");

    Console.ReadKey();
}
```



Zadatak 5

Napišite C# kod programa koji učitava cjelobrojnu vrijednost n koja predstavlja broj gradova. Potrebno je kreirati matricu $M = n \times n$ koja predstavlja udaljenosti između gradova i popuniti ju prema sljedećim pravilima:

- udaljenost između jednog te istog grada treba imati vrijednost nula (0) (pomoć: vrijednost glavne dijagonale matrice trebaju biti nula) i
- u svim ostalim slučajevima upisati slučajan realan broj unutar intervala $[0, 10>$,

Na kraju je sadržaj matrice potrebno ispisati u formatu kao u primjeru.

Neka je $n = 4$, a ispisana matrica popunjena na adekvatan način treba izgledati (vrijednosti su slučajne):

0	9,37	4,38	7,56
8,35	0	2,61	2,22
3,78	2,61	0	1,25
7,78	2,38	1,78	0

```
static void Main(string[] args)
{
    Console.Write("Unesite željeni broj gradova: ");
    int n = Convert.ToInt32(Console.ReadLine());

    double[,] M = new double[n, n];

    Random rnd = new Random();

    for (int i = 0; i < M.GetLength(0); i++)
    {
        for (int j = 0; j < M.GetLength(1); j++)
        {
            if (i == j)
            {
                M[i, j] = 0;
            }
            else
            {
                M[i, j] = rnd.NextDouble() * 10;
            }

            Console.Write(Math.Round(M[i, j], 2) + "\t");
        }
        Console.WriteLine();
    }

    Console.ReadKey();
}
```



Ispit #46 (09.09.20. - grupa A)

Zadatak 1

Što će se ispisati u konzolu nakon izvršavanja sljedećeg programskog odsječka. Potrebno je naznačiti redoslijed ispisivanja. Kao točno rješenje priznaje se samo ono u kojem je vidljiv postupak i sadržaj varijabli.

```
DateTime t1 = DateTime.Now;
Thread.Sleep(200);
DateTime t2 = DateTime.Now;
Thread.Sleep(300);
Console.WriteLine((DateTime.Now - t1).TotalSeconds);
TimeSpan ts = t2 - t1;
Console.WriteLine(ts.TotalMilliseconds);
int godina = t1.Year;
int mjesec = t2.Month;
Console.WriteLine((godina + 1) / (mjesec + 1));
Console.WriteLine((godina + 1) % (mjesec + 1));
```



Zadatak 2

Što će se ispisati u konzolu nakon izvršavanja sljedećeg programskog odsječka. Potrebno je naznačiti redoslijed ispisivanja. Kao točno rješenje priznaje se samo ono u kojem je vidljiv postupak i sadržaj varijabli u svakoj iteraciji.

```
int[] p = new int[7];
for (int i = 0; i < p.Length - 1; i++)
{
    p[i + 1] = (i - 1) * 2;
}
foreach (int v in p)
{
    if (v < p.Length / 2)
    {
        Console.WriteLine(v);
    }
}
```



Zadatak 3

Napišite C# kod rekurzivne metode naziva `Rekurzija` koja kao argument prima cijeli broj `n` te ispisuje sve neparne brojeve manje ili jednake `n`.

```
static void Rekurzija(int n)
{
    if (n > 1)
    {
        Rekurzija(n - 1);
    }
    if (n % 2 == 1)
    {
        Console.WriteLine(n);
    }
}
```



Zadatak 4

Napišite C# kod programa koji služi za simulaciju duljina vozila u softverskom alatu. S konzole je potrebno učitati niz znakova T koji predstavljaju za koji se tip vozila provodi simulacija (auto, kombi, bus ili sve – prema sljedećoj tablici).

Tip vozila T	Donja granica dg [km/h]	Gornja granica gg [km/h]	Trajanje t [s]
auto	0	5	0,4
kombi	6	10	2
bus	10	20	4
sve	0	20	10

Nakon što se učitava tip vozila potrebno je napraviti simulaciju u trajanju određenom stupcem Trajanje t [s]. Pretpostavka je da jedna iteracija petlje (bilo koje) traje 2 ms (milisekunda).

Duljine unutar simulacije potrebno je upisivati u datoteku naziva `duljine_tip_vozila.txt`, gdje umjesto „tip_vozila“ (u imenu datoteke) treba upisati odgovarajući naziv tipa vozila T. Unutar svake iteracije potrebno je dohvatiti slučajnu duljinu (cijeli broj) unutar intervala [dg, gg] za odgovarajući tip vozila i zapisati ju u datoteku.



Rješenje:

```
using System.IO;

static void Main(string[] args)
{
    Console.Write("Unesite željeni tip vozila: ");
    string T = Console.ReadLine();

    int dg = 0;
    int gg = 20;
    int trajanje = 10000;

    if (T == "auto")
    {
        dg = 0;
        gg = 5;
        trajanje = 400;
    }
    else if (T == "kombi")
    {
        dg = 6;
        gg = 10;
        trajanje = 2000;
    }
    else if (T == "bus")
    {
        dg = 10;
        gg = 20;
        trajanje = 4000;
    }

    trajanje /= 2;

    StreamWriter writer = new StreamWriter("duljine_" + T + ".txt");
    Random rnd = new Random();

    for (int i = 0; i < trajanje; i++)
    {
        writer.WriteLine(rnd.Next(dg, gg + 1));
    }

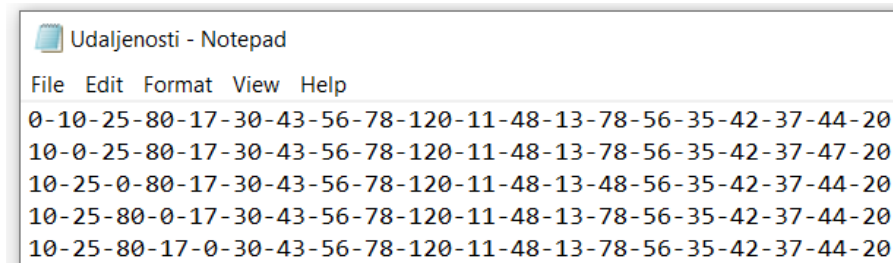
    writer.Close();

    Console.ReadKey();
}
```



Zadatak 5

Napišite C# kod programa koji učitava sadržaj datoteke Udaljenost.txt u matricu $M = 20 \times 20$. U datoteci je zapisano 400 vrijednosti udaljenosti (cijeli brojevi), po 20 u svakom retku, međusobno odvojenim znakom – (crtica) kao što je prikazano na sljedećoj slici.



Način punjena matrice je proizvoljan, ali je potrebno svih 400 vrijednosti zapisati u matricu.

Na kraju je potrebno ispisati ukupnu sumu udaljenosti u matrici u proizvoljnom formatu i napraviti ispis elemenata na glavnoj dijagonali matrice.

```
using System.IO;

static void Main(string[] args)
{
    StreamReader reader = new StreamReader("Udaljenosti.txt");

    int[,] M = new int[20, 20];
    int sumaUdaljenosti = 0;

    Console.WriteLine("Elementi na glavnoj dijagonali matrice:");

    for (int i = 0; i < M.GetLength(0); i++)
    {
        string linija = reader.ReadLine();
        string[] dijeloviLinije = linija.Split('-');

        for (int j = 0; j < M.GetLength(1); j++)
        {
            M[i, j] = Convert.ToInt32(dijeloviLinije[j]);
            sumaUdaljenosti += M[i, j];

            if (i == j)
            {
                Console.WriteLine(M[i, j]);
            }
        }
    }

    reader.Close();

    Console.WriteLine("Suma udaljenosti u matrici je " + sumaUdaljenosti + ".");

    Console.ReadKey();
}
```



Ispit #47 (09.09.20. - grupa B)

Zadatak 1

Što će se ispisati u konzolu nakon izvršavanja sljedećeg programskog odsječka. Potrebno je naznačiti redoslijed ispisivanja. Kao točno rješenje priznaje se samo ono u kojem je vidljiv postupak i sadržaj varijabli.

```
DateTime t1 = DateTime.Now;
Thread.Sleep(500);
DateTime t2 = DateTime.Now;
Thread.Sleep(100);
TimeSpan ts = t2 - t1;
Console.WriteLine(ts.TotalSeconds);
Console.WriteLine((t2 - t1).TotalMilliseconds);
int dan = t1.Day;
int mjesec = t2.Month;
Console.WriteLine((dan - 2) % mjesec);
Console.WriteLine((dan - 2) / mjesec);
```



Zadatak 2

Što će se ispisati u konzolu nakon izvršavanja sljedećeg programskog odsječka. Potrebno je naznačiti redoslijed ispisivanja. Kao točno rješenje priznaje se samo ono u kojem je vidljiv postupak i sadržaj varijabli u svakoj iteraciji.

```
int[] p = new int[6];
for (int i = 1; i < p.Length; i++)
{
    p[i - 1] = i / 2;
}
foreach (int v in p)
{
    if (v < p.Length / 2)
    {
        Console.WriteLine(v);
    }
}
```



Zadatak 3

Napišite C# kod rekurzivne metode naziva `Rekurzija` koja kao argument prima cijeli broj `n` te ispisuje sve parne brojeve manje ili jednake `n`, uključujući i 0.

```
static void Rekurzija(int n)
{
    if (n > 0)
    {
        Rekurzija(n - 1);
    }
    if (n % 2 == 0)
    {
        Console.WriteLine(n);
    }
}
```



Zadatak 4

Napišite C# kod programa koji služi za simulaciju brzina u softverskom alatu. S konzole je potrebno učitati niz znakova P koji predstavljaju za koje se područje provodi simulacija (autocesta, ulica, avenija ili sve – prema sljedećoj tablici).

Područje P	Donja granica dg [km/h]	Gornja granica gg [km/h]	Trajanje t [s]
ulica	0	40	0,5
avenija	40	90	1
autocesta	90	150	2
sve	0	150	5

Nakon što se učitaju područja potrebno je napraviti simulaciju u trajanju određenom stupcem Trajanje t [s]. Pretpostavka je da jedna iteracija petlje (bilo koje) traje 1 ms (milisekunda).

Brzine unutar simulacije potrebno je upisivati u datoteku naziva brzine_područje.txt, gdje se umjesto područja (u imenu datoteke) treba upisati odgovarajući naziv područja P. Unutar svake iteracije potrebno je dohvatiti slučajnu brzinu (cijeli broj) unutar intervala <dg, gg> za odgovarajuće područje i zapisati ju u datoteku.



Rješenje:

```
using System.IO;

static void Main(string[] args)
{
    Console.Write("Unesite željeno područje simulacije: ");
    string P = Console.ReadLine();

    int dg = 0;
    int gg = 150;
    int trajanje = 5000;

    if (P == "ulica")
    {
        dg = 0;
        gg = 40;
        trajanje = 500;
    }
    else if (P == "avenija")
    {
        dg = 40;
        gg = 90;
        trajanje = 1000;
    }
    else if (P == "autocesta")
    {
        dg = 90;
        gg = 150;
        trajanje = 2000;
    }

    StreamWriter writer = new StreamWriter("brzine_" + P + ".txt");
    Random rnd = new Random();

    for (int i = 0; i < trajanje; i++)
    {
        writer.WriteLine(rnd.Next(dg + 1, gg));
    }

    writer.Close();

    Console.ReadKey();
}
```



Zadatak 5

Napišite C# kod programa koji učitava sadržaj datoteke udaljenosti.txt preko relativne putanje u matricu $M = 10 \times 10$. U datoteci je zapisano 100 udaljenosti (realni brojevi) svaka u zasebnom retku. Način punjenja matrice je proizvoljan, ali je potrebno svih 100 vrijednosti zapisati u matricu.

Na kraju je potrebno ispisati prosječnu vrijednost udaljenosti u matrici u formatu kao u primjeru i napraviti ispis matrice stupac po stupac (proizvoljan ispis pojedinih elemenata).

Primjer ispisa prosječne vrijednosti: „Prosječna vrijednost udaljenosti u matrici $M = 10 \times 10$ iznosi 9.87!!!“, pri čemu je 9.87 izračunata vrijednost i treba ju zamijeniti s vrijednošću odgovarajuće varijable u Vašem kodu.

```
using System.IO;

static void Main(string[] args)
{
    double[,] M = new double[10, 10];

    StreamReader reader = new StreamReader("udaljenosti.txt");

    double sumaUdaljenosti = 0.0;

    Console.WriteLine("Izgled matrice: ");

    for (int i = 0; i < M.GetLength(0); i++)
    {
        for (int j = 0; j < M.GetLength(1); j++)
        {
            M[i, j] = Convert.ToDouble(reader.ReadLine());

            sumaUdaljenosti += M[i, j];

            Console.Write(M[i, j] + "\t");
        }
        Console.WriteLine();
    }

    double prosjekUdaljenosti = sumaUdaljenosti / M.Length;

    Console.WriteLine("Prosječna vrijednost udaljenosti u matrici  $M = 10 \times 10$  iznosi " + prosjekUdaljenosti + "!!!");

    Console.ReadKey();
}
```



Ispit #48 (21.11.20. - grupa B)

Zadatak 1

Što će se ispisati u konzolu nakon izvršavanja sljedećeg programskog odsječka, ukoliko se s konzole učitava niz „marKA“. Potrebno je naznačiti redoslijed ispisivanja. Kao točno rješenje priznaje se samo ono u kojem je vidljiv postupak i sadržaj varijabli u svakoj iteraciji.

```
string tekst = Console.ReadLine();
for (int i = tekst.Length - 2; i > 0; i--)
{
    switch (tekst[i])
    {
        case 'a':
            Console.WriteLine('r');
            break;
        case 'm':
            Console.WriteLine('k');
            break;
        case 'r':
            Console.WriteLine('a');
            break;
        case 'k':
            Console.WriteLine('m');
            break;
        default:
            Console.WriteLine('l');
            break;
    }
}
```



Zadatak 2

Napišite C# kod rekurzivne metode koja računa matematički izraz:

$$R(n) = R(n - 3) + R(n - 1) * R(n - 2), \forall n \geq 3$$

pri čemu je $R(0) = 1$, $R(1) = 1$, $R(2) = -1$.

Unutar Main metode potrebno je pozvati rekurzivnu metodu s argumentom $n = 6$ te ispisati rezultat.

```
static void Main(string[] args)
{
    Console.WriteLine("R(6) = " + R(6));

    Console.ReadKey();
}

static int R(int n)
{
    if (n == 0)
    {
        return 1;
    }
    else if (n == 1)
    {
        return 1;
    }
    else if (n == 2)
    {
        return -1;
    }

    return R(n - 3) + R(n - 1) * R(n - 2);
}
```



Zadatak 3

Što će se ispisati u konzolu nakon izvršavanja sljedećeg programskog odsječka. Potrebno je naznačiti redoslijed ispisivanja. Kao točno rješenje priznaje se samo ono u kojem je vidljiv postupak i sadržaj varijabli u svakoj iteraciji.

```
List<int> lista = new List<int>();
lista.AddRange(new int[] { -1, 0, 1, 2 });
lista.RemoveAt(lista.Count - 3);
lista.Insert(lista.Count - 3, 2);
lista.Add(-1);
foreach (int v in lista)
{
    if (v != 1)
    {
        Console.WriteLine(v);
    }
}
```



Zadatak 4

Napišite kod programa koji služi za simulaciju brzina u softverskom alatu. S konzole je potrebno učitati niz znakova P koji predstavljaju za koje se područje provodi simulacija (autocesta, ulica, avenija ili sve – prema sljedećoj tablici).

Područje P	Donja granica dg [km/h]	Gornja granica gg [km/h]	Trajanje t [s]
ulica	0	40	0,5
avenija	40	90	1
autocesta	90	150	2
sve	0	150	5

Nakon što se učitaju područja potrebno je napraviti simulaciju u trajanju određenom stupcem Trajanje t [s]. Pretpostavka je da jedna iteracija petlje (bilo koje) traje 1 ms (milisekunda).

Brzine unutar simulacije potrebno je upisivati u datoteku naziva brzine_područje.txt, gdje se umjesto područja (u imenu datoteke) treba upisati odgovarajući naziv područja P. Unutar svake iteracije potrebno je dohvatiti slučajnu brzinu (cijeli broj) unutar intervala <dg, gg> za odgovarajuće područje i zapisati ju u datoteku.



Rješenje:

```
using System.IO;

static void Main(string[] args)
{
    Console.WriteLine("Unesite željeno područje simulacije: ");
    string P = Console.ReadLine();

    int dg = 0;
    int gg = 150;
    int trajanje = 5000;

    if (P == "ulica")
    {
        dg = 0;
        gg = 40;
        trajanje = 500;
    }
    else if (P == "avenija")
    {
        dg = 40;
        gg = 90;
        trajanje = 1000;
    }
    else if (P == "autocesta")
    {
        dg = 90;
        gg = 150;
        trajanje = 2000;
    }

    StreamWriter writer = new StreamWriter("brzine_" + P + ".txt");
    Random rnd = new Random();

    for (int i = 0; i < trajanje; i++)
    {
        writer.WriteLine(rnd.Next(dg + 1, gg));
    }

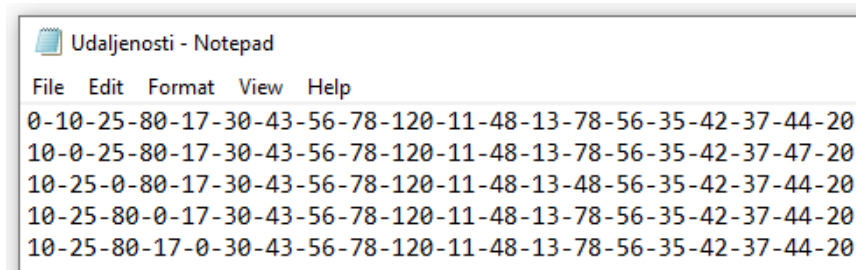
    writer.Flush();
    writer.Close();

    Console.ReadKey();
}
```



Zadatak 5

Napišite C# kod programa koji učitava sadržaj datoteke Udaljenosti.txt u matricu $M = 20 \times 20$. U datoteci je zapisano 400 vrijednosti udaljenosti (cijeli brojevi), po 20 u svakom retku, međusobno odvojenih znakom – (crtica) kao što je prikazano na sljedećoj slici.



Način punjenja matrice je proizvoljan, ali je potrebno svih 400 vrijednosti zapisati u matricu.

Na kraju je potrebno ispisati ukupnu sumu udaljenosti u matrici u proizvoljnom formatu i napraviti ispis elemenata na glavnoj dijagonali matrice.

```
static void Main(string[] args)
{
    StreamReader reader = new StreamReader("Udaljenosti.txt");

    int[,] M = new int[20, 20];
    int sumaUdaljenosti = 0;

    Console.WriteLine("Elementi na glavnoj dijagonali matrice:");

    for (int i = 0; i < M.GetLength(0); i++)
    {
        string linija = reader.ReadLine();
        string[] dijeloviLinije = linija.Split('-');

        for (int j = 0; j < M.GetLength(1); j++)
        {
            M[i, j] = Convert.ToInt32(dijeloviLinije[j]);
            sumaUdaljenosti += M[i, j];

            if (i == j)
            {
                Console.WriteLine(M[i, j]);
            }
        }
    }

    reader.Close();

    Console.WriteLine("Suma udaljenosti u matrici je " + sumaUdaljenosti + ".");

    Console.ReadKey();
}
```



Ispit #49 (04.02.21. - grupa A)

Zadatak 1

Što će se ispisati u konzolu nakon izvršavanja sljedećeg programskog odsječka. Potrebno je naznačiti redoslijed ispisivanja. Kao točno rješenje priznaje se samo ono u kojem je vidljiv postupak i sadržaj varijabli u svakom pozivu metoda.

```
static void Main(string[] args)
{
    Console.WriteLine(metoda(8));
}
static int metoda(int x)
{
    if (x == 0)
    {
        return 1;
    }
    else if (x == 5)
    {
        return x * metoda(x - 5);
    }
    else if (x < 5)
    {
        return x - metoda(x + 1);
    }
    else if (x > 3)
    {
        return x + metoda(x / 2);
    }
    else
    {
        return metoda(x - 1);
    }
}
```



Zadatak 2

Što će se ispisati u konzolu nakon izvršavanja sljedećeg programskog odsječka. Potrebno je naznačiti redoslijed ispisivanja. Kao točno rješenje priznaje se samo ono u kojem je vidljiv postupak i sadržaj varijabli u svakoj iteraciji.

```
List<int> lista = new List<int>();  
lista.Add(0);  
lista.AddRange(new int[] { -2, 0, 3 });  
lista.Remove(0);  
lista.RemoveAt(lista.Count - 3);  
for (int i = 0; i < lista.Count; i++)  
{  
    Console.WriteLine(lista[i]);  
}
```



Zadatak 3

Napišite C# kod za simuliranje i obradu podataka o brzinama dobivenih s radara preko konzole. S konzole je potrebno učitati broj N koji predstavlja broj simuliranih brzina te popuniti jednodimenzionalno polje od N elemenata sa slučajnim vrijednostima unutar intervala $[0, 150>$.

Koristeći sortirano polje potrebno je implementirati filter koji će sve podatke koji nisu unutar intervala $[Q_1, Q_3]$ smatrati outlierima.

Indeksi kvartalnih granica u sortiranom polju računaju se kao $I_{Q_1} = 0.25 * N$ i $I_{Q_3} = 0.75 * N$, pri čemu je indeks potrebno zaokružiti na najbliži cijeli broj.

Na kraju je potrebno ispisati iznos dolje kvartalne granice Q_1 , gornje kvartalne granice Q_3 i broj outliera u sljedećem formatu: „Donja kvartalna granica iznosi Q_1 , a gornja kvartalna granica iznosi Q_3 ! Broj outliera: X “.

Deklaracija metode InsertionSort dana je u nastavku i nije ju potrebno implementirati već samo iskoristiti.

```
private static void InsertionSort(double[] polje)
```



Rješenje:

```
static void Main(string[] args)
{
    Random rnd = new Random();

    Console.WriteLine("Unesite željeni broj simuliranih brzina N: ");
    int N = Convert.ToInt32(Console.ReadLine());

    double[] brzine = new double[N];

    for (int i = 0; i < brzine.Length; i++)
    {
        brzine[i] = rnd.NextDouble() * 150;
    }

    InsertionSort(brzine);

    int IQ1 = Convert.ToInt32(Math.Round(0.25 * N));
    int IQ3 = Convert.ToInt32(Math.Round(0.75 * N));

    int brojOutliera = IQ1 + (brzine.Length - 1 - IQ3);

    //drugi način računanja outliera
    //int brojOutliera = 0;

    //for (int i = 0; i < brzine.Length; i++)
    //{
    //    if (i < IQ1 || i > IQ3)
    //    {
    //        brojOutliera++;
    //    }
    //}

    Console.WriteLine("Donja granica iznosi: " + brzine[IQ1]
        + ", a gornja granica iznosi " + brzine[IQ3]
        + "! Broj outliera: " + brojOutliera);

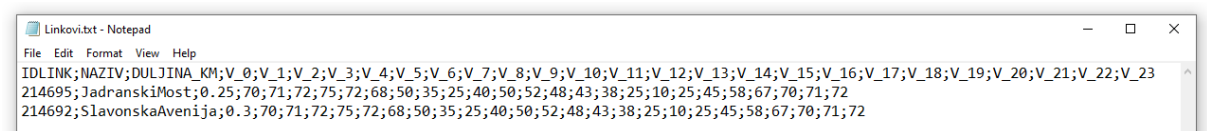
    Console.ReadKey();
}
```



Zadatak 4

Napišite C# kod za izračun vremena putovanja rute zadanog poljem ID-eva linkova (cestovnih segmenata) IDLinkovi i pripadajućih indeksa sata početka putovanja na linku zadanih poljem vrijemeIndeks, pri čemu indeks nula označava satni interval 00:00-01:00, a indeks 23 satni interval 23:00-00:00.

U datoteci Linkovi.txt prikazanoj na slici nalaze se svi linkovi na području Zagreba s podacima o ID-u linka (IDLINK), nazivu linka (NAZIV), duljini linka u kilometrima (DULJINA_KM) i brzini u km/h u određenom satnom intervalu (V_0 označava brzinu u satu 00:00-01:00, a V_23 u satu 23:00-00:00).



Potrebno je koristiti try-catch blok te nije potrebno dodati imenski prostor System.IO.

```
int[] IDLinkovi = new int[] { 214692, 214693, ..., 21717, 214695 };  
int[] vrijemeIndeks = new int[] { 0, 5, ..., 23, 10 };
```



Rješenje:

```
using System.IO;

static void Main(string[] args)
{
    int[] IDLinkovi = new int[] { 214692, 214693, ..., 21717, 214695 };
    int[] vrijemeIndeks = new int[] { 0, 5, ..., 23, 10 };

    try
    {
        StreamReader reader = new StreamReader("Linkovi.txt");
        string linija = reader.ReadLine();
        double vrijemePutovanja = 0.0;

        while (!reader.EndOfStream)
        {
            linija = reader.ReadLine();
            string[] dijeloviLinije = linija.Split(';');

            int IDLINK = Convert.ToInt32(dijeloviLinije[0]);

            for (int i = 0; i < IDLinkovi.Length; i++)
            {
                if (IDLinkovi[i] == IDLINK)
                {
                    double DULJINA_KM = Convert.ToDouble(dijeloviLinije[2]);
                    int brzina =
                        Convert.ToInt32(dijeloviLinije[vrijemeIndeks[i] + 3]);
                    vrijemePutovanja += DULJINA_KM / brzina;
                }
            }

            reader.Close();

            Console.WriteLine("Vrijeme putovanja rute je "
                + vrijemePutovanja + "h.");
        }
        catch (Exception ex)
        {
            Console.WriteLine("Greška: " + ex.Message);
        }

        Console.ReadKey();
    }
}
```

Napomena: ako ćete provjeravati zadatak moguće je da će Vam zadatak izračunati krivo vrijeme putovanja. Do problema dolazi zbog formata brojeva u Windowsima. Hrvatski format ignorira točku unutar broja, pa je za decimalni broj potrebno staviti zarez umjesto točke koja je napisana u datoteci Linkovi.txt. Tako će Vam za duljinu od 0.25km biti učitana vrijednost od 25km. Problem se može popraviti naredbom Replace prije konverzije učitano podatka u decimalni broj. Primjer popravljivanja rješenja:

Red:

```
double DULJINA_KM = Convert.ToDouble(dijeloviLinije[2]);
```

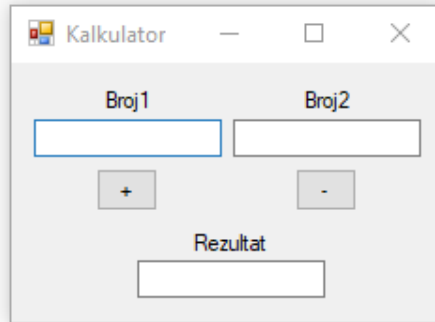
Zamijeniti redom:

```
double DULJINA_KM = Convert.ToDouble(dijeloviLinije[2].Replace('.', ','))
```



Zadatak 5

Nadopunite C# kod Windows forme koja predstavlja kalkulator prikazan na slici koji radi zbrajanje i oduzimanje dvaju realnih brojeva.



Kontrole koje se koriste su: txtBroj1, txtBroj2, txtRezultat, labBroj1, labBroj2, labRezultat, btnPlus i btnMinus.

```
public partial class Form1 : Form
{
    public Form1()
    {
        InitializeComponent();
    }

    private void btnPlus_Click(object sender, EventArgs e)
    {
        //Nadopuni - 1
    }

    private void btnMinus_Click(object sender, EventArgs e)
    {
        //Nadopuni - 2
    }
}
```



Rješenje:

```
public partial class Form1 : Form
{
    public Form1()
    {
        InitializeComponent();
    }

    private void btnPlus_Click(object sender, EventArgs e)
    {
        //Nadopuni - 1
        double broj1 = Convert.ToDouble(txtBroj1.Text);
        double broj2 = Convert.ToDouble(txtBroj2.Text);

        double rezultat = broj1 + broj2;

        txtRezultat.Text = rezultat.ToString();
    }

    private void btnMinus_Click(object sender, EventArgs e)
    {
        //Nadopuni - 2
        double broj1 = Convert.ToDouble(txtBroj1.Text);
        double broj2 = Convert.ToDouble(txtBroj2.Text);

        double rezultat = broj1 - broj2;

        txtRezultat.Text = rezultat.ToString();
    }
}
```



Ispit #50 (18.02.21. - grupa A)

Zadatak 1

Što će se ispisati u konzolu nakon izvršavanja sljedećeg programskog odsječka. Potrebno je naznačiti redoslijed ispisivanja. Kao točno rješenje priznaje se samo ono u kojem je vidljiv postupak i sadržaj varijabli u svakom pozivu metoda.

```
static void Main(string[] args)
{
    Console.WriteLine(metoda(9));
}
static int metoda(int x)
{
    if (x == 0)
    {
        return 1;
    }
    else if (x == 5)
    {
        return x * metoda(x - 5);
    }
    else if (x < 5)
    {
        return x - metoda(x + 1);
    }
    else if (x > 3)
    {
        return x + metoda(x / 2);
    }
    else
    {
        return metoda(x - 1);
    }
}
```



Zadatak 2

Što će se ispisati u konzolu nakon izvršavanja sljedećeg programskog odsječka. Potrebno je naznačiti redoslijed ispisivanja. Kao točno rješenje priznaje se samo ono u kojem je vidljiv postupak i sadržaj varijabli u svakoj iteraciji.

```
int a = 3;
int b = 10;
int c, d;
if (b / 3 > 3)
{
    c = a % 2;
}
else
{
    c = b + 2 * a;
}
bool e = c % 4 == 0;
if (!(e ^ true))
{
    d = Convert.ToInt32(Math.Round(Math.Sqrt(c)));
}
else
{
    d = Convert.ToInt32(Math.Round(Math.Pow(c, 2)));
}
Console.WriteLine(a + "," + b + "," + c + "," + d);
```



Zadatak 3

Napišite C# kod za simuliranje i obradu podataka o brzinama dobivenih s radara preko konzole. S konzole je potrebno učitati broj N koji predstavlja broj simuliranih brzina te popuniti jednodimenzionalno polje od N elemenata sa slučajnim vrijednostima unutar intervala $[0, 200>$.

Koristeći sortirano polje potrebno je implementirati filter koji će sve podatke koji nisu unutar intervala $[Q_1, Q_3]$ smatrati outlierima.

Indeksi granica u sortiranom polju računaju se kao $I_{Q_1} = 0.15 * N$ i $I_{Q_3} = 0.85 * N$, pri čemu je indeks potrebno zaokružiti na najbliži cijeli broj.

Na kraju je potrebno ispisati iznos dolje granice Q_1 , gornje granice Q_3 i broj outliera u sljedećem formatu: „Donja granica iznosi Q_1 , a gornja granica iznosi Q_3 ! Broj outliera: X “.

Deklaracija metode BubbleSort dana je u nastavku i nije ju potrebno implementirati već samo iskoristiti.

```
private static void BubbleSort(double[] polje)
```



Rješenje:

```
static void Main(string[] args)
{
    Random rnd = new Random();

    Console.WriteLine("Unesite željeni broj simuliranih brzina N: ");
    int N = Convert.ToInt32(Console.ReadLine());

    double[] brzine = new double[N];

    for (int i = 0; i < brzine.Length; i++)
    {
        brzine[i] = rnd.NextDouble() * 200;
    }

    BubbleSort(brzine);

    int IQ1 = Convert.ToInt32(Math.Round(0.15 * N));
    int IQ3 = Convert.ToInt32(Math.Round(0.85 * N));

    int brojOutliera = IQ1 + (brzine.Length - 1 - IQ3);

    Console.WriteLine("Donja granica iznosi: " + brzine[IQ1]
        + ", a gornja granica iznosi " + brzine[IQ3]
        + "! Broj outliera: " + brojOutliera);

    Console.ReadKey();
}
```



Zadatak 4

Napišite C# kod za učitavanje ID-a linka i indeksa sata s tipkovnice sve dok se za ID linka ne učitava broj manji ili jednak nuli.

U programu je već inicijalizirana lista linkovi s instancama klase LINK, te listu nije potrebno popuniti već samo iskoristiti. Klasa LINK dostupna je u nastavku, te sadrži attribute ID, NAZIV, DULJINA_KM, te polje BRZINE_U_SATU koje sadrži 24 zapisa brzina u km/h, pri čemu indeks nula (0) označava satni interval 00:00 – 01:00, a indeks 23 satni interval 23:00 – 00:00.

Za svaki učitani ID linka potrebno je pronaći odgovarajući instancu klase LINK u listi linkovi i izračunati vrijeme putovanja na linku ovisno o učitanoj satnom indeksu.

Dodatno je u datoteku linkoviRute.txt koja se treba nalaziti na putanji D:\Podaci za svaki učitani ID linka potrebno zapisati attribute linka u formatu:

ID;NAZIV;DULJINA_KM;SATNI_INDEKS;VRIJEME_PUTOVANJA_U_SATNOM_INDEKSU

Potrebno je koristiti try-catch blok te nije potrebno dodati imenski prostor System.IO.

```
class LINK
{
    public int ID;
    public string NAZIV;
    public int DULJINA_KM;
    public double[] BRZINE_U_SATU;
}
static void Main(string[] args)
{
    List<LINK> linkovi = inicijalizirana lista linkova;
}
```



Rješenje:

```
static void Main(string[] args)
{
    List<LINK> linkovi = inicijalizirana lista linkova;

    try
    {
        StreamWriter writer = new StreamWriter("D:\\Podaci\\linkoviRute.txt");

        int ID = 0;

        do
        {
            Console.Write("Unesite ID linka: ");
            ID = Convert.ToInt32(Console.ReadLine());

            if (ID > 0)
            {
                Console.Write("Unesite indeks sata: ");
                int satniIndeks = Convert.ToInt32(Console.ReadLine());

                int indeksElementa = -1;

                for (int i = 0; i < linkovi.Count; i++)
                {
                    if (linkovi[i].ID == ID)
                    {
                        indeksElementa = i;
                        break;
                    }
                }

                double vrijemePutovanja = linkovi[indeksElementa].DULJINA_KM /
                    linkovi[indeksElementa].BRZINE_U_SATU[satniIndeks];

                writer.WriteLine(linkovi[indeksElementa].ID + ";"
                    + linkovi[indeksElementa].NAZIV + ";"
                    + linkovi[indeksElementa].DULJINA_KM + ";"
                    + satniIndeks + ";" + vrijemePutovanja);
            }
            else
            {
                Console.WriteLine("Unesen je pogrešan ID linka i učitavanje se
                    prekida!");
            }
        } while (ID > 0);

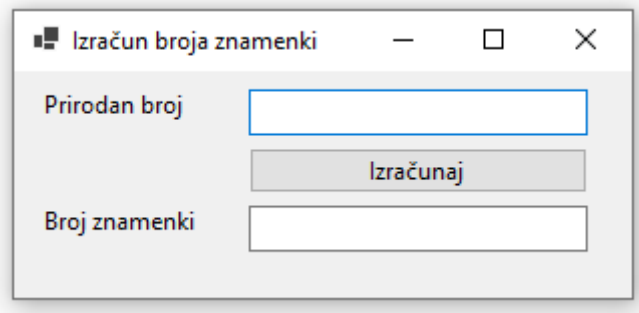
        writer.Flush();
        writer.Close();
    }
    catch (Exception ex)
    {
        Console.WriteLine("Greška: " + ex.Message);
    }

    Console.ReadKey();
}
```



Zadatak 5

Nadopunite C# kod Windows forme na slici koja izračunava koliko znamenaka ima uneseni prirodni broj.



Kontrole koje se koriste su: txtBroj (TextBox), txtBrojZnamenki (TextBox), labBroj (Label), labBrojZnamenki (Label) i btnIzracunaj (Button). Potrebno je napisati samo dijelove koda ispod komentara „Nadopuni“.

```
public Form1()
{
    InitializeComponent();
}
private void btnIzracunaj_Click(object sender, EventArgs e)
{
    //Nadopuni
}
```

Rješenje:

```
public Form1()
{
    InitializeComponent();
}
private void btnIzracunaj_Click(object sender, EventArgs e)
{
    //Nadopuni

    int broj = Convert.ToInt32(txtBroj.Text);

    int brojZnamenki = 0;

    while (broj > 0)
    {
        broj /= 10;
        brojZnamenki++;
    }

    txtBrojZnamenki.Text = brojZnamenki.ToString();
}
```



Ispit #51 (17.04.21. - grupa A)

Napomena: ispit je u potpunosti isti kao i kolokvij koji je pisan 13.04.2021. (grupa A).

Zadatak 1

Što će se ispisati u konzolu nakon izvršavanja sljedećeg programskog odsječka, ukoliko se s konzole učitava tekst ALGPRO. Potrebno je naznačiti redoslijed ispisivanja. Kao točno rješenje priznaje se samo ono u kojem je vidljiv postupak i sadržaj varijabli u svakoj iteraciji.

```
static void Main(string[] args)
{
    string unos = Console.ReadLine();
    Console.WriteLine(metoda(unos));
}
static int metoda(string x)
{
    for (int i = x.Length; i >= 0; i--)
    {
        if (i % 2 == 1)
        {
            Console.Write(x[i]);
        }
    }
    Console.WriteLine();
    return x.Length;
}
```



Rješenje:

Komentar	unos	x	i	TRENUTNI ISPIS	KOMPLETAN ISPIS
unos tražene vrijednosti s tipkovnice unos = "ALGPRO"	"ALGPRO"				
unutar Console.WriteLine() poziva se metoda unos kojoj se šalje vrijednost "ALGPRO"					
pokreće se metoda koja za x prima vrijednost "ALGPRO"		"ALGPRO"			
ulazak u for petlju, obzirom da je x.Length = 6 (broj slova u stringu s) postavlja se da je i=6			6		
uvjet ponavljanja petlje 6 >= 0 je true pa ulazimo u petlju					
uvjet provjerava je li i % 2 == 1 tj. uvjet će biti istina kada je i neparan; obzirom da je i paran uvjet se neće ostvariti					
na kraju petlje se i smanjuje za 1 (i--) i petlja se pokreće ponovno jer je uvjet ponavljanja petlje istinit (i >= 0)			5		
uvjet provjerava je li i % 2 == 1, uvjet se ostvaruje i ispisuje se x[i]				0	0
na kraju petlje se i smanjuje za 1 (i--) i petlja se pokreće ponovno jer je uvjet ponavljanja petlje istinit (i >= 0)			4		
uvjet provjerava je li i % 2 == 1, uvjet se neće ostvariti					
na kraju petlje se i smanjuje za 1 (i--) i petlja se pokreće ponovno jer je uvjet ponavljanja petlje istinit (i >= 0)			3		
uvjet provjerava je li i % 2 == 1 uvjet se ostvaruje i ispisuje se x[i]				p	OP



Komentar	unos	x	i	TREKUTNI ISPIS	KOMPLETAN ISPIS
na kraju petlje se i smanjuje za 1 (i--) i petlja se pokreće ponovno jer je uvjet ponavljanja petlje istinit (i >= 0)			2		
uvjet provjerava je li i % 2 == 1, uvjet se neće ostvariti					
na kraju petlje se i smanjuje za 1 (i--) i petlja se pokreće ponovno jer je uvjet ponavljanja petlje istinit (i >= 0)			1		
uvjet provjerava je li i % 2 == 1, uvjet se ostvaruje i ispisuje se x[i]				L	OPL
na kraju petlje se i smanjuje za 1 (i--) i petlja se pokreće ponovno jer je uvjet ponavljanja petlje istinit (i >= 0)			0		
uvjet provjerava je li i % 2 == 1, uvjet se neće ostvariti					
na kraju petlje se i smanjuje za 1 (i--) i petlja se ne pokreće ponovno jer je uvjet ponavljanja petlje nije istinit (i >= 0)					
Console.WriteLine(); samo kursor prebaci u novi red					
return x.Length vraća broj znakova u stringu x (x.Length = 6)					
broj 6 se vraća gdje me metoda bila pozvana tj. u Console.WriteLine() iz Main-a te se tako ispisuje broj 6				6	OPL 6



Zadatak 2

Što će se ispisati u konzolu nakon izvršavanja sljedećeg programskog odsječka, ukoliko su s konzole učitane vrijednosti 7 i 3. Potrebno je naznačiti redoslijed ispisivanja. Kao točno rješenje priznaje se samo ono u kojem je vidljiv postupak i sadržaj varijabli u svakom pozivu metoda.

```
static void Main(string[] args)
{
    int x = Convert.ToInt32(Console.ReadLine());
    int y = Convert.ToInt32(Console.ReadLine());
    if (x / y > 2)
    {
        metoda(x, y);
    }
    else
    {
        metoda(y, x);
    }
    Console.WriteLine(x + " , " + y);
}
static void metoda(double x, int y)
{
    double z = x / y;
    switch (z)
    {
        case 2.0:
            Console.Write("-", );
            break;
        case 2.5:
            Console.Write("X, ");
            break;
        default:
            Console.Write("=", );
            break;
    }
    x = y++;
    Console.Write((x + y) + ",");
}
```



Rješenje:

Komentar	x (Main)	y (Main)	x (metoda)	y (metoda)	z	TRENUTNI ISPIS	KOMPLETAN ISPIS
x i y se unose s tipkovnice x = 7 y = 3	7	3					
provjerava se uvjet $x / y > 2$, $7 / 3 = 2$ (zaokružuje se jer je $\text{int} / \text{int} = \text{int}$) pa uvjet $2 > 2$ nije istinit, zbog toga se izvršava else dio if grananja							
u else dijelu se poziva metoda kojoj se šalju 3 i 7							
pokreće se metoda koja prima 3 i 7, tj. x u metodi ima vrijednost 3, a y ima vrijednost 7			3	7			
$z = x / y = 3 / 7 = 0.42$, ne zaokružuje se na nula jer $\text{double} / \text{int} = \text{double}$					0,42		
switch provjerava varijablu z, z nije jednak niti jednom case-u pa se radi samo default						=,	=,
kod reda $x = y++$; x prima vrijednost od y, i onda se y povećava za 1			7	8			
Console.Write() ispisuje poruku						15,	=, 15,
metoda se završava i vraćamo se u Main gdje je ostao samo ispis						7 , 3	=, 15,7 , 3



Zadatak 3

Napišite C# kod za izračun udjela osobnih i teretnih vozila. S konzole je potrebno učitati broj N koji predstavlja ukupan broj vozila, te potom učitati N cjelobrojnih vrijednosti s konzole. Ukoliko je unesena vrijednost 1 radi se o osobnom automobilu, a ako je unesena vrijednost 2 radi se o teretnom vozilu (nije potrebna provjera da li je unesena ispravna vrijednost).

Ispis udjela osobnih i teretnih vozila potrebno je napraviti u zasebnoj metodi Ispis koja kao argumente prima ukupan broj osobnih i ukupan broj teretnih vozila.

Ispis mora biti u formatu kao u primjeru: „Udio osobnih vozila iznosi: 42.7123645%, a udio teretnih vozila iznosi 57.2876355%!“.

Rješenje:

```
static void Main(string[] args)
{
    Console.Write("Unesite željeni broj vozila N: ");
    int n = Convert.ToInt32(Console.ReadLine());

    int brojOsobnihAutomobila = 0;
    int brojTeretnihVozila = 0;

    for (int i = 1; i <= n; i++)
    {
        Console.Write("Unesite " + i + ". vozilo: ");
        int vozilo = Convert.ToInt32(Console.ReadLine());

        if (vozilo == 1)
        {
            brojOsobnihAutomobila++;
        }
        else if (vozilo == 2)
        {
            brojTeretnihVozila++;
        }
    }

    Ispis(brojOsobnihAutomobila, brojTeretnihVozila);

    Console.ReadKey();
}

static void Ispis(int brojOsobnihAutomobila, int brojTeretnihVozila)
{
    int ukupanBrojVozila = brojOsobnihAutomobila + brojTeretnihVozila;

    double udioOsobnihAutomobila = (Convert.ToDouble(brojOsobnihAutomobila)
    / ukupanBrojVozila) * 100;
    double udioTeretnihVozila = (Convert.ToDouble(brojTeretnihVozila)
    / ukupanBrojVozila) * 100;

    Console.WriteLine("Udio osobnih vozila iznosi: " + udioOsobnihAutomobila
    + "%, a udio teretnih vozila iznosi " + udioTeretnihVozila + "%!");
}
```



Zadatak 4

Napišite C# kod za izračun udaljenosti između 100 telekomunikacijskih baznih stanica, čije su koordinate upisane u datoteci bazneStanice.txt. U svakom retku datoteke upisane su x i y koordinate (realni brojevi) bazne stanice u formatu x;y (ukupno ima 100 redaka).

Podatke je potrebno spremati u zasebno polje.

Nakon učitavanja podataka potrebno je kreirati i ispisati 100 x 100 matricu udaljenosti između baznih stanica. Format ispisa je proizvoljan.

Za izračun udaljenosti potrebno je koristiti euklidsku udaljenost zadanu jednadžbom:

$$d(T_1(x_1, y_1), T_2(x_2, y_2)) = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

Rješenje:

```
static void Main(string[] args)
{
    StreamReader reader = new StreamReader("bazneStanice.txt");

    double[,] bazneStanice = new double[100, 2];
    double[,] matricaUdaljenosti = new double[100, 100];
    int trenutniRed = 0;

    while (!reader.EndOfStream)
    {
        string[] dijeloviLinije = reader.ReadLine().Split(';');
        bazneStanice[trenutniRed, 0] = Convert.ToDouble(dijeloviLinije[0]);
        bazneStanice[trenutniRed, 1] = Convert.ToDouble(dijeloviLinije[1]);

        trenutniRed++;
    }

    Console.WriteLine("Udaljenosti između stanica: ");
    Console.Write("\t");
    for (int i = 0; i < matricaUdaljenosti.GetLength(0); i++)
    {
        Console.Write("s" + (i + 1) + "\t");
    }
    Console.WriteLine();

    for (int i = 0; i < matricaUdaljenosti.GetLength(0); i++)
    {
        Console.Write("s" + (i + 1) + "\t");
        for (int j = 0; j < matricaUdaljenosti.GetLength(0); j++)
        {
            matricaUdaljenosti[i, j] = Math.Sqrt(Math.Pow(bazneStanice[i, 0]
                - bazneStanice[j, 0], 2) + Math.Pow(bazneStanice[i, 1]
                - bazneStanice[j, 1], 2));
            Console.Write(matricaUdaljenosti[i, j] + "\t");
        }
        Console.WriteLine();
    }
    Console.ReadKey();
}
```



Zadatak 5

Napišite C# kod za učitavanje podataka o vozilu na tehničkom pregledu. Podatke o vozilima potrebno je učitavati s konzole u instance priložene klase VOZILO.

```
class VOZILO
{
    public int ID;
    public string REGISTRACIJA;
    public int GODINA_PROIZVODNJE;
    public string MODEL;
    public bool PROLAZAK_NA_PREGLEDU;
}
```

Vozila je potrebno učitavati sve dok se za ID vozila ne unese vrijednost 0. Sva unesena vozila potrebno je spremiti u listu vozila.

Nakon unesenih vozila potrebno je u proizvoljnom formatu ispisati registarske tablice svih vozila koja nisu prošla tehnički pregled.



Rješenje:

```
static void Main(string[] args)
{
    List<VOZILO> vozila = new List<VOZILO>();

    int id = -1;

    do
    {
        Console.Write("Unesite ID vozila: ");
        id = Convert.ToInt32(Console.ReadLine());

        if (id != 0)
        {
            VOZILO v = new VOZILO();
            v.ID = id;

            Console.Write("Unesite registraciju vozila: ");
            v.REGISTRACIJA = Console.ReadLine();

            Console.Write("Unesite godinu proizvodnje vozila: ");
            v.GODINA_PROIZVODNJE = Convert.ToInt32(Console.ReadLine());

            Console.Write("Unesite model vozila: ");
            v.MODEL = Console.ReadLine();

            Console.Write("Je li vozilo prošlo na tehničkom pregledu?
            (true/false) ");
            v.PROLAZAK_NA_PREGLEDU = Convert.ToBoolean(Console.ReadLine());

            vozila.Add(v);
        }
    } while (id != 0);

    Console.WriteLine("Registarske tablice svih vozila koja nisu prošla
    tehnički pregled:");

    foreach (VOZILO v in vozila)
    {
        if (!v.PROLAZAK_NA_PREGLEDU)
        {
            Console.WriteLine(v.REGISTRACIJA);
        }
    }

    Console.ReadKey();
}
```



Ispit #60 (08.07.21. - grupa A)

Zadatak 1

Napišite C# kod metode *IzracunStraniceTrokuta* koja prima argumente stranice **a**, **b** i kuta α (sve brojevi s plivajućim zarezom).

Metoda *IzracunStraniceTrokuta* u konzolu treba ispisati duljinu stranice **c** zaokruženu na 3 decimale u proizvoljnom formatu.

U *Main* metodi potrebno je unositi stranice i ispisati duljinu stranice **c** sve dok korisnik unosi ispravne vrijednosti, odnosno stranice **a** i **b**, te kut α moraju biti veći od nula.

Formula za izračun stranice: $c = \sqrt{a^2 + b^2 - 2ab * \cos(\alpha)}$

Rješenje:

```
static void Main(string[] args)
{
    double a = 0;
    double b = 0;
    double alfa = 0;

    do
    {
        Console.Write("Unesite stranicu a: ");
        a = Convert.ToDouble(Console.ReadLine());

        Console.Write("Unesite stranicu b: ");
        b = Convert.ToDouble(Console.ReadLine());

        Console.Write("Unesite kut alfa: ");
        alfa = Convert.ToDouble(Console.ReadLine());

        if (a > 0 && b > 0 && alfa > 0)
        {
            IzracunStraniceTrokuta(a, b, alfa);
        }
    } while (a > 0 && b > 0 && alfa > 0);

    Console.ReadKey();
}

static void IzracunStraniceTrokuta(double a, double b, double alfa)
{
    Console.WriteLine(Math.Round(Math.Sqrt(Math.Pow(a, 2)
        + Math.Pow(b, 2) - 2 * a * b * Math.Cos(alfa)), 3));
}
```



Zadatak 2

Napišite C# kod koji ispisuje i računa cijenu najjeftinijeg prijevoznika tereta na osnovu mase tereta i duljine puta. Prijevoznici *CveleTrans*, *PRIJEvoz* i *DataTrans* imaju vlastiti način izračuna cijene prijevoza zadan sljedećom tablicom:

Prijevoznik	Cijena
CveleTrans	$\sqrt{Masa^{\frac{3}{2}} * Duljina}$
PRIJEvoz	$\sqrt{Masa * Duljina^2}$
DataTrans	$Masa * 2 + Duljina$

U *Main* dijelu programa potrebo je učitati masu tereta i duljinu puta te u konzolu ispisati poruku u formatu: *Najjeftinija cijena prijevoznika (ime prijevoznika) iznosi: x kn!* pri čemu umjesto *(ime prijevoznika)* piše ime, a varijabla *x* predstavlja izračunatu cijenu prijevoza.

Rješenje:

```
static void Main(string[] args)
{
    Console.Write("Unesite masu tereta: ");
    double masa = Convert.ToDouble(Console.ReadLine());

    Console.Write("Unesite duljinu puta: ");
    double duljina = Convert.ToDouble(Console.ReadLine());

    double cijena1 = Math.Sqrt(Math.Pow(masa, 3.0 / 2) * duljina);
    double cijena2 = Math.Sqrt(masa * Math.Pow(duljina, 2));
    double cijena3 = masa * 2 + duljina;

    double najmanjaCijena = 0;
    string najjeftinijiPrijevoznik = "";

    if (cijena1 <= cijena2 && cijena1 <= cijena3)
    {
        najmanjaCijena = cijena1;
        najjeftinijiPrijevoznik = "CveleTrans";
    }
    else if (cijena2 <= cijena1 && cijena2 <= cijena3)
    {
        najmanjaCijena = cijena2;
        najjeftinijiPrijevoznik = "PRIJEvoz";
    }
    else
    {
        najmanjaCijena = cijena3;
        najjeftinijiPrijevoznik = "DataTrans";
    }

    Console.WriteLine("Najjeftinija cijena prijevoznika "
        + najjeftinijiPrijevoznik + " iznosi: " + najmanjaCijena + " kn!");

    Console.ReadKey();
}
```



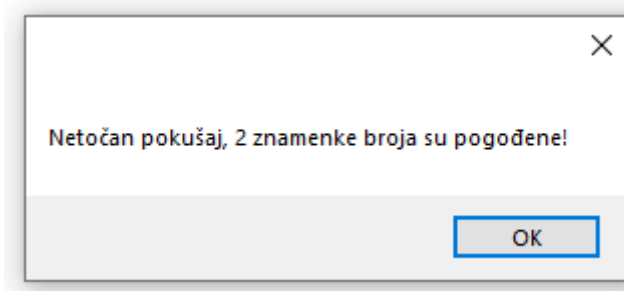
Zadatak 3

Napišite C# kod aplikacije koja traži od korisnika da pogodi nasumično generiran prirodan troznamenkasti broj. Korisnik treba pritisnuti gumb *Započni* (*btnGenerirajBroj*) kako bi se generirao slučajan troznamenkasti broj koji treba pogoditi.

Potom, korisnik treba unositi brojeve u *TextBox* naziva *txtPokusaj* te pritiskati gumb *Pogodi* (*btnPogodi*) sve dok ne pogodi nasumično generiran troznamenkasti broj. Povijest pogađanja potrebno je bilježiti u *RichTextBox*-u naziva *txtPovijestPogađanja*.

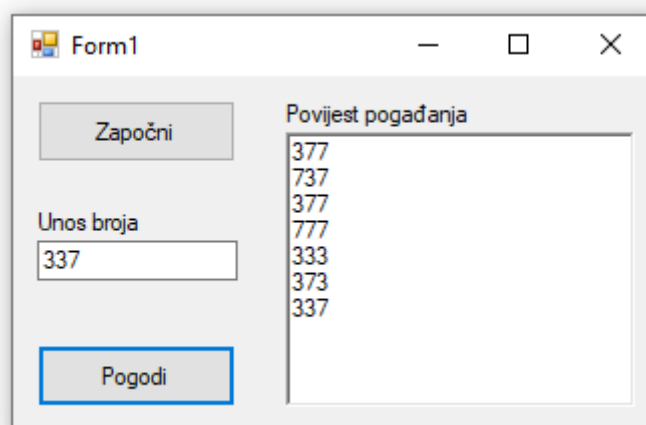
Nakon svakog pokušaja, u *MessageBox* treba se ispisati je li korisnik pogodio broj ili nije. Ukoliko nije pogodio broj, u *MessageBox*-u je potrebno ispisati koliko znamenki (računajući redoslijed) je korisnik pogodio. Za dohvat znamenki cijelog broja potrebno je koristiti već napisanu metodu *dohvatiZnamenke* (nije ju potrebno pisati), koja vraća polje cijelih brojeva, pri čemu je prvi element polja prva znamenka predanog broja, a zadnji element polja zadnja znamenka predanog broja.

Izgled poruke prikazan je na slici:



U exam.netu napisati samo dijelove koda unutar *Nadopuni*.

Izgled sučelja prikazan je na slici:



```

public partial class Form1 : Form
{
    //Nadopuni - 1

    public Form1()
    {
        InitializeComponent();
    }

    private void btnGenerirajBroj_Click(object sender, EventArgs e)
    {
        //Nadopuni - 2

    }

    private void btnPogodi_Click(object sender, EventArgs e)
    {
        //Nadopuni - 3

    }

    private static int[] dohvatiZnamenke(int broj)
    {
        ...
    }
}

```



Rješenje:

```
public partial class Form1 : Form
{
    //Nadopuni - 1
    Random rnd = new Random();
    int nasumicniBroj = 0;

    public Form1()
    {
        InitializeComponent();
    }

    private void btnGenerirajBroj_Click(object sender, EventArgs e)
    {
        //Nadopuni - 2
        nasumicniBroj = rnd.Next(100, 1000);
    }

    private void btnPogodi_Click(object sender, EventArgs e)
    {
        //Nadopuni - 3
        int pokusaj = Convert.ToInt32(txtPokusaj.Text);

        txtPovijestPogađanja.Text += pokusaj + "\n";

        int[] pokusajKaoPolje = dohvatiZnamenke(pokusaj);
        int[] nasumicniBrojKaoPolje = dohvatiZnamenke(nasumicniBroj);
        int brojPogodaka = 0;

        for (int i = 0; i < nasumicniBrojKaoPolje.Length; i++)
        {
            if (nasumicniBrojKaoPolje[i] == pokusajKaoPolje[i])
            {
                brojPogodaka++;
            }
        }

        if (brojPogodaka == 3)
        {
            MessageBox.Show("Pogodili ste nasumičan broj!");
        }
        else
        {
            MessageBox.Show("Netočan pokušaj, " + brojPogodaka
                + " znamenke broja su pogođene!");
        }
    }

    private static int[] dohvatiZnamenke(int broj)
    {
        ...
    }
}
```



Zadatak 4

Napišite C# kod konzolne aplikacije koja računa cijenu autobusnog prijevoza.

Potrebno je prvo napraviti klasu naziva *BusPrijevoznici* koja sadrži javne attribute *ImePrijevoznika*, *BrojSjedala* (cijeli broj), *CijenaKarte* (broj s plivajućim zarezom).

U *Main* dijelu programa potrebno je kreirati 2 autobusna prijevoznika po želji preko konstruktora klase koji treba primiti sve attribute (nije potrebno učitavanje iz konzole).

Potom je preko konzole potrebno unijeti broj putnika i udaljenost koju treba prevesti, te na temelju unesenog, izračunati i ispisati ukupnu dobit za svakog prijevoznika po formuli:

$$\text{Dobit} = \text{BrojPutnika} * \text{CijenaKarte} - \text{Udaljenost} * 4.5$$

Ispis u konzolu treba sadržavati ime prijevoznika i njegovu dobit.

Rješenje:

```
static void Main(string[] args)
{
    BusPrijevoznici prijevoznik1 = new BusPrijevoznici("APP Split", 60, 120);
    BusPrijevoznici prijevoznik2 = new BusPrijevoznici("APP Zagreb", 50, 150);

    Console.Write("Unesite broj putnika:");
    int brojPutnika = Convert.ToInt32(Console.ReadLine());

    Console.Write("Unesite udaljenost: ");
    double udaljenost = Convert.ToDouble(Console.ReadLine());

    double dobitPrijevoznika1 = brojPutnika * prijevoznik1.CijenaKarte
        - udaljenost * 4.5;
    double dobitPrijevoznika2 = brojPutnika * prijevoznik2.CijenaKarte
        - udaljenost * 4.5;

    Console.WriteLine("Prijevoznik " + prijevoznik1.ImePrijevoznika
        + " ima dobit " + dobitPrijevoznika1);
    Console.WriteLine("Prijevoznik " + prijevoznik2.ImePrijevoznika
        + " ima dobit " + dobitPrijevoznika2);

    Console.ReadKey();
}

class BusPrijevoznici
{
    public string ImePrijevoznika;
    public int BrojSjedala;
    public double CijenaKarte;

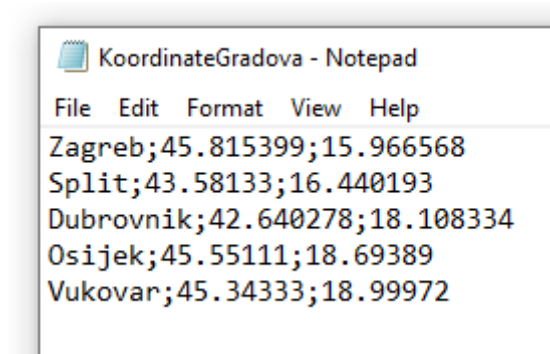
    public BusPrijevoznici(string IMEPRIJEVOZNIKA, int BROJSJEDALA,
        double CIJENAKARTE)
    {
        ImePrijevoznika = IMEPRIJEVOZNIKA;
        BrojSjedala = BROJSJEDALA;
        CijenaKarte = CIJENAKARTE;
    }
}
```



Zadatak 5

Napišite C# kod programa koji pritiskom na gumb *Učitaj* (*btnUcitaj*) iz tekstualne datoteke *KoordinateGradova.txt*, smještenoj na relativnoj putanji, učitava zapise *ImeGrada;širina;dužina*, pri čemu je *širina* geografska širina (y koordinata), a *dužina* geografska dužina (x koordinata).

Izgled datoteke *KoordinateGradova.txt* prikazan je na slici:



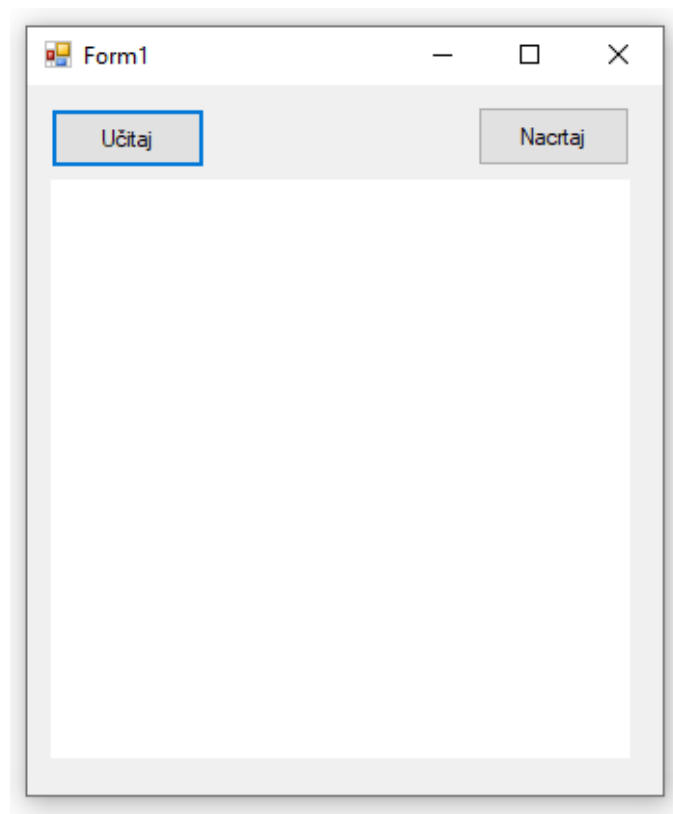
Pritiskom na gumb *Nacrtaj* (*btnNacrtaj*) potrebno je u *PictureBox* naziva *slika* iscrtati liniju između točke (0,0) i grada najudaljenijeg točki (0,0). Dio C# koda za iscrtavanje linije već je napisan.

U exam.netu napisati samo dijelove koda unutar *Nadopuni*.

Za izračun udaljenosti između dvije točke koristiti Euklidsku udaljenost.



Izgled sučelja prikazan je na slici:



```
public partial class Form1 : Form
{
    //Nadopuni - 1

    public Form1()
    {
        InitializeComponent();
    }

    private void btnUcitaj_Click(object sender, EventArgs e)
    {
        //Nadopuni - 2
    }

    private void btnNacrtaj_Click(object sender, EventArgs e)
    {
        //Nadopuni - 3

        slika.Refresh();
    }

    private void slika_Paint(object sender, PaintEventArgs e)
    {
        e.Graphics.DrawLine(Pens.Black, 0, 0, (float)xND, (float)yND);
    }
}
```



Rješenje:

```
public partial class Form1 : Form
{
    Nadopuni - 1
    double xND = 0;
    double yND = 0;

    public Form1()
    {
        InitializeComponent();
    }

    private void btnUcitaj_Click(object sender, EventArgs e)
    {
        //Nadopuni - 2
        StreamReader reader = new StreamReader("KoordinateGradova.txt");

        double najvecaUdaljenost = double.MinValue;

        while (!reader.EndOfStream)
        {
            string linija = reader.ReadLine();
            string[] dijeloviLinije = linija.Split(';');

            double x = Convert.ToDouble(dijeloviLinije[2]);
            double y = Convert.ToDouble(dijeloviLinije[1]);

            double trenutnaUdaljenost = Math.Sqrt(Math.Pow(x, 2)
                + Math.Pow(y, 2));

            if (trenutnaUdaljenost > najvecaUdaljenost)
            {
                najvecaUdaljenost = trenutnaUdaljenost;
                xND = x;
                yND = y;
            }
        }

        reader.Close();
    }

    private void btnNacrta_Click(object sender, EventArgs e)
    {
        //Nadopuni - 3

        slika.Refresh();
    }

    private void slika_Paint(object sender, PaintEventArgs e)
    {
        e.Graphics.DrawLine(Pens.Black, 0, 0, (float)xND, (float)yND);
    }
}
```



Ispit #61 (08.07.21. - grupa B)

Zadatak 1

Napišite C# kod metode *OpsegMnogokuta* koja kao argument prima listu duljina stranica mnogokuta. Metoda u konzolu treba ispisati opseg mnogokuta u formatu: „Mnogokut sa n stranica ima opseg O “, pri čemu vrijednost varijabli n i O treba zamijeniti odgovarajućim vrijednostima.

U *Main* metodi je potrebno preko konzole unijeti broj n koji predstavlja broj stranica mnogokuta, popuniti listu s duljinama stranica (broj s plivajućim zarezom) također učitanih s konzole i pozvati metodu *OpsegMnogokuta*.

Rješenje:

```
static void Main(string[] args)
{
    List<double> lista = new List<double>();

    Console.WriteLine("Unesite broj stranica mnogokuta n: ");
    int n = Convert.ToInt32(Console.ReadLine());

    for (int i = 0; i < n; i++)
    {
        Console.WriteLine("Unesite duljinu prve stranice: ");
        double a = Convert.ToDouble(Console.ReadLine());

        lista.Add(a);
    }

    OpsegMnogokuta(lista);

    Console.ReadKey();
}

static void OpsegMnogokuta(List<double> lista)
{
    Console.WriteLine("Mnogokut sa " + lista.Count
        + " stranica ima opseg " + lista.Sum());
}
```



Zadatak 2

Napišite C# kod konzolne aplikacije koja traži od korisnika da pogodi slučajno generiran prirodan troznamenkasti broj. Korisnik mora unositi brojeve sve dok ne pogodi nasumično generiran troznamenkasti broj.

Također, aplikacija nakon svakog pokušaja mora ispisati je li korisnik pogodio znamenke broja kojeg pogađa, te ako nije treba ispisati koliko znamenki (prema redoslijedu) je pogodio.

Pomoć: za dohvat pojedinih znamenki troznamenkastog broja može se koristiti cjelobrojno dijeljenje i operator za izračun ostatka dijeljenja.

Primjer:

Nasumično generiran broj: 231

Korisnik unio broj: 831

Primjer ispisa: „Netočan pokušaj, 2 znamenke broja su pogođene!“



Rješenje:

```
static void Main(string[] args)
{
    Random rnd = new Random();
    int generiraniBroj = rnd.Next(100, 1000);
    int generiraniBrojTemp= 0;
    int brojPogodjenihZnamenki = 0;

    do
    {
        brojPogodjenihZnamenki = 0;
        generiraniBrojTemp = generiraniBroj;

        Console.Write("Unesite broj: ");
        int x = Convert.ToInt32(Console.ReadLine());

        do
        {
            int ZadnjaZnamenkaX = x % 10;
            int ZadnjaZnamenkaGenerirano = generiraniBrojTemp % 10;

            if (ZadnjaZnamenkaX == ZadnjaZnamenkaGenerirano)
            {
                brojPogodjenihZnamenki++;
            }

            generiraniBrojTemp /= 10;
            x /= 10;
        } while (generiraniBrojTemp > 0);

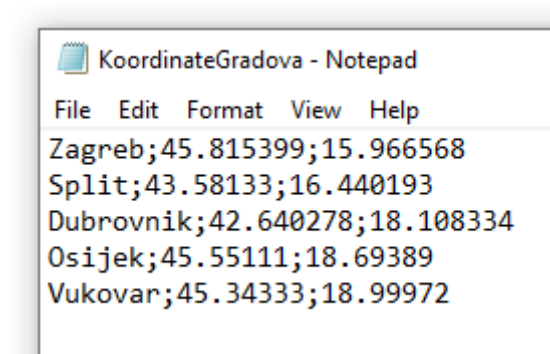
        if (brojPogodjenihZnamenki == 3)
        {
            Console.WriteLine("Pogodili ste broj!");
        }
        else
        {
            Console.WriteLine("Netočan pokušaj, "
                + brojPogodjenihZnamenki + " znamenke broja su pogođene!");
        }
    } while (brojPogodjenihZnamenki != 3);

    Console.ReadKey();
}
```



Zadatak 3

Napišite C# kod programa koji učitava tekstualnu datoteku *KoordinateGradova.txt* smještenu na relativnoj putanji, s 10 redova zapisa u formatu *ImeGrada;širina;dužina*, pri čemu je *širina* geografska širina (y koordinata), a *dužina* geografska dužina (x koordinata). Izgled datoteke *KoordinateGradova.txt* prikazan je na slici:



U *Main* dijelu programa potrebno je pročitati sadržaj datoteke i spremiti ju na proizvoljan način.

Potom, napišite i pozovite metodu *NajvecaUdaljenost* koja kao argument prima učitane podatke (prethodno odabran način spremanja podataka) te ispisuje ime i udaljenost grada koji je najudaljeniji od točke (0, 0).

Primjer ispisa: „*Najudaljeniji grad je grad D, s udaljenosti X km!*“, pri čemu vrijednost *G* i *X* treba zamijeniti s nazivom grada, odnosno udaljenošću. Za izračun udaljenosti potrebno je koristiti Euklidsku udaljenost pomnoženu s vrijednošću 111, jer je $1^\circ \approx 111\text{km}$.



Rješenje:

```
static void Main(string[] args)
{
    StreamReader reader = new StreamReader("KoordinateGradova.txt");

    string[,] podaci = new string[10, 3];

    for (int i = 0; i < podaci.GetLength(0); i++)
    {
        string linija = reader.ReadLine();
        string[] dijeloviLinije = linija.Split(';');

        podaci[i, 0] = dijeloviLinije[0];
        podaci[i, 1] = dijeloviLinije[1];
        podaci[i, 2] = dijeloviLinije[2];
    }

    reader.Close();

    NajvecaUdaljenost(podaci);

    Console.ReadKey();
}

static void NajvecaUdaljenost(string[,] podaci)
{
    double najvecaUdaljenost = double.MinValue;
    int indeks = -1;

    for (int i = 0; i < podaci.GetLength(0); i++)
    {
        double d = Math.Sqrt(Math.Pow(Convert.ToDouble(podaci[i, 2]),
            2) + Math.Pow(Convert.ToDouble(podaci[i, 1]), 2)) * 111;
        if (d > najvecaUdaljenost)
        {
            najvecaUdaljenost = d;
            indeks = i;
        }
    }

    Console.WriteLine("Najudaljeniji grad je grad " + podaci[indeks, 0]
        + ", s udaljenosti " + najvecaUdaljenost + " km!");
}
```



Zadatak 4

Napišite C# kod konzolne aplikacije koja računa cijenu taksi prijevoza.

Potrebno je prvo napisati klasu *TaxiPrijevoznici* koja sadrži privatni atribut *CijenaStart* i javne attribute *CijenaKmDan*, *CijenaKmNoc* (sve brojevi s plivajućim zarezom) i u *Main* dijelu kreirati 2 taxi prijevoznika po želji preko konstruktora koji prima sve attribute klase (nije potrebno učitavanje iz konzole).

Nakon toga, preko konzole korisnik treba unijeti doba dana i duljinu puta na temelju čega je potrebno ispisati dobit svakog od kreiranih taxi prijevoznika u proizvoljnom formatu.

Za dohvat odgovarajućih privatnih atributa potrebno je napisati odgovarajuće metode u klasi.

Dobit se računa sljedećim izrazom:

$$D = \text{CijenaStart} + \text{CijenaKm} * \text{DuljinaPut}$$



Rješenje:

```
static void Main(string[] args)
{
    TaxiPrijevoznici prijevoznik1 = new TaxiPrijevoznici(5, 10, 15);
    TaxiPrijevoznici prijevoznik2 = new TaxiPrijevoznici(6, 12, 20);

    Console.Write("Unesite doba dana: ");
    string dobaDana = Console.ReadLine();

    Console.Write("Unesite duljinu puta: ");
    double duljinaPuti = Convert.ToDouble(Console.ReadLine());

    double cijenaStart1 = prijevoznik1.DohvatiCijenuStarta();
    double cijenaKm1 = 0;
    double cijenaStart2 = prijevoznik2.DohvatiCijenuStarta();
    double cijenaKm2 = 0;

    if (dobaDana == "dan")
    {
        cijenaKm1 = prijevoznik1.CijenaKmDan;
        cijenaKm2 = prijevoznik2.CijenaKmDan;
    }
    else
    {
        cijenaKm1 = prijevoznik1.CijenaKmNoc;
        cijenaKm2 = prijevoznik2.CijenaKmNoc;
    }

    double D1 = cijenaStart1 + cijenaKm1 * duljinaPuti;
    double D2 = cijenaStart2 + cijenaKm2 * duljinaPuti;

    Console.WriteLine("Dobit prvog prijevoznika: " + D1 + " kn");
    Console.WriteLine("Dobit drugog prijevoznika: " + D2 + " kn");

    Console.ReadKey();
}

class TaxiPrijevoznici
{
    private double CijenaStart;
    public double CijenaKmDan;
    public double CijenaKmNoc;

    public TaxiPrijevoznici(double CIJENASTART, double CIJENAKMDAN,
        double CIJENAKMNOĆ)
    {
        CijenaStart = CIJENASTART;
        CijenaKmDan = CIJENAKMDAN;
        CijenaKmNoc = CIJENAKMNOĆ;
    }

    public double DohvatiCijenuStarta()
    {
        return CijenaStart;
    }
}
```



Zadatak 5

Nadopišite C# kod koji računa i ispisuje cijenu najjeftinijeg prijevoznika tereta na osnovu mase tereta i duljine puta. Prijevoznici *CveleTrans*, *PRIJEvoz* i *DataTrans* imaju vlastiti način izračuna cijene prijevoza zadan tablicom:

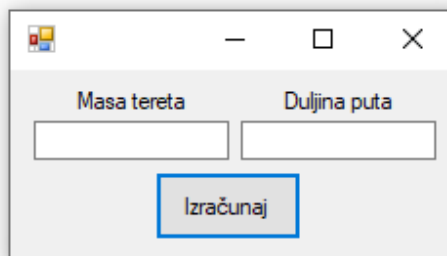
Prijevoznik	Cijena
CveleTrans	$\sqrt{Masa^{\frac{3}{2}} * Duljina}$
PRIJEvoz	$\sqrt{Masa * Duljina^2}$
DataTrans	$Masa * 2 + Duljina$

Pritiskom na gumb *Izračunaj* (*btnIzracunaj*) potrebno je izračunati cijenu prijevoza za svakog prijevoznika na temelju upisanih vrijednosti mase tereta u *TextBox* (*txtMasa*) i duljine puta u *TextBox* (*txtDuljina*) (brojevi s plivajućim zarezom).

Varijable cijene i imena najjeftinijeg prijevoznika potrebno je deklarirati na razini klase kako bi varijable bile vidljive svim metodama unutar klase.

Ukupnu cijenu i ime najjeftinijeg prijevoznika potrebno je ispisati u skočnom prozoru (*MessageBox*) u formatu *Najjeftinija cijena prijevoznika (ime prijevoznika) iznosi: x kn!* pri čemu umjesto (*ime prijevoznika*) treba pisati ime najjeftinijeg prijevoznika, a varijabla *x* predstavlja izračunatu cijenu prijevoza.

Sučelje je prikazano sljedećom slikom:



U exam.netu potrebno je napisati samo dijelove koda unutar *Nadopuni*.

```
public partial class Form1 : Form
{
    //Nadopuni - 1

    public Form1()
    {
        InitializeComponent();
    }

    private void btnIzracunaj_Click(object sender, EventArgs e)
    {
        //Nadopuni - 2
    }
}
```



Rješenje:

```
public partial class Form1 : Form
{
    //Nadopuni - 1
    double cijenaNajjeftinijeg = 0;
    string imeNajjeftinijeg = "";

    public Form1()
    {
        InitializeComponent();
    }

    private void btnIzracunaj_Click(object sender, EventArgs e)
    {
        //Nadopuni - 2
        double masaTereta = Convert.ToDouble(txtMasa.Text);
        double duljinaPuti = Convert.ToDouble(txtDuljina.Text);

        double cijena1 = Math.Sqrt(Math.Pow(masaTereta, 3.0 / 2) * duljinaPuti);
        double cijena2 = Math.Sqrt(masaTereta * Math.Pow(duljinaPuti, 2));
        double cijena3 = masaTereta * 2 + duljinaPuti;

        if (cijena1 <= cijena2 && cijena1 <= cijena3)
        {
            cijenaNajjeftinijeg = cijena1;
            imeNajjeftinijeg = "CveleTrans";
        }
        else if (cijena2 <= cijena1 && cijena2 <= cijena3)
        {
            cijenaNajjeftinijeg = cijena2;
            imeNajjeftinijeg = "PRIJEvoz";
        }
        else
        {
            cijenaNajjeftinijeg = cijena3;
            imeNajjeftinijeg = "DataTrans";
        }

        MessageBox.Show("Najjeftinija cijena prijevoznika " + imeNajjeftinijeg
            + " iznosi: " + cijenaNajjeftinijeg + " kn!");
    }
}
```



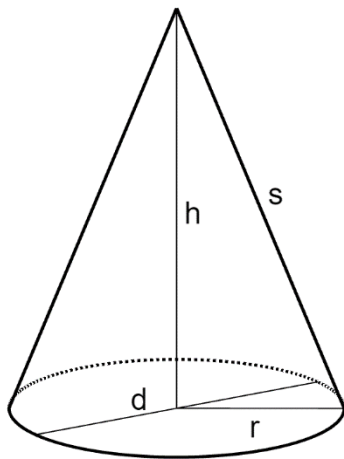
Napomena: rok nije kompletan (nedostaje zadatak 5)

Zadatak 1

Napišite C# kod koji temeljem unosa korisnika računa oplošje ili volumen stošca.

Potrebno je napraviti metodu *IzracunZaStozac* koja kao argument prima izbor unesen od korisnika (oplošje ili volumen) i dimenzije stošca (polumjer i visinu – realni brojevi), a kao rezultat vraća oplošje ili volumen stošca ovisno o izboru.

U *Main* metodi potrebno je unijeti izbor unesen od strane korisnika i dimenzije stošca (polumjer i visinu), pozvati metodu *IzracunZaStozac* i ispisati rezultat u konzolu u proizvoljnom formatu.



$$V = \frac{1}{3}\pi r^2 h$$

$$O = \pi r(r + s)$$

$$s = \sqrt{r^2 + h^2}$$

V – obujam

O – oplošje

r – polumjer

s – polumjer plašta

h – visina



Rješenje:

```
static void Main(string[] args)
{
    Console.Write("Unesite polumjer stošca: ");
    double r = Convert.ToDouble(Console.ReadLine());

    Console.Write("Unesite visinu stošca: ");
    double h = Convert.ToDouble(Console.ReadLine());

    Console.Write("Što želite računati? (oplošje / volumen) ");
    string izbor = Console.ReadLine();

    double rezultat = IzracunZaStozac(izbor, r, h);

    if (izbor == "volumen")
    {
        Console.WriteLine("V = " + rezultat);
    }
    else
    {
        Console.WriteLine("O = " + rezultat);
    }

    Console.ReadKey();
}

static double IzracunZaStozac(string izbor, double r, double h)
{
    if (izbor == "volumen")
    {
        return 1.0 / 3 * Math.PI * Math.Pow(r, 3) * h;
    }
    else
    {
        return Math.PI * r * (r + Math.Sqrt(Math.Pow(r, 2)
        + Math.Pow(h, 2)));
    }
}
```



Zadatak 2

Napišite C# kod konzolne aplikacije koja izračunava zbroj znamenki dva peteroznamenkasta broja. Učitavanje dva broja i ispis zbroja njihovih znamenki u proizvoljnom formatu potrebno je vršiti sve dok se unose ispravni peteroznamenkasti brojevi u rasponu [10000, 99999].

Rješenje:

```
static void Main(string[] args)
{
    do
    {
        Console.WriteLine("Unesite 1. broj: ");
        int x = Convert.ToInt32(Console.ReadLine());

        Console.WriteLine("Unesite 2. broj: ");
        int y = Convert.ToInt32(Console.ReadLine());

        int sumaZnamenki = 0;

        if (x >= 10000 && x <= 99999 && y >= 10000 && y <= 99999)
        {
            do
            {
                int zadnjaZnamenkaX = x % 10;
                int zadnjaZnamenkaY = y % 10;

                sumaZnamenki += zadnjaZnamenkaX + zadnjaZnamenkaY;

                x /= 10;
                y /= 10;
            } while (x > 0);

            Console.WriteLine("Suma znamenki unesenih brojeva je "
                + sumaZnamenki);
        }
        else
        {
            break;
        }
    } while (true);

    Console.ReadKey();
}
```



Zadatak 3

Napišite C# kod konzolne aplikacije koja računa cijenu avionskih karata.

Potrebno je prvo napraviti klasu *AvioKompanije* koja sadrži javne attribute *ImePrijevoznika*, *BrojSjedala* (cijeli broj), *CijenaKarte* (cijeli broj) te u *Main* dijelu kreirati 2 proizvoljna avio prijevoznika bez konstruktora (instance klase *AvioKompanije*, nije potrebno učitavanje vrijednosti iz konzole)).

Nakon toga, preko konzole korisnik treba unijeti broj putnika (cijeli broj) koji će se prevesti. Na temelju unesenog, izračunajte i ispišite neostvarenu dobit za svakog prijevoznika po formuli:

$$\text{NeostvarenaDobit} = \text{BrojPraznihSjedala} * \text{CijenaKarte}$$

Format ispisa je proizvoljan, ali mora sadržavati ime prijevoznika i neostvarenu dobit.

Rješenje:

```
static void Main(string[] args)
{
    AvioKompanije kompanija1 = new AvioKompanije();
    AvioKompanije kompanija2 = new AvioKompanije();

    kompanija1.ImePrijevoznika = "AK prijevoz";
    kompanija1.BrojSjedala = 80;
    kompanija1.CijenaKarte = 350;

    kompanija2.ImePrijevoznika = "Avio transport";
    kompanija2.BrojSjedala = 100;
    kompanija2.CijenaKarte = 400;

    Console.WriteLine("Unesite broj putnika: ");
    int brojPutnika = Convert.ToInt32(Console.ReadLine());

    int neostvarenaDobit1 = (kompanija1.BrojSjedala - brojPutnika)
    * kompanija1.CijenaKarte;
    int neostvarenaDobit2 = (kompanija2.BrojSjedala - brojPutnika)
    * kompanija2.CijenaKarte;

    Console.WriteLine("Neostvarena dobit prijevoznika "
    + kompanija1.ImePrijevoznika + ": " + neostvarenaDobit1);
    Console.WriteLine("Neostvarena dobit prijevoznika "
    + kompanija2.ImePrijevoznika + ": " + neostvarenaDobit2);

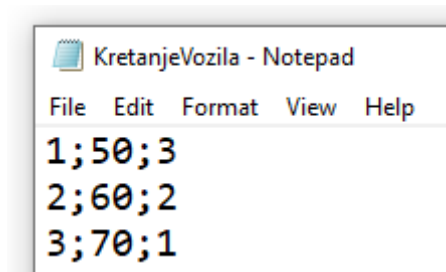
    Console.ReadKey();
}

class AvioKompanije
{
    public string ImePrijevoznika;
    public int BrojSjedala;
    public int CijenaKarte;
}
```



Zadatak 4

Napišite C# kod programa koji iz tekstualne datoteke *KretanjeVozila.txt* Smještene na D: disku, s 1000 redova zapisa, učitava zapise u formatu *VoziloID;brzina_km/h;VrijemePutovanja_h*. Izgled datoteke *KretanjeVozila.txt* prikazan je na slici:



U *Main* dijelu programa potrebno je pročitati i spremiti sadržaj datoteke u dvodimenzionalno polje.

Napišite metodu *DuljinaPutovanja* koja kao argument prima učitano polje i ID vozila za koji se želi izračunati prijeđena udaljenost na temelju brzine vozila i vremena putovanja. Metoda treba ispisati prijeđenu udaljenost vozila.

Format ispisa: „*Prijeđena udaljenost vozila (IDVozila) je X km*“. Varijabla *X* kod ispisa predstavlja izračunatu prijeđenu udaljenost vozila, dok (*IDVozila*) treba zamijeniti ID-em vozila za kojeg se računa udaljenost.



Rješenje:

```
static void Main(string[] args)
{
    int[,] podaci = new int[100, 3];

    StreamReader reader = new StreamReader("D:\\KretanjeVozila.txt");

    int trenutniRed = 0;

    while (!reader.EndOfStream)
    {
        string linija = reader.ReadLine();
        string[] dijeloviLinije = linija.Split(';');

        podaci[trenutniRed, 0] = Convert.ToInt32(dijeloviLinije[0]);
        podaci[trenutniRed, 1] = Convert.ToInt32(dijeloviLinije[1]);
        podaci[trenutniRed, 2] = Convert.ToInt32(dijeloviLinije[2]);

        trenutniRed++;
    }

    reader.Close();

    Console.ReadKey();
}

static void DuljinaPutovanja(int[,] podaci, int IDvozila)
{
    for (int i = 0; i < podaci.GetLength(0); i++)
    {
        if (podaci[i, 0] == IDvozila)
        {
            int udaljenost = podaci[i, 1] * podaci[i, 2];
            Console.WriteLine("Prijeđena udaljenost vozila " + IDvozila
                + " je " + udaljenost + " km");
            break;
        }
    }
}
```



Ispit #63 (02.09.21. - grupa A)

Zadatak 1

Kako će izgledati polje 45, -8, -44, 6, 99, 7 nakon svih iteracija selection sort algoritma (sortiranje od najmanjeg prema najvećem).

Potrebno je naznačiti svaku zamjenu elemenata polja zasebno u svim koracima svake iteracije.

Rješenje:

zadano polje	45	-8	-44	6	99	7
1. korak	-44	-8	45	6	99	7
2. korak	-44	-8	45	6	99	7
3. korak	-44	-8	6	45	99	7
4. korak	-44	-8	6	7	99	45
5. korak	-44	-8	6	7	45	99
6. korak	-44	-8	6	7	45	99

Objašnjenje: selection sort-om bira se najmanji element kojeg se postavlja na prvo mjesto, potom se sljedeći najmanji dodaje na drugo mjesto itd. dok cijelo polje ne bude sortirano.



Zadatak 2

Što će biti zapisano u datoteci *data.txt* nakon što se izvrši naredba *Flush* na kraju sljedećeg programskog odsječka.

Kao točno rješenje priznaje se samo ono u kojem je vidljiv postupak i sadržaj varijabli u svakoj iteraciji i sadržaj upisan u datoteku nakon izvršavanja.

```
static void Main(string[] args)
{
    StreamWriter izlaz = new StreamWriter("data.txt");
    for (int i = 2; i <= 13; i++)
    {
        if (Provjera(i))
        {
            izlaz.WriteLine("PAR");
        }
        else
        {
            izlaz.WriteLine(i);
        }
        if (i % 8 == 1) break;
    }
    izlaz.Flush();
    izlaz.Close();
}
public static bool Provjera(int i)
{
    return (i % 2 == 0);
}
```

Rješenje:

Kratko objašnjenje: for petlja je postavljena da se ponavlja za vrijednosti brojača *i* [2, 13], ali uvjet na kraju petlje (*i* % 8 == 1) prekida petlju prije toga, tj. petlja će se prekinuti na kraju prolaska za vrijednost brojača *i* = 9. Obzirom da se u metodi *Provjera* samo provjerava je li broj paran može se zaključiti da će se za parne vrijednosti brojača u datoteku zapisati „PAR“, a za neparne vrijednosti brojača će se ispisati trenutna vrijednost brojača *i*. Svi zapisi se prvo zapisuju u memoriju. Tek kod poziva metode *Flush* se sadržaj mi memorije zapisuje u datoteku. Zapis u datoteci će biti:

```
PAR
3
PAR
5
PAR
7
PAR
9
```



Objašnjenje po koracima:

Akcija	i	Zapis u memoriji	Zapis u datoteci	Dodatno objašnjenje
ulazak u petlju	2			uvjet petlje je istinit pa se ulazi u petlju
uvjet (Provjera(i))				metoda provjera vraća true za parne vrijednosti brojača <i>i</i> , a false za neparne vrijednosti; obzirom da je <i>i</i> paran broj metoda Provjera vraća true
zapis u datoteku		PAR		obzirom da je uvjet if-a bio true ispisuje se "PAR"
uvjet ($i \% 8 == 1$)				uvjet nije istina pa se naredba break ne izvršava
sljedeća iteracija petlje	3			petlja se završava za $i = 2$ i pokreće se ponovno za $i = 3$; uvjet petlje je istinit pa se petlja petlja ponavlja
uvjet (Provjera(i))				obzirom da je <i>i</i> neparan broj metoda Provjera vraća false
zapis u datoteku		PAR 3		obzirom da je uvjet if-a bio false ispisuje se vrijednost brojača <i>i</i>
uvjet ($i \% 8 == 1$)				uvjet nije istina pa se naredba break ne izvršava
sljedeća iteracija petlje	4			petlja se završava za $i = 3$ i pokreće se ponovno za $i = 4$; uvjet petlje je istinit pa se petlja petlja ponavlja
uvjet (Provjera(i))				obzirom da je <i>i</i> paran broj metoda Provjera vraća true
zapis u datoteku		PAR 3 PAR		obzirom da je uvjet if-a bio true ispisuje se "PAR"
uvjet ($i \% 8 == 1$)				uvjet nije istina pa se naredba break ne izvršava



Akcija	i	Zapis u memoriji	Zapis u datoteci	Dodatno objašnjenje
sljedeća iteracija petlje	5			petlja se završava za $i = 4$ i pokreće se ponovno za $i = 5$; uvjet petlje je istinit pa se petlja petlja ponavlja
uvjet (Provjera(i))				obzirom da je i neparan broj metoda Provjera vraća false
zapis u datoteku		PAR 3 PAR 5		obzirom da je uvjet if-a bio false ispisuje se vrijednost brojača i
uvjet ($i \% 8 == 1$)				uvjet nije istina pa se naredba break ne izvršava
sljedeća iteracija petlje	6			petlja se završava za $i = 5$ i pokreće se ponovno za $i = 6$; uvjet petlje je istinit pa se petlja petlja ponavlja
uvjet (Provjera(i))				obzirom da je i paran broj metoda Provjera vraća true
zapis u datoteku		PAR 3 PAR 5 PAR		obzirom da je uvjet if-a bio true ispisuje se "PAR"
uvjet ($i \% 8 == 1$)				uvjet nije istina pa se naredba break ne izvršava
sljedeća iteracija petlje	7			petlja se završava za $i = 6$ i pokreće se ponovno za $i = 7$; uvjet petlje je istinit pa se petlja petlja ponavlja
uvjet (Provjera(i))				obzirom da je i neparan broj metoda Provjera vraća false
zapis u datoteku		PAR 3 PAR 5 PAR 7		obzirom da je uvjet if-a bio false ispisuje se vrijednost brojača i
uvjet ($i \% 8 == 1$)				uvjet nije istina pa se naredba break ne izvršava



Akcija	i	Zapis u memoriji	Zapis u datoteci	Dodatno objašnjenje
sljedeća iteracija petlje	8			petlja se završava za $i = 7$ i pokreće se ponovno za $i = 8$; uvjet petlje je istinit pa se petlja petlja ponavlja
uvjet (Provjera(i))				obzirom da je i paran broj metoda Provjera vraća true
zapis u datoteku		PAR 3 PAR 5 PAR 7 PAR		obzirom da je uvjet if-a bio true ispisuje se "PAR"
uvjet ($i \% 8 == 1$)				uvjet nije istina pa se naredba break ne izvršava
sljedeća iteracija petlje	9			petlja se završava za $i = 6$ i pokreće se ponovno za $i = 7$; uvjet petlje je istinit pa se petlja petlja ponavlja
uvjet (Provjera(i))				obzirom da je i neparan broj metoda Provjera vraća false
zapis u datoteku		PAR 3 PAR 5 PAR 7 PAR 9		obzirom da je uvjet if-a bio false ispisuje se vrijednost brojača i
uvjet ($i \% 8 == 1$)				uvjet je istina pa se naredba break izvršava; break bezuvjetno napušta petlju



Akcija	i	Zapis u memoriji	Zapis u datoteci	Dodatno objašnjenje
izlaz.Flush();			PAR 3 PAR 5 PAR 7 PAR 9	podaci iz memorije se zapisuju na disk; sada u memoriji više nema zapisa
izlaz.Close();				zatvara se otvorena datoteka



Zadatak 3

Napišite C# kod metode *IzracunVolumenaStosca* koja prima argumente polumjera baze r i visine h (brojevi s plivajućim zarezom).

Metoda *IzracunVolumenaStosca* u konzolu treba ispisati volumen stošca zaokružen na 2 decimale u proizvoljnom formatu.

U Main metodi potrebno je osigurati unos duljine radijusa i visine stošca te napraviti provjeru upisanih vrijednosti (r i h moraju biti veći od 0).

Formula za izračun volumena $V = \frac{1}{3}r^2\pi h$

Rješenje:

```
static void Main(string[] args)
{
    double r = 0.0;
    double h = 0.0;

    do
    {
        Console.Write("Unesite polumjer stošca: ");
        r = Convert.ToDouble(Console.ReadLine());

        if (r <= 0)
        {
            Console.WriteLine("Pogrešan unos polumjera! Pokušajte ponovno!");
        }
    } while (r <= 0);

    do
    {
        Console.Write("Unesite visinu stošca: ");
        h = Convert.ToDouble(Console.ReadLine());

        if (h <= 0)
        {
            Console.WriteLine("Pogrešan unos visine! Pokušajte ponovno!");
        }
    } while (h <= 0);

    IzracunVolumenaStosca(r, h);

    Console.ReadKey();
}

static void IzracunVolumenaStosca(double r, double h)
{
    double V = (1.0 / 3) * Math.Pow(r, 2) * Math.PI * h;
    Console.WriteLine("V = " + Math.Round(V, 2));
}
```



Zadatak 4

Napišite C# kod konzolne aplikacije za upravljanje parkiralištem.

Potrebno je napisati klasu *Vozilo* koja sadrži javne attribute *Registracija (tekst)*, *VrijemeUlaska (DateTime)*, *VrijemeIzlaska (DateTime)*.

U aplikaciji je prvo potrebno odabrati radi li se o ulasku novog vozila ili o izlasku vozila s parkirališta. Ukoliko se radi o ulasku potrebno je zabilježiti vrijeme dolaska vozila, a ukoliko se radi o izlasku potrebno je ispisati vrijeme provedeno na parkiralištu.

Sva vozila unutar parkirališta trebaju imati zaseban objekt unutar liste *ListaVozilaParking* koja je već kreirana unutar aplikacije i nije ju potrebno inicijalizirati.



Rješenje:

```
static void Main(string[] args)
{
    List<Vozilo> ListaVozilaParking = new List<Vozilo>();

    Console.Write("Unesite izlazak/ulazak vozila: ");
    string unos = Console.ReadLine();

    if (unos == "ulazak")
    {
        Vozilo v = new Vozilo();

        Console.Write("Unesite registraciju vozila: ");
        v.Registracija = Console.ReadLine();
        v.VrijemeUlaska = DateTime.Now;
        ListaVozilaParking.Add(v);
    }
    else
    {
        Console.Write("Unesite registraciju vozila: ");
        string registracija = Console.ReadLine();

        for (int i = 0; i < ListaVozilaParking.Count; i++)
        {
            if (registracija == ListaVozilaParking[i].Registracija)
            {
                ListaVozilaParking[i].VrijemeIzlaska = DateTime.Now;
                TimeSpan vrijemeParkinga =
                    ListaVozilaParking[i].VrijemeIzlaska
                    - ListaVozilaParking[i].VrijemeUlaska;

                int sati =
                    Convert.ToInt32(Math.Floor(vrijemeParkinga.TotalHours));
                int minute =
                    Convert.ToInt32(Math.Floor(vrijemeParkinga.TotalMinutes))
                    - sati * 60;
                int sekunde =
                    Convert.ToInt32(Math.Floor(vrijemeParkinga.TotalSeconds))
                    - sati * 3600 - minute * 60;

                Console.WriteLine("Vrijeme provedeno na parkingu je "
                    + sati + ":" + minute + ":" + sekunde);
                break;
            }
        }
    }

    Console.ReadKey();
}

class Vozilo
{
    public string Registracija;
    public DateTime VrijemeUlaska;
    public DateTime VrijemeIzlaska;
}
```



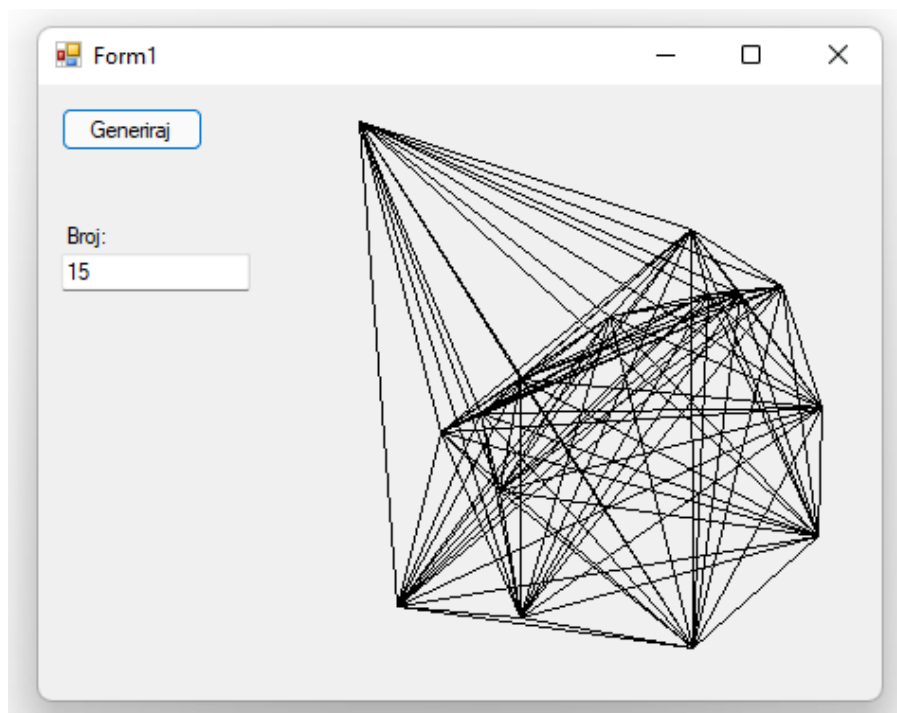
Zadatak 5

Napišite C# kod programa koji pritiskom na gumb *Generiraj* stvara dvodimenzionalno polje s dva stupca i brojem redaka jednakih vrijednosti upisanoj u TextBoxu *txtBoxBroj*.

U svaki element polja potrebno je upisati nasumični cijeli broj u rasponu od 0 do 300. Prvi stupac predstavlja x-koordinatu, dok drugi predstavlja y koordinatu točke.

Nakon generiranja brojeva, potrebno je u PictureBox (postavljen na dimenzije 300 x 300) iscrtati sve moguće linije koje povezuju generirane točke.

U exam.netu napisati samo dijelove koda unutar Nadopuni.



```
public partial class Form1 : Form
{
    //Nadopuni 1

    public Form1()
    {
        InitializeComponent();
    }

    private void btnGeneriraj_Click(object sender, EventArgs e)
    {
        //Nadopuni 2

        pictureBox1.Refresh();
    }

    private void pictureBox1_Paint(object sender, PaintEventArgs e)
    {
        //Nadopuni 3
    }
}
```



Rješenje:

```
public partial class Form1 : Form
{
    //Nadopuni 1
    int[,] polje = new int[0, 0];
    Random rnd = new Random();

    public Form1()
    {
        InitializeComponent();
    }

    private void btnGeneriraj_Click(object sender, EventArgs e)
    {
        //Nadopuni 2

        int brojRedova = Convert.ToInt32(txtBoxBroj.Text);

        polje = new int[brojRedova, 2];

        for (int i = 0; i < polje.GetLength(0); i++)
        {
            for (int j = 0; j < polje.GetLength(1); j++)
            {
                polje[i, j] = rnd.Next(0, 300);
            }
        }

        pictureBox1.Refresh();
    }

    private void pictureBox1_Paint(object sender, PaintEventArgs e)
    {
        //Nadopuni 3

        for (int i = 0; i < polje.GetLength(0); i++)
        {
            int x1 = polje[i, 0];
            int y1 = polje[i, 1];

            for (int j = 0; j < polje.GetLength(0); j++)
            {
                int x2 = polje[j, 0];
                int y2 = polje[j, 1];
                e.Graphics.DrawLine(Pens.Black, x1, y1, x2, y2);
            }
        }
    }
}
```



Ispit #64 (02.09.21. - grupa B)

Zadatak 1

Kako će izgledati polje 45, -8, -44, 6, 99, 7 nakon svih iteracija insertion sort algoritma (**sortiranje od najvećeg prema najmanjem**). Potrebno je naznačiti svaku zamjenu elemenata polja zasebno u svim koracima svake iteracije.

Rješenje:

zadano polje	45	-8	-44	6	99	7
1. korak	45	-8	-44	6	99	7
2. korak	45	-8	-44	6	99	7
3. korak	45	-8	-44	6	99	7
4. korak	45	6	-8	-44	99	7
5. korak	99	45	6	-8	-44	7
6. korak	99	45	7	6	-8	-44

Objašnjenje: skup elemenata za sortiranje dijeli se na sortirani i nesortirani. Svakim korakom povećava se sortirani dio tako da se iz nesortiranog dijela uzima prvi element i ubacuje ga se na odgovarajuće mjesto u sortiranom dijelu.



Zadatak 2

Što će biti zapisano u datoteci *data.txt* nakon što se izvrši naredba *Flush* na kraju sljedećeg programskog odsječka.

Kao točno rješenje priznaje se samo ono u kojem je vidljiv postupak i sadržaj varijabli u svakoj iteraciji i sadržaj upisan u datoteku nakon izvršavanja naredbe.

```
static void Main(string[] args)
{
    StreamWriter izlaz = new StreamWriter("data.txt");
    bool[] A = new bool[14];
    for (int i = 2; i <= 13; i++) A[i] = true;
    for (int i = 2; i <= 13; i++)
    {
        if (A[i])
        {
            izlaz.WriteLine(i);
            for (int j = i * 2; j <= 13; j = j + i)
            {
                A[j] = false;
            }
        }
    }
    izlaz.Flush();
    izlaz.Close();
}
```



Rješenje:

Red „`bool[] A = new bool[14];`“ će kreirati polje s 14 bool vrijednosti i sve spremljene vrijednosti će biti false.

Petlja „`for (int i = 2; i <= 13; i++) A[i] = true;`“ će proći kroz polje i promijeniti vrijednosti polja od indeksa 2 do indeksa 13 u true, tj. polje će nakon ovog reda u sebi imati sljedeće vrijednosti:

indeks	0	1	2	3	4	5	6	7	8	9	10	11	12	13
vrijednost	false	false	true	true	true	true	true	true	true	true	true	true	true	true

Potom ulazimo u petlju za koju ćemo pratiti vrijednost brojača i. Ulazimo u petlju za vrijednost $i = 2$ i provjerava se uvjet `if (A[i])`. Obzirom da je uvjet istinit u memoriju se zapisuje trenutna vrijednost od i, a to je 2. Onda se ulazi u petlju koja ima brojač $j = i * 2 = 2 * 2 = 4$. Brojač ove petlje se povećava za 2 svakim prolaskom kroz petlju, tj. petlja se ponavlja za vrijednost brojača j: 4, 6, 8, 10, 12. Unutarnja for petlja mijenja vrijednosti polja u false na svim indeksima koji su zapravo vrijednosti brojača j. Zbog toga će na kraju prolaska unutarnje petlje polje u sebi imati sljedeće vrijednosti:

indeks	0	1	2	3	4	5	6	7	8	9	10	11	12	13
vrijednost	false	false	true	true	false	true	false	true	false	true	false	true	false	true

Vanjska petlja se ponavlja za $i = 3$ i provjerava se uvjet `if (A[i])`. Obzirom da je uvjet istinit u memoriju se zapisuje trenutna vrijednost od i, a to je 3. Onda se ulazi u petlju koja ima brojač $j = i * 2 = 3 * 2 = 6$. Brojač ove petlje se povećava za 3 svakim prolaskom kroz petlju, tj. petlja se ponavlja za vrijednost brojača j: 6, 9, 12. Unutarnja for petlja mijenja vrijednosti polja u false na svim indeksima koji su zapravo vrijednosti brojača j. Zbog toga će na kraju prolaska unutarnje petlje polje u sebi imati sljedeće vrijednosti:

indeks	0	1	2	3	4	5	6	7	8	9	10	11	12	13
vrijednost	false	false	true	true	false	true	false	true	false	false	false	true	false	true

Vanjska petlja se ponavlja za $i = 4$ i provjerava se uvjet `if (A[i])`. Ispis i unutarnja petlja se preskaču jer uvjet nije istinit.

Vanjska petlja se ponavlja za $i = 5$ i provjerava se uvjet `if (A[i])`. Obzirom da je uvjet istinit u memoriju se zapisuje trenutna vrijednost od i, a to je 5. Onda se ulazi u petlju koja ima brojač $j = i * 2 = 5 * 2 = 10$. Brojač ove petlje se povećava za 5 svakim prolaskom kroz petlju, tj. petlja se ponavlja samo za vrijednost brojača j= 10. Unutarnja for petlja mijenja vrijednosti polja u false na svim indeksima koji su zapravo vrijednosti brojača j. Polje time ostaje isto kao i prije jer je na desatom indeksu već spremljena vrijednost false.

Vanjska petlja se ponavlja za $i = 6$ i provjerava se uvjet `if (A[i])`. Ispis i unutarnja petlja se preskaču jer uvjet nije istinit.

Vanjska petlja se ponavlja za $i = 7$ i provjerava se uvjet `if (A[i])`. Obzirom da je uvjet istinit u memoriju se zapisuje trenutna vrijednost od i, a to je 7. Onda se ulazi u petlju koja ima brojač $j = i * 2 = 7 * 2 = 14$. Obzirom da uvjet ponavljanja unutarnje petlje nije zadovoljen ($14 <= 13$), unutarnja petlja se nikada ne pokreće.

Vanjska petlja se ponavlja za $i = 8$ i provjerava se uvjet `if (A[i])`. Ispis i unutarnja petlja se preskaču jer uvjet nije istinit. Isto je i za vrijednosti 9 i 10 brojača i.



Vanjska petlja se ponavlja za $i = 11$ i provjerava se uvjet `if (A[i])`. Obzirom da je uvjet istinit u memoriju se zapisuje trenutna vrijednost od i , a to je **11**. Onda se ulazi u petlju koja ima brojač $j = i * 2 = 11 * 2 = 22$. Obzirom da uvjet ponavljanja unutarnje petlje nije zadovoljen ($22 \leq 13$), unutarnja petlja se nikada ne pokreće.

Vanjska petlja se ponavlja za $i = 12$ i provjerava se uvjet `if (A[i])`. Ispis i unutarnja petlja se preskaču jer uvjet nije istinit.

Vanjska petlja se ponavlja za $i = 13$ i provjerava se uvjet `if (A[i])`. Obzirom da je uvjet istinit u memoriju se zapisuje trenutna vrijednost od i , a to je **13**. Onda se ulazi u petlju koja ima brojač $j = i * 2 = 13 * 2 = 26$. Obzirom da uvjet ponavljanja unutarnje petlje nije zadovoljen ($26 \leq 13$), unutarnja petlja se nikada ne pokreće.

Završava se vanjska petlja i izvršava se naredba „`izlaz.Flush()`“ koja će sve iz memorije zapisati na disk, tj. ispis u datoteci će biti:

2
3
5
7
11
13

Naredba „`izlaz.Close()`“ zatvara datoteku.



Zadatak 3

Napišite C# kod metode *IzracunVolumenaPiramide* koja prima argumente površine baze B i visine h (sve brojevi s plivajućim zarezom).

Metoda *IzracunVolumenaPiramide* u konzolu treba ispisati volumen piramide zaokružen na 3 decimale u proizvoljnom formatu.

U Main metodi potrebno je osigurati unos površine baze i visine piramide, te napraviti provjeru upisanih vrijednosti (B i h moraju biti veći od 0).

Formula za izračun volumena $V = \frac{1}{3}B * h$

Rješenje:

```
static void Main(string[] args)
{
    double B = 0.0;
    double h = 0.0;

    do
    {
        Console.Write("Unesite površinu baze piramide B: ");
        B = Convert.ToDouble(Console.ReadLine());

        if (B <= 0)
        {
            Console.WriteLine("Pogrešan unos površine baze! Pokušajte ponovno!");
        }
    } while (B <= 0);

    do
    {
        Console.Write("Unesite visinu piramide h: ");
        h = Convert.ToDouble(Console.ReadLine());

        if (h <= 0)
        {
            Console.WriteLine("Pogrešan unos visine! Pokušajte ponovno!");
        }
    } while (h <= 0);

    IzracunVolumenaPiramide(B, h);

    Console.ReadKey();
}

static void IzracunVolumenaPiramide(double B, double h)
{
    double V = (1.0 / 3) * B * h;
    Console.WriteLine("V = " + Math.Round(V, 3));
}
```



Zadatak 4

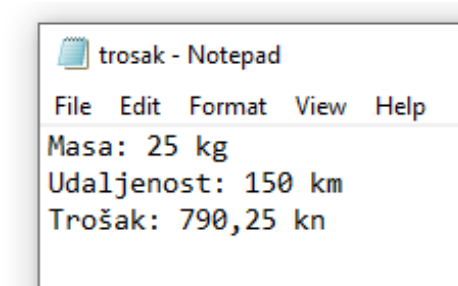
Napišite C# kod aplikacije za izračun troška prijevoza tereta prema unesenim parametrima *Udaljenost* i *Masa* (brojevi s plivajućim zarezom).

Klikom na gumb *Izracunaj* (*btnIzracunaj*) računa se ukupni trošak koji se ispisuje u obliku MessageBox-a. Trošak je potrebno izračunati prema izrazu:

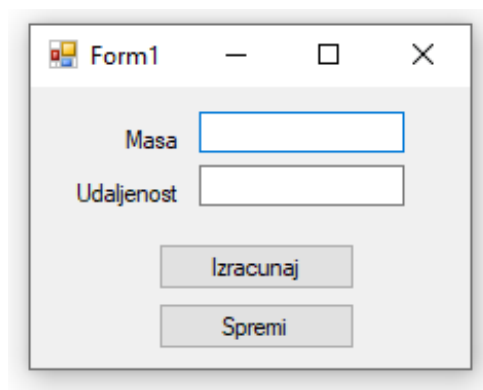
$$T = 100 + Udaljenost * 4,23 + Masa * 2,23, \quad \text{ako je } Masa < 100kg$$

$$T = 100 + Udaljenost * 4,23 + Masa * 1,98, \quad \text{ako je } Masa \geq 100kg$$

Nakon izračuna klikom na gumb *Spremi* (*btnSpremi*) se posljednji izračun sprema u tekstualnu datoteku prema primjeru na slici:



U exam.netu napisati samo dijelove koda unutar Nadopuni.



```
public partial class Form1 : Form
{
    //Nadopuni 1
    public Form1()
    {
        InitializeComponent();
    }
    private void btnIzracunaj_Click(object sender, EventArgs e)
    {
        //Nadopuni 2
    }
    private void btnSpremi_Click(object sender, EventArgs e)
    {
        //Nadopuni 3
    }
}
```



Rješenje:

```
public partial class Form1 : Form
{
    //Nadopuni 1
    double Masa = 0.0;
    double Udaljenost = 0.0;
    double T = 0.0;

    public Form1()
    {
        InitializeComponent();
    }

    private void btnIzracunaj_Click(object sender, EventArgs e)
    {
        //Nadopuni 2
        Masa = Convert.ToDouble(txtMasa.Text);
        Udaljenost = Convert.ToDouble(txtUdaljenost.Text);

        if (Masa < 100)
        {
            T = 100 + Udaljenost * 4.23 + Masa * 2.23;
        }
        else
        {
            T = 100 + Udaljenost * 4.23 + Masa * 1.98;
        }

        MessageBox.Show("Trošak: " + T + " kn");
    }

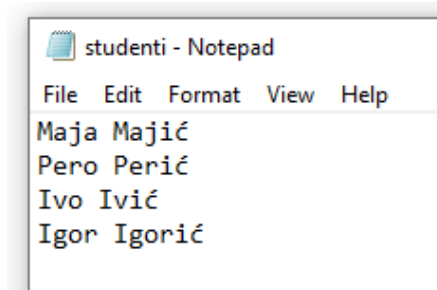
    private void btnSpremi_Click(object sender, EventArgs e)
    {
        //Nadopuni 3
        StreamWriter writer = new StreamWriter("trosak.txt");
        writer.WriteLine("Masa: " + Masa + " kg");
        writer.WriteLine("Udaljenost: " + Udaljenost + " kg");
        writer.WriteLine("Trošak: " + T + " kn");

        writer.Flush();
        writer.Close();
    }
}
```

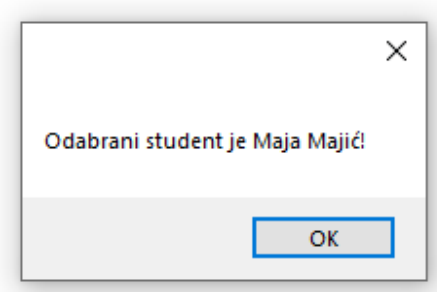


Zadatak 5

Napišite C# kod aplikacije koja može učitati tekstualnu datoteku s popisom studenata koja je prikazana na slici:



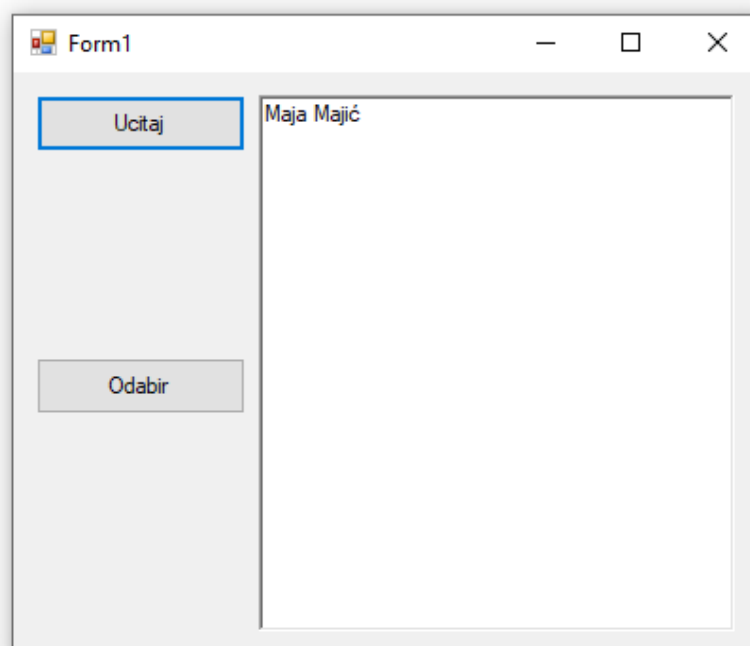
Korisnik treba pritisnuti gumb *Odabir* (*btnOdabir*) kako bi se s liste studenata odabrao nasumičan student, te se u MessageBox-u trebaju ispisati njegovi podaci kao u primjeru:



Ponovnim klikom na gumb *Odabir* bira se novi nasumični student, ali jednom odabrani student više ne može biti odabran do ponovnog učitavanja datoteke.

Podaci o svakom odabranom studentu se bilježe u RichTextBox-u *rchTxtBoxOdabrani*.

U exam.netu napisati samo dijelove koda unutar Nadopuni.



```
public partial class Form1 : Form
{
    //Nadopuni 1
    public Form1()
    {
        InitializeComponent();
    }

    private void btnUcitaj_Click(object sender, EventArgs e)
    {
        //Nadopuni 2
    }

    private void btnOdabir_Click(object sender, EventArgs e)
    {
        //Nadopuni 3
    }
}
```



Rješenje:

```
public partial class Form1 : Form
{
    //Nadopuni 1
    List<string> studenti = new List<string>();

    public Form1()
    {
        InitializeComponent();
    }

    private void btnUcitaj_Click(object sender, EventArgs e)
    {
        //Nadopuni 2
        studenti.Clear();

        try
        {
            StreamReader reader = new StreamReader("studenti.txt");

            while (!reader.EndOfStream)
            {
                studenti.Add(reader.ReadLine());
            }

            reader.Close();
        }
        catch (Exception ex)
        {
            MessageBox.Show("Greška: " + ex.Message);
        }
    }

    private void btnOdabir_Click(object sender, EventArgs e)
    {
        //Nadopuni 3
        if (studenti.Count > 0)
        {
            Random rnd = new Random();

            int indeksOdabranogStudenta = rnd.Next(0, studenti.Count);

            rchTextBoxOdabrani.Text += studenti[indeksOdabranogStudenta] + "\n";

            MessageBox.Show("Odabrani student je "
                + studenti[indeksOdabranogStudenta] + "!");

            studenti.RemoveAt(indeksOdabranogStudenta);
        }
        else
        {
            MessageBox.Show("Potrebno je učitati studente prije odabira!");
        }
    }
}
```



Zadatak 1

Kako će izgledati polje 45, 6, 99, 7 nakon svih iteracija bubble sort algoritma (sortiranje od najvećeg prema najmanjem). Potrebno je naznačiti svaku zamjenu elemenata polja zasebno u svim koracima svake iteracije.

Rješenje:

	1. iteracija sortiranja			
zadano polje	45	6	99	7
1. korak	45	6	99	7
2. korak	45	99	6	7
3. korak	45	99	7	6
4. korak	45	99	7	6

	2. iteracija sortiranja			
1. korak	99	45	7	6
2. korak	99	45	7	6
3. korak	99	45	7	6
4. korak	99	45	7	6

	3. iteracija sortiranja			
1. korak	99	45	7	6
2. korak	99	45	7	6
3. korak	99	45	7	6
4. korak	99	45	7	6

Objašnjenje: bubble sort je sortiranje zamjenom susjednih elemenata. Mijenjaju se vrijednosti dva susjedna elementa (ako je prvi element manji od drugog elementa). Sortiranje se završava kada prolaskom kroz cijeli niz ne bude zamjena.



Zadatak 2

Što će biti zapisano u datoteci *data.txt* nakon što se izvrši naredba *Flush* na kraju sljedećeg programskog odsječka.

Kao točno rješenje priznaje se samo ono u kojem je vidljiv postupak i sadržaj varijabli u svakoj iteraciji i sadržaj upisan u datoteku nakon izvršavanja naredbe.

```
static void Main(string[] args)
{
    StreamWriter izlaz = new StreamWriter("data.txt");
    int[] niz = { 3, 4, 2, 1, 5, 6 };
    int temp = 0;
    for (int i = 0; i < niz.GetLength(0); i++)
    {
        if (niz[i] > temp)
        {
            izlaz.WriteLine(niz[i]);
            temp = niz[i];
        }
        else
        {
            izlaz.Write(0);
            temp = niz[i];
        }
    }
    izlaz.Flush();
    izlaz.Close();
}
```



Rješenje:

Naredba „`int[] niz = { 3, 4, 2, 1, 5, 6 };`“ kreira jednodimenzionalno polje sa šest elemenata.

For petlja prolazi za vrijednosti brojača `i` `[0, 6>`. Gornja granica je šest jer „`niz.GetLength(0);`“ daje broj elemenata prve dimenzije polja, a obzirom da polje ima samo jednu dimenziju dat će broj elemenata, a to je šest. Može se izvući zaključak da je `GetLenght(0)` isto što i `Lenght` kad se radi o jednodimenzionalnom polju.

Bez obzira na uvjet, varijabla `temp` će uvijek poprimati vrijednost `niz[i]` jer ista naredba („`temp = niz[i];`“) nalazi `i` u `if-u` i u `else-u`.

Ako je uvjet točan („`niz[i] > temp`“) onda se ispisuje vrijednost `u` polju na indeksu `i` te se pokazivač prebacuje u novi red (jer se ispis obavlja naredbom „`WriteLine`“). Ako uvjet nije točan ispisuje se „0“ i pokazivač ostaje u istom redu (jer se ispis obavlja naredbom „`Write`“).

Ispis na ekranu na kraju programa:

3	- za <code>i = 0</code>
4	- za <code>i = 1</code>
005	- nule se ispisuju za <code>i = 2, i = 3</code> ; petica se ispisuje za <code>i = 4</code>
6	- za <code>i = 5</code>



Zadatak 3

Napišite C# kod koji ispisuje i računa cijenu najjeftinijeg pozivnog vozila (TAXI) na osnovu broja putnika i duljine puta. Prijevoznici TAXI-M, ZooM i FPZTAXI imaju vlastiti način izračuna cijene prijevoza zadan sljedećom tablicom:

Prijevoznik	Cijena
TAXI-M	$10 * BrojPutnika + Duljina * 5,43$
ZooM	$4 * BrojPutnika + Duljina * 7,32$
FPZTAXI	$20 + BrojPutnika * Duljina * 3,45$

U *Main* dijelu programa potrebno je učitati broj putnika (cijeli broj) i duljinu puta (broj s plivajućim zarezom) te u konzolu ispisati poruku u formatu: *Najjeftinija cijena (ime prijevoznika) iznosi: x kn!* pri čemu umjesto *(ime prijevoznika)* treba napisati ime najjeftinijeg prijevoznika, a varijabla *x* predstavlja izračunatu cijenu prijevoza.

Rješenje:

```
static void Main(string[] args)
{
    Console.Write("Unesite broj putnika: ");
    int BrojPutnika = Convert.ToInt32(Console.ReadLine());

    Console.Write("Unesite duljinu puta: ");
    double Duljina = Convert.ToDouble(Console.ReadLine());

    double cijena1 = 10 * BrojPutnika + Duljina * 5.43;
    double cijena2 = 4 * BrojPutnika + Duljina * 7.32;
    double cijena3 = 20 + BrojPutnika * Duljina * 3.45;

    double najmanjaCijena = 0.0;
    string najjeftinijiPrijevoznik = "";

    if (cijena1 <= cijena2 && cijena1 <= cijena3)
    {
        najmanjaCijena = cijena1;
        najjeftinijiPrijevoznik = "TAXI-M";
    }
    else if (cijena2 <= cijena1 && cijena2 <= cijena3)
    {
        najmanjaCijena = cijena2;
        najjeftinijiPrijevoznik = "ZooM";
    }
    else
    {
        najmanjaCijena = cijena3;
        najjeftinijiPrijevoznik = "FPZTAXI";
    }

    Console.WriteLine("Najjeftinija cijena " + najjeftinijiPrijevoznik
        + " iznosi: " + najmanjaCijena + " kn!");

    Console.ReadKey();
}
```



Zadatak 4

Napišite C# kod aplikacije za evidentiranje ocjena studenata na ispitu.

Potrebno je napisati klasu Student s javnim atributima: ime, prezime, JMBAG, ocjena.

U glavnoj metodi programa potrebno je učitati n studenata, pri čemu se n unosi s konzole. Nakon unosa svih podataka potrebno je ispisati prosječnu ocjenu svih studenata na ispitu, te ispisati ime i prezime studenata s najvišim brojem bodova.

Napomena: ovaj rok je prepun grešaka i zadataka koji nisu dovoljno precizno definirani (posebice ova grupa). Greška u ovom zadatku je što se u ovom zadatku traži unos ocjene, a za ispis se traži ispis studenata s najvišim brojem bodova. Pretpostavka je da je bila greška i da je i za ispis trebalo napraviti samo ispis studenata s najvećom ocjenom.

Ovakve greške sam u prošlim rokovima ispravio bez da sam napomenuo kakve greške su bile na ispitima, ali sam ovdje htio skrenuti pozornost na učestalost ovakvih pogrešaka.

Još jedna od stvari koje nisu dovoljno definirane je korištenje klase, jer se ovaj zadatak može riješiti bez upotrebe iste, ali je pretpostavka da su željeli da se svaki student kreira kao instanca klase Student.

Inače su ispiti i na kolegijima „Računalstvo“ i „Algoritmi i programiranje“ tražili samo ispis jednog studenta s najvećom ocjenom, ali kako je ovdje navedeno traži se ispis studenata (u množini) pa sam odlučio riješiti zadatak na taj način (koji je ujedno i teži).



Rješenje:

```
static void Main(string[] args)
{
    Console.Write("Unesite željeni broj studenata: ");
    int n = Convert.ToInt32(Console.ReadLine());

    List<Student> studenti = new List<Student>();
    int najvecaOcjena = 0;
    int zbrojOcjena = 0;

    for (int i = 1; i <= n; i++)
    {
        Student s = new Student();

        Console.Write("Unesite ime " + i + ". studenta: ");
        s.ime = Console.ReadLine();

        Console.Write("Unesite prezime " + i + ". studenta: ");
        s.prezime = Console.ReadLine();

        Console.Write("Unesite JMBAG " + i + ". studenta: ");
        s.JMBAG = Console.ReadLine();

        Console.Write("Unesite ocjenu " + i + ". studenta: ");
        s.ocjena = Convert.ToInt32(Console.ReadLine());

        studenti.Add(s);
        zbrojOcjena += s.ocjena;

        if (s.ocjena > najvecaOcjena)
        {
            najvecaOcjena = s.ocjena;
        }
    }

    double prosjecnaOcjena = zbrojOcjena / Convert.ToDouble(studenti.Count);

    Console.WriteLine("Prosječna ocjena: " + prosjecnaOcjena);

    Console.WriteLine("Studenti s najvećom ocjenom (" + najvecaOcjena + ") su: ");
    for (int i = 0; i < studenti.Count; i++)
    {
        if (studenti[i].ocjena == najvecaOcjena)
        {
            Console.WriteLine(studenti[i].ime + " " + studenti[i].prezime
                + " [ " + studenti[i].JMBAG + " ]");
        }
    }

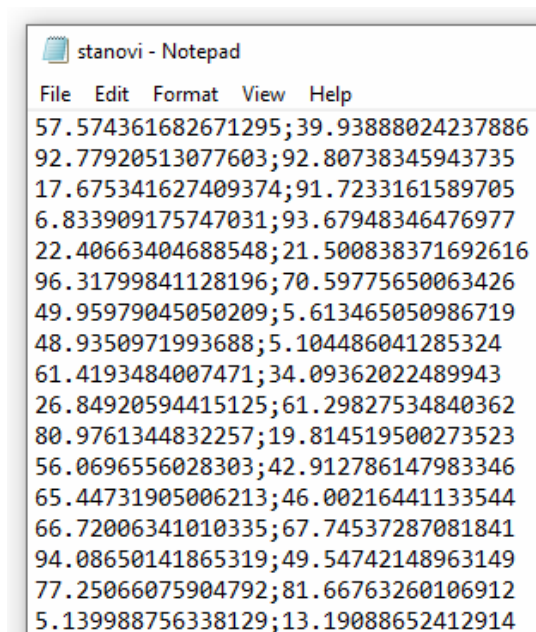
    Console.ReadKey();
}

class Student
{
    public string ime;
    public string prezime;
    public string JMBAG;
    public int ocjena;
}
```



Zadatak 5

Napišite C# kod koji učitava koordinate stambenih objekata iz tekstualne datoteke stanovi.txt.



```
stanovi - Notepad
File Edit Format View Help
57.574361682671295;39.93888024237886
92.77920513077603;92.80738345943735
17.675341627409374;91.7233161589705
6.833909175747031;93.67948346476977
22.40663404688548;21.500838371692616
96.31799841128196;70.59775650063426
49.95979045050209;5.613465050986719
48.9350971993688;5.104486041285324
61.4193484007471;34.09362022489943
26.84920594415125;61.29827534840362
80.9761344832257;19.814519500273523
56.0696556028303;42.912786147983346
65.44731905006213;46.00216441133544
66.72006341010335;67.74537287081841
94.08650141865319;49.54742148963149
77.25066075904792;81.66763260106912
5.139988756338129;13.19088652412914
```

Nakon učitavanja koordinata potrebno je odrediti optimalnu poziciju za izgradnju telekomunikacijske bazne stanice (središte učitanih koordinata) te ispisati izračunate koordinate u konzolu.



Rješenje:

```
using System.IO;

static void Main(string[] args)
{
    try
    {
        StreamReader reader = new StreamReader("stanovi.txt");

        int brojTocaka = 0;
        double sumaX = 0.0;
        double sumaY = 0.0;

        while (!reader.EndOfStream)
        {
            string linija = reader.ReadLine();
            string[] dijeloviLinije = linija.Split(';');

            brojTocaka++;
            sumaX += Convert.ToDouble(dijeloviLinije[0]);
            sumaY += Convert.ToDouble(dijeloviLinije[1]);
        }

        reader.Close();

        double xSredista = sumaX / brojTocaka;
        double ySredista = sumaY / brojTocaka;

        Console.WriteLine("Optimalna pozicija za izgradnju  
telekomunikacijske bazne stanice je T(" + xSredista + ", "  
+ ySredista + ").");
    }
    catch (Exception ex)
    {
        Console.WriteLine("Greška: " + ex.Message);
    }

    Console.ReadKey();
}
```



Ispit #76 (08.09.22. - grupa A)

Napomena: rok nije kompletan (nedostaju zadaci 1, 4 i 5)

Zadatak 2

Napišite C# kod konzolne aplikacije koja učitava vrijednosti iz datoteke *brojevi.txt* u listu cijelih brojeva.

Potom se izračunava i ispisuje na konzolu prosječna vrijednost svih pozitivnih elemenata liste. Ispis treba biti u formatu *Prosječna vrijednost x pozitivnih elemenata liste je prosjek*, pri čemu je x broj pronađenih pozitivnih vrijednosti u listi, a *prosjek* pronađena izračunata prosječna vrijednost.

Datoteka *brojevi.txt* smještena je na relativnoj putanji. Format zapisa u datoteci je *prvaVrijednost;drugaVrijednost*, svaka u zasebnom retku. Primjer sadržaja datoteke prikazan je na sljedećoj slici.



Potrebno je koristiti *try-catch* blok nad dijelom koda za rad s datotekom, te nije potrebno pisati *Using System.IO*.



Rješenje:

```
using System.IO;

static void Main(string[] args)
{
    List<int> lista = new List<int>();

    try
    {
        StreamReader reader = new StreamReader("brojevi.txt");

        while (!reader.EndOfStream)
        {
            string linija = reader.ReadLine();
            string[] dijeloviLinije = linija.Split(';');

            lista.Add(Convert.ToInt32(dijeloviLinije[0]));
            lista.Add(Convert.ToInt32(dijeloviLinije[1]));
        }

        reader.Close();
    }
    catch (Exception ex)
    {
        Console.WriteLine("Greška: " + ex.Message);
    }

    int x = 0;
    int zbrojPozitivnih = 0;

    for (int i = 0; i < lista.Count; i++)
    {
        if (lista[i] > 0)
        {
            x++;
            zbrojPozitivnih += lista[i];
        }
    }

    if (x > 0)
    {
        double prosjekPozitivnih = zbrojPozitivnih / Convert.ToDouble(x);

        Console.WriteLine("Prosječna vrijednost " + x
            + " pozitivnih elemenata liste je " + prosjekPozitivnih);
    }
    else
    {
        Console.WriteLine("U listi ne postoji niti jedan pozitivan broj!");
    }

    Console.ReadKey();
}
```



Zadatak 3

Što će se ispisati u konzolu nakon izvršavanja sljedećeg programskog odsječka?

```
int a = 7 % 4;
int b = 5 + a;
int c = 0;
int d = 1;

do
{
    if (a >= 2 && (b != 0 ^ d == 1))
    {
        c -= a;
        d = 2 * 10;
        b -= b;
    }
    else
    {
        c += a;
        d = a * 3 + 2;
        a--;
    }
    Console.WriteLine(a + "," + b + "," + c + "," + d);
} while (a + b != 10);
```



Rješenje:

Deklaracija i inicijalizacija varijabli prije petlje:

a	b	c	d
3	8	0	1

Ulazak u do-while petlju. Uvjet petlje se gleda tek nakon prvog prolaska.

U petlji se izvršava uvjetno grananje (if), tj. provjerava se njegov uvjet. Uvjet if-a se svodi na:

$(3 >= 2) \ \&\& \ (8 \neq 0 \wedge 1 == 1) \Rightarrow (true \ \&\& \ (true \wedge true)) \Rightarrow (true \ \&\& \ false) \Rightarrow false$

Obzirom da je uvjet if-a pogrešan, izvršava se else, tj. varijable će imati sljedeće vrijednosti:

a	b	c	d
2	8	3	11

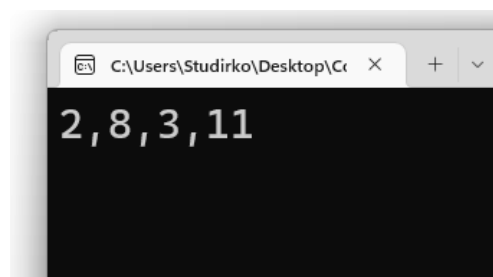
Naredba Console.WriteLine na ekran ispisuje poruku: „2,8,3,11“

Provjerava se uvjet ponavljanja petlje koji se svodi na:

$(2 + 8 \neq 10) \Rightarrow (10 \neq 10) \Rightarrow (false)$

Obzirom da uvjet petlje nije istinit petlja završava.

Izgled ekrana na kraju izvršavanja programa:



Ispit #77 (08.09.22. - grupa B)

Zadatak 1

Napišite C# kod metode *IspisiRetke* koja kao argument prima dvodimenzionalnu matricu cijelih brojeva *Z* i jednodimenzionalno polje cijelih brojeva *redci*. Metoda ispisuje elemente matrice *Z* na indeksu retka čija je vrijednost jednaka elementima polja *redci*.

Rješenje:

```
static void IspisiRetke(int[,] Z, int[] redci)
{
    for (int i = 0; i < redci.Length; i++)
    {
        Console.WriteLine("Red matrice na indeksu " + redci[i] + ":");

        for (int j = 0; j < Z.GetLength(1); j++)
        {
            Console.Write(Z[redci[i], j] + "\t");
        }
        Console.WriteLine();
    }
}
```

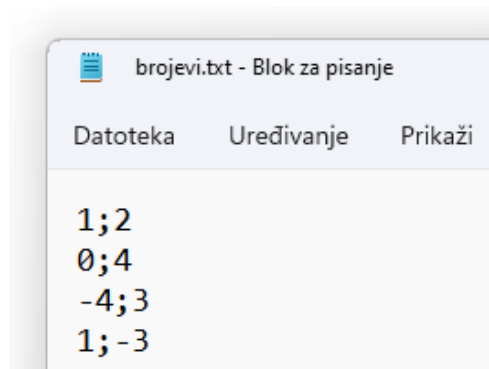


Zadatak 2

Napišite C# kod konzolne aplikacije koja učitava vrijednosti iz datoteke *brojevi.txt* u listu cijelih brojeva.

Potom se izračunava i ispisuje na konzolu najmanja vrijednost svih parnih elemenata liste. Ispis treba biti u formatu *Najmanja vrijednost x parnih elemenata liste je najmanjaVrijednost*, pri čemu je x broj pronađenih parnih vrijednosti u listi, a *najmanjaVrijednost* pronađena najmanja vrijednost.

Datoteka *brojevi.txt* smještena je na relativnoj putanji. Format zapisa u datoteci je *prvaVrijednost;drugaVrijednost*, svaka u zasebnom retku. Primjer sadržaja datoteke prikazan je na sljedećoj slici.



Potrebno je koristiti *try-catch* blok nad dijelom koda za rad s datotekom, te nije potrebno pisati *Using System.IO*.



Rješenje:

```
using System.IO;

static void Main(string[] args)
{
    List<int> lista = new List<int>();

    try
    {
        StreamReader reader = new StreamReader("brojevi.txt");

        while (!reader.EndOfStream)
        {
            string linija = reader.ReadLine();
            string[] dijeloviLinije = linija.Split(';');

            lista.Add(Convert.ToInt32(dijeloviLinije[0]));
            lista.Add(Convert.ToInt32(dijeloviLinije[1]));
        }

        reader.Close();
    }
    catch (Exception ex)
    {
        Console.WriteLine("Greška: " + ex.Message);
    }

    int x = 0;
    int najmanjaVrijednost = int.MaxValue;

    for (int i = 0; i < lista.Count; i++)
    {
        if (lista[i] % 2 == 0 && lista[i] != 0)
        {
            x++;

            if (lista[i] < najmanjaVrijednost)
            {
                najmanjaVrijednost = lista[i];
            }
        }
    }

    if (x > 0)
    {
        Console.WriteLine("Najmanja vrijednost " + x
            + " parnih elemenata liste je " + najmanjaVrijednost);
    }
    else
    {
        Console.WriteLine("U listi ne postoji niti jedna parna vrijednost!");
    }

    Console.ReadKey();
}
```



Zadatak 3

Što će se ispisati u konzolu nakon izvršavanja sljedećeg programskog odsječka?

```
int a = 10 % 2;
int b = 3 + a;
int c = 1;
int d = 0;

do
{
    if (a >= 2 && (b != 0 ^ d == 1))
    {
        c -= a;
        d = 2 * 10;
        b -= b;
    }
    else
    {
        c += a;
        d = a * 3 + 2;
        a--;
    }
    Console.WriteLine(a + "," + b + "," + c + "," + d);
} while (c + d == 1);
```



Rješenje:

Na ekranu će se ispisati: -1,3,1,2

Objašnjenje:

Red „`int a = 10 % 2;`“ će varijablu a postaviti na 0 jer % (modulo) je operator koji daje ostatak cjelobrojnog dijeljenja. Obzirom da dijeljenje broja 10 sa 2 daje 5 i ostatak 0 – varijabla a se postavlja na nulu.

Red „`int b = 3 + a;`“ postavlja vrijednost varijable b na 3 jer je vrijednost od a nula, pa je $3 + 0 = 3$.

Sljedeća dva reda postavljaju vrijednosti varijabli c na 1 i d na nula pa su vrijednosti poznatih varijabli kao u tablici:

a	b	c	d
0	3	1	0

Potom ulazimo u do-while petlju. Petlja se izvršava barem jednom jer se uvjet njezinog ponavljanja gleda tek na kraju svakog prolaska petlje.

Uvjet „`if (a >= 2 && (b != 0 ^ d == 1))`“ nije ispravan, pa se if dio neće izvršavati, već se izvršava else dio uvjetnog grananja.

Objašnjenje zašto uvjet „`a >= 2 && b != 0 ^ d == 1`“ nije istinit:

- prvo se rješavaju zagrade, tj. prvo se mora provjeriti dio uvjeta „`b != 0 ^ d == 1`“
- `b != 0` daje istinu, `d == 1` je laž, zbog toga je taj dio uvjeta zapravo `true ^ false`
- `^` je logički operator isključivo ili (XOR), tj. `true ^ false` daje `true`
- time smo cijeli dio uvjeta „`b != 0 ^ d == 1`“ sveli na `true`
- potom se promatra „`a >= 2`“ što daje laž, pa se cijeli uvjet svodi na `false && true`
- `&&` je logički operator i (AND), tj. `false && true` daje `false` i time cijeli uvjet nije istinit

U else dijelu se izvršavaju tri reda. Red „`c += a;`“ je isti kao da je napisano „`c = c + a;`“, tj. vrijednost varijable c (1) se povećava za vrijednost varijable a(0). Obzirom da je $1 + 0 = 1$ vrijednost varijable c i dalje ostaje 1.

Red „`d = a * 3 + 2;`“ postavlja vrijednost varijable d na 2 jer je $0 * 3 + 2 = 2$.

Red „`a--;`“ smanjuje vrijednost varijable a za 1 (dekrement), pa je vrijednost varijable a sada -1.

Da sumiramo, nakon izvršavanja else dijela uvjetnog grananja vrijednosti varijabli su:

a	b	c	d
-1	3	1	2

„`Console.WriteLine(a + "," + b + "," + c + "," + d);`“ ispisuje vrijednosti varijabli i između njih stavlja zarez. Time se na ekran ispisuje „-1,3,1,2“.

Obzirom da smo došli do kraja petlje provjerava se uvjet ponavljanja „`c + d == 1`“. Taj uvjet nije istinit jer $1 + 2$ daju 3, tj. $3 == 1$ nije istina. Petlja se prekida jer uvjet njezinog ponavljanja nije istinit.



Zadatak 4

U zadatku je potrebno naznačiti redoslijed i vrijednosti koje se ispisuju. Kao točno rješenje priznaje se samo ono u kojem je vidljiv postupak i opis zašto je ispisana vrijednost poprimila baš tu vrijednost.

```
static void Main(string[] args)
{
    int n = 2;
    string vozac = "";
    double x = 0;
    double y = 0;
    for (int i = 0; i < n; i++)
    {
        vozac = "Z";
        x = n - i;
        y = n + i;
    }
    Taxi taksi = new Taxi(vozac, x, y);
    taksi.povecajX();
    Console.WriteLine("Pozicija x taksi je " + taksi.POZICIJA_X + "!");
    Console.WriteLine("Pozicija y taksi je " + taksi.POZICIJA_Y + "!");
}

class Taxi
{
    public string VOZAC;
    public double POZICIJA_X;
    public double POZICIJA_Y;
    public Taxi(string vozac, double pozicija_x, double pozicija_y)
    {
        VOZAC = vozac;
        POZICIJA_X = pozicija_x;
        POZICIJA_Y = pozicija_y;
        postaviX(pozicija_x + 3);
        postaviY(pozicija_y + 2);
    }
    public void postaviX(double x)
    {
        Console.WriteLine(POZICIJA_X);
        POZICIJA_X = x;
    }
    public void postaviY(double y)
    {
        POZICIJA_Y = y;
        Console.WriteLine(POZICIJA_Y);
    }
    public void povecajX()
    {
        POZICIJA_X = POZICIJA_X + 2;
    }
    public void povecajY()
    {
        POZICIJA_Y = POZICIJA_Y + 1;
    }
}
```



Rješenje:

Na ekranu će se ispisati:

1

5

Pozicija x taksija je 6!

Pozicija y taksija je 5!

Objašnjenje:

Prva četiri reda samo deklariraju i inicijaliziraju vrijednosti za četiri varijable, kako je prikazano u tablici:

n	vozac	x	y
2	""	0	0

for petlja se ponavlja dva puta, i to kad je brojač i jednak 0 i kad je jednak 1. Brojač i će se povećati na 2, ali za tu vrijednost nikada neće ući u petlju jer će uvjet ponavljanja biti lažan ($i < n$ će biti $2 < 2$ što daje laž).

Prvi prolazak for petlje (za $i = 0$) će varijablu vozac postaviti na „Z“, dok će se vrijednost od x postaviti na 2 ($n - i$ će biti $2 - 0$ što daje 0), a vrijednost od y se također postavlja na 2 ($n + i$ će biti $2 + 0$ što daje 2). Zbog toga će varijable imati sljedeće vrijednosti nakon prvog prolaska for petlje:

n	vozac	x	y
2	"Z"	2	2

U drugom prolasku for petlje (za $i = 1$) vrijednost varijable vozač i dalje ostaje „Z“. Varijabla x se postavlja na 1 ($n - i$ će biti $2 - 1$ što daje 1), a vrijednost y se postavlja na 3 ($n + i$ će biti $2 + 1$ što daje 3). Zbog toga će varijable imati sljedeće vrijednosti nakon drugog prolaska for petlje:

n	vozac	x	y
2	"Z"	1	3

Red „`Taxi taksi = new Taxi(vozac, x, y);`“ kreira instancu klase Taxi koja se zove taksi. Vrijednosti varijabli vozac, x i y se šalju konstruktoru klase. Konstruktor klase prima vrijednost varijable vozac u varijablu vozac, vrijednost varijable x prima u varijablu pozicija_x, a vrijednost varijable y prima u varijablu pozicija_y. Potom se, prema primljenim varijablama, postavljaju vrijednosti atributa (instance taksi) VOZAC, POZICIJA_X i POZICIJA_Y. Njihove vrijednosti će biti:

VOZAC	POZICIJA_X	POZICIJA_Y
"Z"	1	3



Još smo u konstruktoru klase. Red „`postaviX(pozicija_x + 3);`“ poziva metodu `postaviX` kojoj šalje vrijednost varijable `pozicija_x` uvećana za 3. Varijablu `pozicija_x` konstruktor je primio iz `Main`-a s vrijednosti 1, tj. metodi `postaviX` se šalje vrijednost 4.

Metoda `postaviX` u varijablu `x` prima vrijednost 4. Prvo se ispisuje vrijednost atributa `POZICIJA_X` instance klase `Taxi` (koju smo nazvali taksi), tj. ispisuje 1 i prebacuje kursor u novi red (sljedeći ispis će biti u novom redu). Nakon ispisa postavlja se vrijednost atributa `POZICIJA_X` na vrijednost `x`-a što je četiri. Kada je metoda gotova, vraćamo se na mjesto gdje je metoda bila pozvana, tj. u konstruktor. U konstruktoru ostaje još jedan red, a to je poziv metode `postaviY` kojoj se šalje vrijednost varijable `pozicija_y` koju je konstruktor primio iz `Main`-a (a ta vrijednost je 3) uvećana za 2. Tako se broj 5 šalje metodi `postaviY` koja će vrijednost 5 spremi u varijablu `y`.

Metoda `postaviY` sprema vrijednost varijable `y` (5) u atribut `POZICIJA_Y`, te ga nakon toga ispisuje, tj. ispisuje se broj 5. Kada je metoda gotova, vraćamo se na mjesto poziva metode, tj. konstruktor. Poziv metode `postaviY` je ujedno bila i zadnja naredba u konstruktoru klase `Taxi`, pa se vraćamo u `Main` gdje je konstruktor bio pozvan. Trenutne vrijednosti glavnih atributa instance klase `Taxi` su:

VOZAC	POZICIJA_X	POZICIJA_Y
"Z"	4	5

Sada se izvršava red „`taksi.povecajX();`“. To je poziv metode `povecajX` nad instancom klase `Taxi` koja se zove taksi. U metodi se atribut `POZICIJA_X` instance taksi povećava za 2. Tako da su sada vrijednosti svih atributa:

VOZAC	POZICIJA_X	POZICIJA_Y
"Z"	6	5

Nakon što se metoda `povecajX` završi vraćamo se u `Main` gdje su ostale samo dvije naredbe za ispis. Svaka naredba će ispisati tekst i prebaciti pokazivač u novi red, a ispisuju se atributi `POZICIJA_X` i `POZICIJA_Y` objekta taksi. Tako će prva `Console.WriteLine` naredba ispisati „Pozicija x taksija je 6!“, a druga će ispisati „Pozicija y taksija je 5!“.



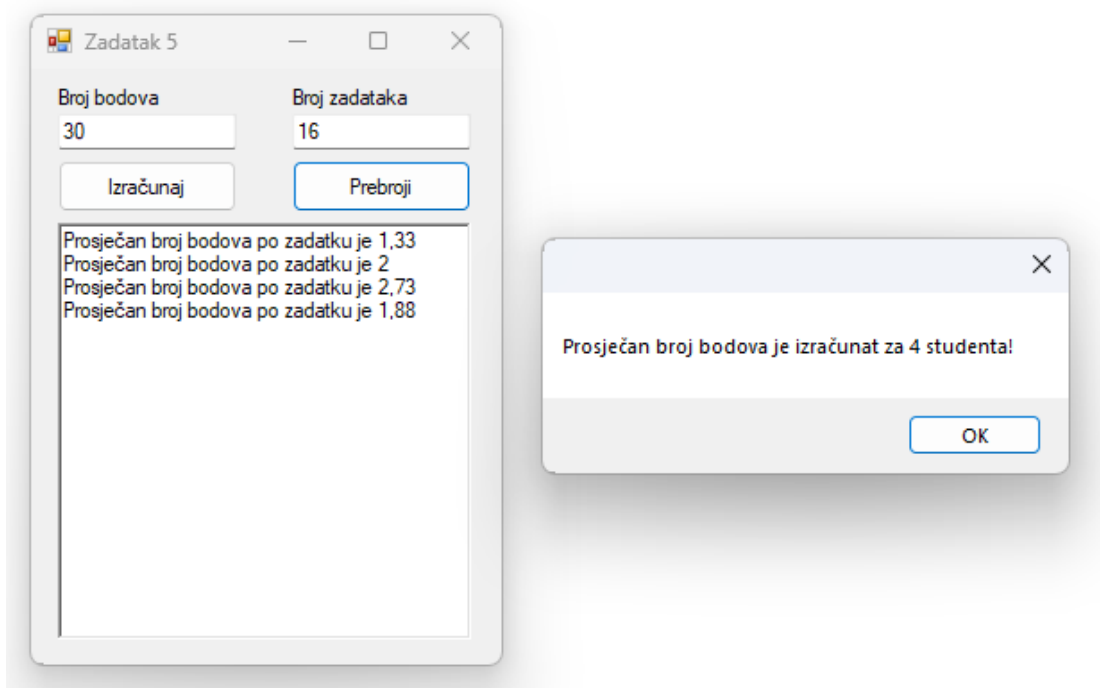
Zadatak 5

Napišite C# kod Windows forme koja izračunava prosječan broj bodova po zadatku na ispitu.

Korisnik unosi ukupan broj bodova studenta na ispitu u *TextBox* naziva *txtBodovi* te broj zadataka na ispitu u *TextBox* naziva *txtZadatci*. Na pritisak gumba naziva *btnIzracunaj* i teksta *Izračunaj* potrebno je izračunati prosječan broj bodova po zadatku te ga ispisati u *RichTextBox* naziva *rtxtIspis*. Svaki novi izračunati prosjek (novi pritisak gumba *Izračunaj*) prikazuje se u zasebnom retku *RichTextBoxa*.

Dodatno, pritiskom gumba naziva *btnPrebroji* i teksta *Prebroji* potrebno je u skočnom prozoru ispisati ukupan broj studenata za koje je prosječan broj bodova po zadatku izračunat. Format ispisa je proizvoljan.

Sučelje, s primjerom unesenih vrijednosti, prikazano je na sljedećoj slici.



```
public partial class Form1 : Form
{

    public Form1()
    {
        InitializeComponent();
    }

    private void btnIzracunaj_Click(object sender, EventArgs e)
    {

    }

    private void btnPrebroji_Click(object sender, EventArgs e)
    {

    }

}
```



Rješenje:

```
public partial class Form1 : Form
{
    int brojStudenata = 0;

    public Form1()
    {
        InitializeComponent();
    }

    private void btnIzracunaj_Click(object sender, EventArgs e)
    {
        brojStudenata++;

        int brojBodova = Convert.ToInt32(txtBodovi.Text);
        int brojZadataka = Convert.ToInt32(txtZadatci.Text);

        double prosjekBodova = brojBodova / Convert.ToDouble(brojZadataka);

        rtxtIspis.Text += "Prosječan broj bodova po zadatku je "
            + prosjekBodova + "\n";
    }

    private void btnPrebroji_Click(object sender, EventArgs e)
    {
        MessageBox.Show("Prosječan broj bodova je izračunat za "
            + brojStudenata + " studenta!");
    }
}
```



Ispit #78 (02.02.23. - grupa B)

Zadatak 1

Napišite C# kod metode **obradaMatrice** koja kao argument prima dvodimenzionalnu matricu brojeva s plivajućim zarezom **M**. Metoda treba vratiti jednodimenzionalno polje brojeva koji predstavljaju operaciju zbrajanja ili oduzimanja po redcima matrice **M**.

Ako je matrica **M** kvadratna izvršava se operacija zbrajanja vrijednosti u recima, inače ako matrica **M** nije kvadratna izvršava se operacija oduzimanja vrijednosti u redcima.

Prilikom operacija zbrajanja, odnosno oduzimanja, u obzir se uzimaju samo vrijednosti matrice koje se nalaze **ili** u intervalu $[-\infty, -10]$ ili u intervalu $[100, 200]$.

Rješenje:

```
static double[] obradaMatrice(double[,] M)
{
    double[] rezultat = new double[M.GetLength(0)];

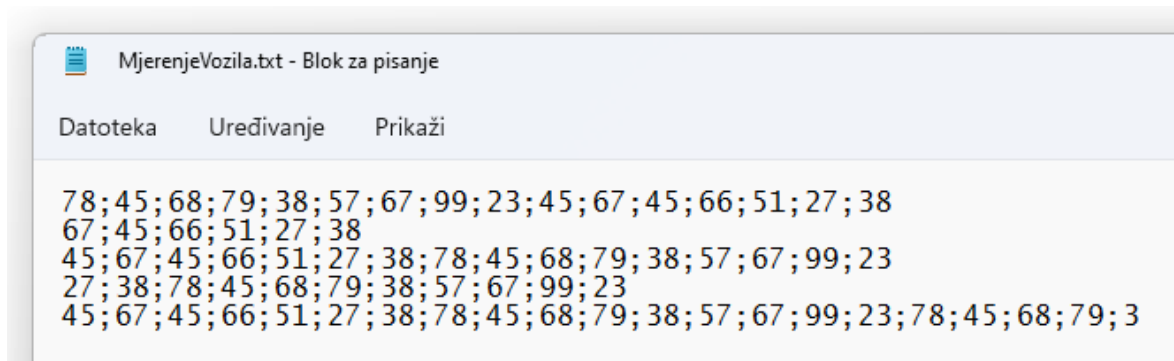
    for (int i = 0; i < M.GetLength(0); i++)
    {
        for (int j = 0; j < M.GetLength(1); j++)
        {
            if (M[i, j] <= -10 || (M[i, j] >= 100 && M[i, j] < 200))
            {
                if (M.GetLength(0) == M.GetLength(1))
                {
                    rezultat[i] += M[i, j];
                }
                else
                {
                    rezultat[i] -= M[i, j];
                }
            }
        }
    }

    return rezultat;
}
```



Zadatak 2

Napišite C# kod programa za učitavanje datoteke **MjerenjeVozila.txt** prikazane na sljedećoj slici smještene u direktoriju **D:**.

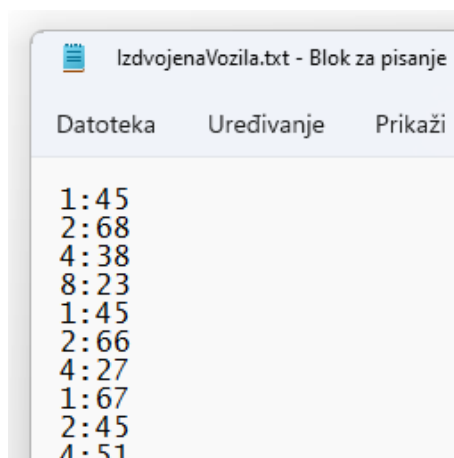


```
MjerenjeVozila.txt - Blok za pisanje
Datoteka   Uređivanje   Prikaži
78;45;68;79;38;57;67;99;23;45;67;45;66;51;27;38
67;45;66;51;27;38
45;67;45;66;51;27;38;78;45;68;79;38;57;67;99;23
27;38;78;45;68;79;38;57;67;99;23
45;67;45;66;51;27;38;78;45;68;79;38;57;67;99;23;78;45;68;79;3
```

U datoteci **MjerenjeVozila.txt** u pojedinom retku su izmjerene brzine vozila (cijeli brojevi) međusobno odvojeni točka-zarezom (;) na određenom dijelu prometnice u određeno vrijeme. Broj zabilježenih brzina po retku je varijabilan.

Program treba u novu datoteku **IzdvojenaVozila.txt**, smještenu u istom direktoriju **D:**, zapisati pojedine brzine iz svakog retka u formatu **X:Y** (u zasebnom retku), gdje znak **X** predstavlja indeks brzine u retku, a znak **Y** vrijednost te brzine. Iz svakog retka datoteke **MjerenjeVozila.txt** uzimaju se brzine na sljedećim indeksima 1, 2, 4, 8, 16, 32, ..., ∞ , odnosno dok god ima elemenata u retku na pojedinom indeksu (pomoć: eksponencijalna funkcija).

Potrebno je koristiti **try-catch** blok. Primjer zapisa u datoteci prikazan je na sljedećoj slici.



```
IzdvojenaVozila.txt - Blok za pisanje
Datoteka   Uređivanje   Prikaži
1:45
2:68
4:38
8:23
1:45
2:66
4:27
1:67
2:45
4:51
```



Rješenje 1:

Prvo rješenje je rađeno na način kako bi zadatak shvatili matematičari: indeks se računa eksponencijalnom funkcijom, tj. potenciranjem broja 2 na 0, 1, 2, 3 itd.

```
static void Main(string[] args)
{
    try
    {
        StreamReader reader = new StreamReader("D:\\MjerenjeVozila.txt");
        StreamWriter writer = new StreamWriter("D:\\IzdvojenaVozila.txt");

        while (!reader.EndOfStream)
        {
            string linija = reader.ReadLine();
            string[] dijeloviLinije = linija.Split(';');
            int eksponent = 0;

            for (int i = 0; i < dijeloviLinije.Length; i++)
            {
                if (i == Math.Pow(2, eksponent))
                {
                    int brojZaIspis = Convert.ToInt32(dijeloviLinije[i]);
                    writer.WriteLine(i + ":" + brojZaIspis);
                    eksponent++;
                }
            }

            reader.Close();
            writer.Flush();
            writer.Close();
        }
        catch (Exception ex)
        {
            Console.WriteLine(ex.Message);
        }

        Console.ReadKey();
    }
}
```



Rješenje 2:

Drugo rješenje je rađeno na način kako bi ga shvatili programeri: indeks kreće od 1 i svaki idući je dva puta veći od prethodnog.

```
static void Main(string[] args)
{
    try
    {
        StreamReader reader = new StreamReader("D:\\MjerenjeVozila.txt");
        StreamWriter writer = new StreamWriter("D:\\IzdvojenaVozila.txt");

        while (!reader.EndOfStream)
        {
            string linija = reader.ReadLine();
            string[] dijeloviLinije = linija.Split(';');

            for (int i = 1; i < dijeloviLinije.Length; i *= 2)
            {
                int brojZaIspis = Convert.ToInt32(dijeloviLinije[i]);
                writer.WriteLine(i + ":" + brojZaIspis);
            }

            reader.Close();
            writer.Flush();
            writer.Close();
        }
        catch (Exception ex)
        {
            Console.WriteLine(ex.Message);
        }

        Console.ReadKey();
    }
}
```



Zadatak 3

Napišite C# kod klase **Nesreca** koja mora imati tri javna atributa: *duzina* (broj s plivajućim zarezom), *sirina* (broj s plivajućim zarezom) i *policajska uprava* (niz znakova).

Klasa dodatno treba imati javnu metodi **ispis** koja ispisuje sadržaj klase u formatu *Duzina: X, sirina: Y i policijska uprava: Z*, pri čemu znakove X, Y i Z treba zamijeniti odgovarajućim vrijednostima.

U glavnom dijelu programa učitajte **20** proizvoljnih instanci klase **Nesreca** koristeći petlju po vlastitom izboru, dodajte ih u listu proizvoljnog naziva i ispišite sadržaj pojedine instance klase u konzolu koristeći napisanu metodu **ispis**.

Rješenje:

```
static void Main(string[] args)
{
    List<Nesreca> listaNesreca = new List<Nesreca>();

    for (int i = 1; i <= 20; i++)
    {
        Console.WriteLine("Unesite dužinu " + i + ". prometne nesreće: ");
        double duzina = Convert.ToDouble(Console.ReadLine());

        Console.WriteLine("Unesite širinu " + i + ". prometne nesreće: ");
        double sirina = Convert.ToDouble(Console.ReadLine());

        Console.WriteLine("Unesite policijsku upravu " + i
            + ". prometne nesreće: ");
        string policijskaUprava = Console.ReadLine();

        Nesreca n = new Nesreca();
        n.duzina = duzina;
        n.sirina = sirina;
        n.policijskaUprava = policijskaUprava;

        listaNesreca.Add(n);
        n.ispis();
    }

    Console.ReadKey();
}

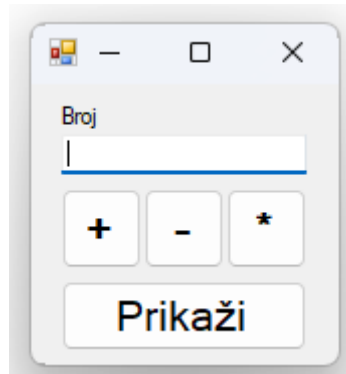
class Nesreca
{
    public double duzina;
    public double sirina;
    public string policijskaUprava;

    public void ispis()
    {
        Console.WriteLine("Dužina: " + duzina + ", širina: " + sirina
            + " i policijska uprava: " + policijskaUprava);
    }
}
```



Zadatak 5

Nadopunite C# kod Windows forme koja radi operacije zbrajanja, oduzimanja i množenja cijelih brojeva. Forma je prikazana na sljedećoj slici.



Početna vrijednost rezultata je **0** (nula). Vrijednost cijelog broja unosi se u *TextBox* naziva *txtBroj*.

Pritiskom na gumb teksta + naziva *btnZbroj* povećava se vrijednost rezultata za uneseni broj.

Pritiskom na gumb teksta – naziva *btnRazlika* smanjuje se vrijednost rezultata za uneseni broj.

Pritiskom na gumb teksta * naziva *btnUmnozak* rezultat se množi s vrijednošću unesenog broja.

Pritiskom na gumb teksta **Prikaži** naziva *btnPrikazi* u skočnom prozoru ispisuje se poruka u formatu *Trenutni rezultat iznosi: X!*, pri čemu znak X predstavlja trenutnu vrijednost rezultata. Primjerice, ukoliko se obavi sljedeći redoslijed operacija: unos 10, zbroj, unos 5, razlika, unos 3 i umnožak - u skočnom prozoru treba se ispisati poruka *Trenutni rezultat iznosi: 15!*.

```
public partial class Form1 : Form
{
    public Form1()
    {
        InitializeComponent();
    }

    private void btnZbroj_Click(object sender, EventArgs e)
    {
    }

    private void btnRazlika_Click(object sender, EventArgs e)
    {
    }

    private void btnUmnozak_Click(object sender, EventArgs e)
    {
    }

    private void btnPrikazi_Click(object sender, EventArgs e)
    {
    }
}
```



Rješenje:

```
public partial class Form1 : Form
{
    int rezultat = 0;

    public Form1()
    {
        InitializeComponent();
    }

    private void btnZbroj_Click(object sender, EventArgs e)
    {
        int x = Convert.ToInt32(txtBroj.Text);
        rezultat += x;
    }

    private void btnRazlika_Click(object sender, EventArgs e)
    {
        int x = Convert.ToInt32(txtBroj.Text);
        rezultat -= x;
    }

    private void btnUmnozak_Click(object sender, EventArgs e)
    {
        int x = Convert.ToInt32(txtBroj.Text);
        rezultat *= x;
    }

    private void btnPrikazi_Click(object sender, EventArgs e)
    {
        MessageBox.Show("Trenutni rezultat iznosi: " + rezultat + "!");
    }
}
```



Ispit #79 (02.02.23. - grupa A)

Zadatak 1

Napišite C# kod konzolne aplikacije koja zbraja realne brojeve unutar intervala $[n-m]$. Granice intervala potrebno je učitati s tipkovnice. Učitani brojevi se zbrajaju sve dok se ne unese broj izvan zadanog intervala.

Provjera da li je učitani broj unutar zadanog intervala vrši se unutar metode *provjeraUnosa* koja prima tri argumenta: učitani broj, donju granicu intervala i gornju granicu intervala. Metoda vraća *bool* vrijednost u ovisnosti da li je učitani broj unutar zadanog intervala ili izvan.

Nakon što je učitani broj izvan zadanog intervala, potrebno je ispisati sumu dosad učitanih brojeva u proizvoljnom formatu unutar glavnog dijela programa.



Rješenje 1:

```
1 static void Main(string[] args)
2 {
3     Console.WriteLine("Unesite donju granicu n:");
4     int n = Convert.ToInt32(Console.ReadLine());
5
6     Console.WriteLine("Unesite gornju granicu m:");
7     int m = Convert.ToInt32(Console.ReadLine());
8
9     double zbroj = 0;
10
11     do
12     {
13         Console.WriteLine("Unesite broj: ");
14         double x = Convert.ToDouble(Console.ReadLine());
15
16         if (provjeraUnosa(x, n, m) == true)
17         {
18             zbroj += x;
19         }
20         else
21         {
22             break;
23         }
24     } while (true);
25
26     Console.WriteLine("Zbroj svih učitanih brojeva: " + zbroj);
27
28     Console.ReadKey();
29 }
30
31 static bool provjeraUnosa(double x, int n, int m)
32 {
33     if (x >= n && x <= m)
34     {
35         return true;
36     }
37
38     return false;
39 }
```



Rješenje 2:

```
1 static void Main(string[] args)
2 {
3     Console.WriteLine("Unesite donju granicu n:");
4     int n = Convert.ToInt32(Console.ReadLine());
5
6     Console.WriteLine("Unesite gornju granicu m:");
7     int m = Convert.ToInt32(Console.ReadLine());
8
9     bool unosJeIspravan = true;
10    double zbroj = 0;
11
12    do
13    {
14        Console.WriteLine("Unesite broj: ");
15        double x = Convert.ToDouble(Console.ReadLine());
16
17        if (provjeraUnosa(x, n, m) == true)
18        {
19            zbroj += x;
20        }
21        else
22        {
23            unosJeIspravan = false;
24        }
25    } while (unosJeIspravan == true);
26
27    Console.WriteLine("Zbroj svih učitanih brojeva: " + zbroj);
28
29    Console.ReadKey();
30 }
31
32 static bool provjeraUnosa(double x, int n, int m)
33 {
34     if (x >= n && x <= m)
35     {
36         return true;
37     }
38
39     return false;
40 }
```



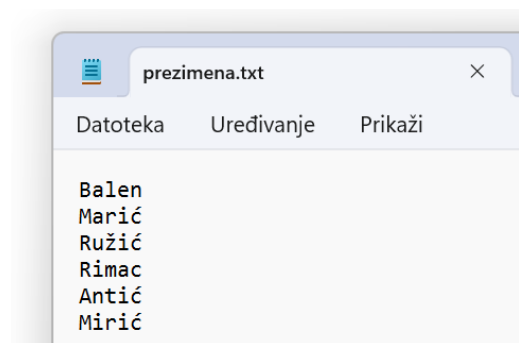
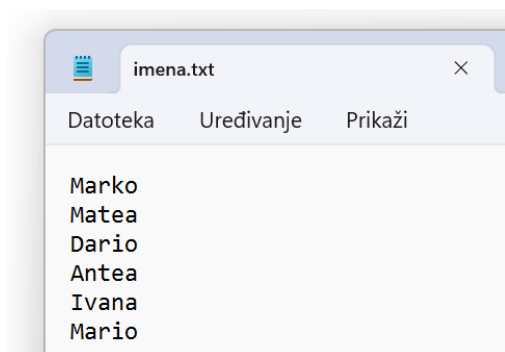
Zadatak 2

Napišite C# kod konzolne aplikacije koja učitava vrijednosti iz datoteka *imena.txt* i *prezimana.txt*. Imena se učitavaju u matricu *ime*, a prezimena u matricu *prezime*. Obje matrice su tipa podataka znaka (*char*). Sva imena i sva prezimena su jednake dužine te je u datotekama točno 6 imena i prezimena.

Nakon učitavanja matrica, inicijali studenata se pohranjuju u listu *inicijali* te ispisuju na konzolu u sljedećem formatu: „Inicijali studenata su: M.B.,M.M.,D.R.,A.R.,I.A.,M.M.“

Potrebno je koristiti *try-catch* blok nad dijelom koda za rad s datotekom, te nije potrebno pisati *Using System.IO*.

Primjer sadržaja datoteka prikazan je na slikama:



Rješenje:

```
1 using System.IO;
2
3 static void Main(string[] args)
4 {
5     char[,] ime = new char[6, 5];
6     char[,] prezime = new char[6, 5];
7
8     try
9     {
10         StreamReader citajIme = new StreamReader("imena.txt");
11         StreamReader citajPrezime = new StreamReader("prezimana.txt");
12
13         for (int i = 0; i < ime.GetLength(0); i++)
14         {
15             string trenutnoIme = citajIme.ReadLine();
16             string trenutnoPrezime = citajPrezime.ReadLine();
17
18             for (int j = 0; j < ime.GetLength(1); j++)
19             {
20                 ime[i, j] = trenutnoIme[j];
21                 prezime[i, j] = trenutnoPrezime[j];
22             }
23         }
24
25         citajIme.Close();
26         citajPrezime.Close();
27
28         List<string> inicijali = new List<string>();
29
30         Console.Write("Inicijali studenata su: ");
31         for (int i = 0; i < ime.GetLength(0); i++)
32         {
33             inicijali.Add(ime[i, 0] + "." + prezime[i, 0] + ".");
34
35             Console.Write(inicijali[i]);
36             if (i < ime.GetLength(0) - 1)
37             {
38                 Console.Write(",");
39             }
40         }
41     }
42     catch (Exception ex)
43     {
44         Console.WriteLine("Greška: " + ex.Message);
45     }
46
47     Console.ReadKey();
48 }
```

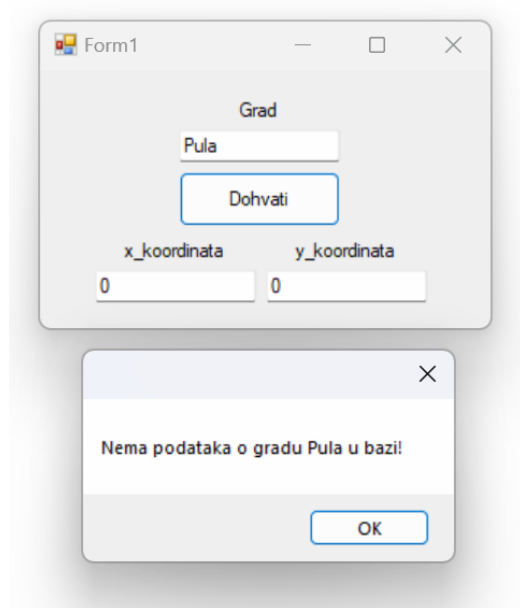


Zadatak 5

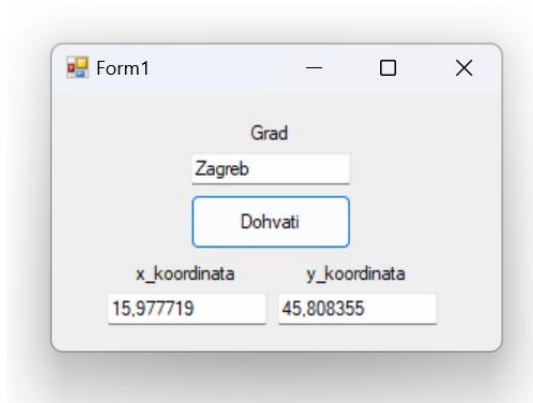
Napišite C# kod windows forme koja dohvaća x i y koordinatu grada na temelju unesenog naziva grada. Naziv grada se unosi *TextBoxom* naziva *txtGrad*.

Na pritisak gumba naziva *btnDohvati* i teksta *Dohvati* provjerava se postoji li uneseni naziv grada u listi *gradovi* te ako postoji u *TextBoxove* naziva *txtXkoord* i *txtYkoord* se ispisuju x i y koordinate grada. Koordinate se nalaze u listama *x_koord* i *y_koord* na istom indeksu na kojem se nalazi naziv odgovarajućeg grada u listi *gradovi*.

Ukoliko naziv grada nije pronađen, prikazuje se skočni prozor s porukom kako je prikazano na sljedećoj slici, a obje koordinate poprimaju vrijednost 0 te se ispisuju u pripadajućim *TextBoxovima*.



Sučelje, s primjerom unesenih vrijednosti grada koji postoji u listi *gradovi*, prikazano je na sljedećoj slici.



```

public partial class Form1 : Form
{
    public Form1()
    {
        InitializeComponent();

        List<string> gradovi = new List<string>() { "Zagreb", "Osijek", "Đakovo",
        "Split", "Fažana", "Gospić" };
        List<double> x_koord = new List<double>() { 15.977719, 18.695038, 18.411567,
        16.440157, 13.806238, 15.380820 };
        List<double> y_koord = new List<double>() { 45.808355, 45.552578, 45.311880,
        53.512867, 44.926796, 44.548078 };

        private void btnDohvati_Click(object sender, EventArgs e)
        {

        }
    }
}

```



Rješenje:

```
1 public partial class Form1 : Form
2 {
3     public Form1()
4     {
5         InitializeComponent();
6     }
7
8     List<string> gradovi = new List<string>() { "Zagreb", "Osijek", "Đakovo",
9         "Split", "Fažana", "Gospić" };
10    List<double> x_koord = new List<double>() { 15.977719, 18.695038, 18.411567,
11        16.440157, 13.806238, 15.380820 };
12    List<double> y_koord = new List<double>() { 45.808355, 45.552578, 45.311880,
13        53.512867, 44.926796, 44.548078 };
14
15    private void btnDohvati_Click(object sender, EventArgs e)
16    {
17        string uneseniGrad = txtGrad.Text;
18        bool gradJePronadjen = false;
19
20        for (int i = 0; i < gradovi.Count; i++)
21        {
22            if (gradovi[i] == uneseniGrad)
23            {
24                txtXkoord.Text = x_koord[i].ToString();
25                txtYkoord.Text = y_koord[i].ToString();
26                gradJePronadjen = true;
27                break;
28            }
29        }
30
31        if (!gradJePronadjen)
32        {
33            txtXkoord.Text = "0";
34            txtYkoord.Text = "0";
35            MessageBox.Show("Nema podataka o gradu " + uneseniGrad + " u bazi!");
36        }
37    }
38 }
```



Ispit #81 (15.06.23. - grupa A)

Zadatak 1

Što će se ispisati u konzolu nakon izvršavanja sljedećeg programskog odsječka? U zadatku je potrebno naznačiti redoslijed i vrijednosti koje se ispisuju.

```
List<string> list = new List<string>();
list.AddRange(new string[] { "Ana", "Marko", "Miro" });
list.InsertRange(list.Count - 2, new string[] { "Tomo", "Orogoro" });
list.Remove("Marko");
int duljina = list[0].Length;
list.RemoveAt(list.Count - duljina);
for (int i = 0; i < list.Count; i++)
{
    int brojac = 0;
    string element = list[i];
    for (int j = 0; j < element.Length; j++)
    {
        if (element[j] == 'o')
        {
            brojac++;
        }
    }
    Console.WriteLine(brojac);
}
```

Rješenje:

Na ekran će se ispisati:

0
3
1



Zadatak 2

Napišite C# kôd metode **provjeraRijeci** koja vraća **istinu ili laž** ovisno o tome da li je riječ predana kao argument ispravna ili ne.

Riječ je ispravna ako je broj znakova u riječi veći od 4 i ako riječ ne sadrži niti jedan od samoglasnika **a, e, i, o i u**.

U glavnom dijelu programa potrebno je kontinuirano učitavati proizvoljne riječi s tipkovnice sve dok se ne učitava neispravna riječ. Za provjeru ispravnosti koristiti metodu *provjeraRijeci*.

Nakon učitavanja pojedine riječi potrebno je ispisati u konzolu da li je ispravna ili ne u formatu kao u primjeru. Primjerice, ako se učitava riječ: **drvnk** program treba ispisati *Učitana riječ drvnk je ispravna!*.

Rješenje:

```
static void Main(string[] args)
{
    do
    {
        Console.WriteLine("Unesite riječ:");
        string rijec = Console.ReadLine();

        if (provjeraRijeci(rijec))
        {
            Console.WriteLine("Učitana riječ " + rijec + " je ispravna!");
        }
        else
        {
            Console.WriteLine("Učitana riječ " + rijec + " nije ispravna!");
            break;
        }
    } while (true);
}

static bool provjeraRijeci(string rijec)
{
    if (rijec.Length <= 4)
    {
        return false;
    }

    rijec = rijec.ToLower();

    for (int i = 0; i < rijec.Length; i++)
    {
        if (rijec[i] == 'a' || rijec[i] == 'e' || rijec[i] == 'i'
            || rijec[i] == 'o' || rijec[i] == 'u')
        {
            return false;
        }
    }

    return true;
}
```



Zadatak 3

Napišite C# kôd klase *Prometnica* koja mora imati pet privatnih atributa: niz znakova kojim se bilježi kategorija prometnice te četiri broja s plivajućim zarezom kojima se bilježe geografske koordinate početne i krajnje točke prometnice (**xPocetak**, **yPocetak**, **xKraj**, **yKraj**).

Klasa treba imati konstruktor koji prima sve attribute klase i postavlja im vrijednosti, te javnu metodu **izracunajDuzinu**. Metoda *izracunajDuzinu* **ne prima argument**, ali izračunava i vraća Euklidsku udaljenost između početne i krajnje točke trenutne instance prometnice, tj. izračunava duljinu prometnice prema sljedećem izrazu:

$$udaljenost(T_1, T_2) = \sqrt{(x_{T_2} - x_{T_1})^2 + (y_{T_2} - y_{T_1})^2}$$

Unutar glavnog dijela programa potrebno je preko tipkovnice učitati sve potrebne vrijednosti (proizvoljne) za stvaranje jedne instance klase, te potom u konzolu ispisati duljinu prometnice te instance koristeći metodu *izracunajDuzinu*. Format ispisa mora biti: *Duljina prometnice kategorije X iznosi Y!* (znakove X i Y potrebno je zamijeniti odgovarajućim vrijednostima).



Rješenje:

```
static void Main(string[] args)
{
    Console.WriteLine("Unesite kategoriju prometnice:");
    string kategorija = Console.ReadLine();
    Console.WriteLine("Unesite x koordinatu početka prometnice:");
    double xPocetak = Convert.ToDouble(Console.ReadLine());
    Console.WriteLine("Unesite y koordinatu početka prometnice:");
    double yPocetak = Convert.ToDouble(Console.ReadLine());
    Console.WriteLine("Unesite x koordinatu kraja prometnice:");
    double xKraj = Convert.ToDouble(Console.ReadLine());
    Console.WriteLine("Unesite y koordinatu kraja prometnice:");
    double yKraj = Convert.ToDouble(Console.ReadLine());

    Prometnica p = new Prometnica(kategorija, xPocetak, yPocetak, xKraj,
    yKraj);
    Console.WriteLine("Duljina prometnice kategorije " + kategorija
        + " iznosi " + p.izracunajDuzinu() + "!");
}

class Prometnica
{
    private string kategorija;
    private double xPocetak;
    private double yPocetak;
    private double xKraj;
    private double yKraj;

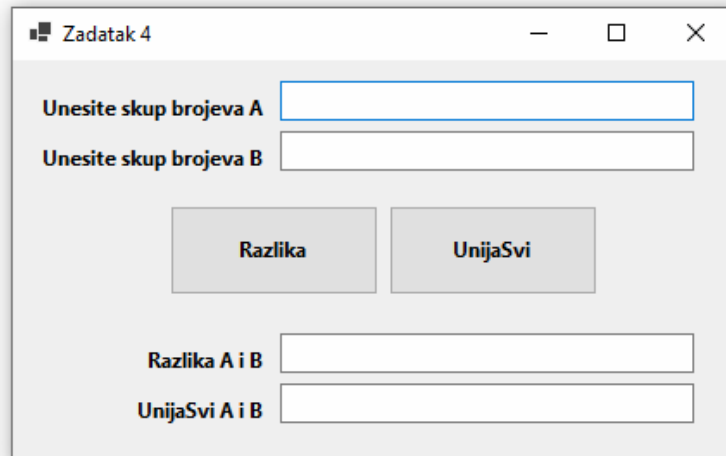
    public Prometnica(string KATEGORIJA, double XPOCETAK, double YPOCETAK,
    double XKRAJ, double YKRAJ)
    {
        kategorija = KATEGORIJA;
        xPocetak = XPOCETAK;
        yPocetak = YPOCETAK;
        xKraj = XKRAJ;
        yKraj = YKRAJ;
    }

    public double izracunajDuzinu()
    {
        return Math.Sqrt(Math.Pow(xKraj - xPocetak, 2)
            + Math.Pow(yKraj - yPocetak, 2));
    }
}
```



Zadatak 4

Nadopunite C# kôd prikazan ispod za windows formu prikazanu na sljedećoj slici:



Forma sadrži 4 *Label*-a s tekstovima prikazanim na slici; 4 *TextBox*-a naziva *TxtSkupaA*, *TxtSkupB*, *TxtRazlika* i *TxtUnijaSvi*; i dva gumba (*Button*-a) naziva *BtnRazlika* i *BtnUnijaSvi* s tekstovima prikazanim na slici.

U *TextBox*-ove *TxtSkupaA* i *TxtSkupB* potrebno je unijeti brojeve skupova A i B (očekuje se da će se za jedan element skupa unositi samo cijeli brojevi od 0 do 9).

Pritiskom na gumb *BtnRazlika* potrebno je izračunati razliku skupa A i B, te ispisati razliku unutar *TextBox*-a *TxtRazlika* (razlika skupa A i B, su svi oni koji su u skupu A, a nisu u skupu B).

Pritiskom na gumb *BtnUnijaSvi* potrebno je ukupnu uniju skupa A i B ispisati unutar *TextBox*-a *TxtUnijaSvi* (unija skupova A i B je skup svih elemenata skupa A i B – čak i onih koji se više puta ponavljaju).

Provjera ispravnosti: A = 789; B = 293; Razlika A i B = 78; Unija Svi A i B = 789293.

```
public partial class Form1 : Form
{
    public Form1()
    {
        InitializeComponent();
    }

    private void BtnRazlika_Click(object sender, EventArgs e)
    {
    }

    private void BtnUnijaSvi_Click(object sender, EventArgs e)
    {
    }
}
```



Rješenje:

```
public partial class Form1 : Form
{
    public Form1()
    {
        InitializeComponent();
    }

    private void BtnRazlika_Click(object sender, EventArgs e)
    {
        string porukaZaIspis = "";

        for (int i = 0; i < TxtSkupA.Text.Length; i++)
        {
            bool slovoPronadjeno = false;

            for (int j = 0; j < TxtSkupB.Text.Length; j++)
            {
                if (TxtSkupA.Text[i] == TxtSkupB.Text[j])
                {
                    slovoPronadjeno = true;
                    break;
                }
            }

            if (!slovoPronadjeno)
            {
                porukaZaIspis += TxtSkupA.Text[i];
            }
        }

        TxtRazlika.Text = porukaZaIspis;
    }

    private void BtnUnijaSvi_Click(object sender, EventArgs e)
    {
        TxtUnijaSvi.Text = TxtSkupA.Text + TxtSkupB.Text;
    }
}
```



Zadatak 5

Što će se ispisati u konzoli nakon izvršavanja sljedećeg programskog odsječka?

```
static void Main(string[] args)
{
    Console.WriteLine(TargetMethod(2));
}
public static int TargetMethod(int n)
{
    if (n % 2 == 0)
    {
        Console.WriteLine(n++);
    }
    if (n > 8)
    {
        return n;
    }
    else
    {
        return TargetMethod(n + 1);
    }
}
```

Rješenje:

Na ekran će se ispisati:

2
4
6
8
9



Ispit #82 (06.07.23. - grupa A)

Napomena: nedostaje peti zadatak!

Zadatak 1

Što će se ispisati u konzolu nakon izvršavanja sljedećeg programskog odsječka?

```
static void Main(string[] args)
{
    ListaBrojeva();
}
public static void ListaBrojeva()
{
    List<int> lista = new List<int>();
    int i;
    for (i = 32; i > 0; i /= 2)
        lista.Add(i--);
    lista.RemoveAt(3);
    lista.InsertRange(i + 2, new int[] { lista[i + 1] / 2,
        lista[lista.Count - 1 - i] / 2 });
    Console.Write(lista[0]);
    for (i = 1; i < lista.Count; i++)
    {
        Console.Write("; " + lista[i]);
    }
}
```

Rješenje:

Na ekran će se ispisati:

32; 15; 7; 0; 7; 1



Zadatak 2

Napišite C# kôd aplikacije koja generira datoteku **data.txt** unutar projekta. Datoteka sadrži 100 slučajno generiranih brzina vozila te ID-eve vozila. Format retka datoteke treba biti **ID;Brzina**.

Brzine su decimalni brojevi u rasponu **[0, 180>**. Generirana brzina treba biti zaokružena na dva decimalna mjesta.

Vrijednost ID-a vozila kreću od **0** te svako novo vozilo ima vrijednost ID-a veću za **1** u usporedbi s prethodnim vozilom.

Nakon popunjavanja, potrebno je pozvati metodu za spremanje sadržaja datoteke, te metodu za zatvaranje datoteke.

Potrebno je u konzolu ispisati najveću generiranu brzinu te pripadni ID vozila. Format ispisa mora biti: *Najveća brzina: X, ID: Y!* (znakove **X** i **Y** potrebno je zamijeniti s odgovarajućim vrijednostima). Potrebno je koristiti **try-catch** blok.

Rješenje:

```
try
{
    StreamWriter writer = new StreamWriter("data.txt");
    Random rnd = new Random();

    double najvecaBrzina = double.MinValue;
    int idNajveceBrzine = -1;

    for (int i = 0; i < 100; i++)
    {
        double brzina = Math.Round(rnd.NextDouble() * 180, 2);
        writer.WriteLine(i + ";" + brzina);
        if (brzina > najvecaBrzina)
        {
            najvecaBrzina = brzina;
            idNajveceBrzine = i;
        }
    }

    writer.Flush();
    writer.Close();

    Console.WriteLine("Najveća brzina: " + najvecaBrzina + ", ID: "
        + idNajveceBrzine + "!");
}
catch (Exception ex)
{
    Console.WriteLine(ex.Message);
}
```



Zadatak 3

Napišite C# kôd klase koja mora imati tri privatna atributa: niz znakova kojima se bilježi status dostave, cijeli broj kojim se bilježi vrijednost kupca te cijeli broj kojim se bilježi vrijednost restorana.

Klasa mora imati konstruktor koji prima sve attribute klase i postavlja im vrijednosti, te javnu metodu **estVrijemeDostave**.

Metoda *estVrijemeDostave* ne prima argumente, ali izračunava očekivano vrijeme dostave hrane. Ako je status dostave **pripremaRobe**, očekivano vrijeme dostave se računa pomoću prvog izraza, u suprotnom pomoću izraza 2, pri čemu *x* predstavlja vrijednost restorana, a *y* vrijednost kupca.

$$(1) \ t = \frac{x^2}{y^{1.5}}$$

$$(2) \ t = \frac{x}{y}$$

Unutar glavnog dijela programa (*Main*) potrebno je inicijalizirati tri liste: *statusi* – sadrži nizove znakova { **pripremaRobe**, **dostavlja** }, *kupci* sadrži cijele brojeve { **2**, **4**, **6** }, i *restorani* – sadrži cijele brojeve { **5**, **10**, **15** }.

Potrebno je stvoriti 100 dostava tako da se svakom atributu nove klase dodijeli slučajna (*Random*) vrijednost iz pripadnih lista. Nakon svakog stvaranja objekta klase *Dostava*, potrebno je ispisati u konzolu očekivano vrijeme dostave koristeći metodu *estVrijemeDostave*. Format ispisa mora biti: *Očekivano vrijeme dostave: X* (znak **X** potrebno je zamijeniti s odgovarajućom vrijednosti).

```
public class Dostava
{
    //Nadopunite kod

}
```



Rješenje:

```
static void Main(string[] args)
{
    Random rnd = new Random();
    List<string> statusi = new List<string>() { "pripremaRobe",
                                                "dostavlja" };
    List<int> kupci = new List<int>() { 2, 4, 6 };
    List<int> restorani = new List<int>() { 5, 10, 15 };

    for (int i = 1; i <= 100; i++)
    {
        Dostava d = new Dostava(statusi[rnd.Next(0, statusi.Count)],
                                kupci[rnd.Next(0, kupci.Count)],
                                restorani[rnd.Next(0, restorani.Count)]);
        Console.WriteLine("Ocekivano vrijeme dostave:"
                          + d.estVrijemeDostave());
    }
}

public class Dostava
{
    private string statusDostave;
    private int vrijednostKupca;
    private int vrijednostRestorana;

    public Dostava(string STATUSDOSTAVE, int VRIJEDNOSTKUPCA,
                    int VRIJEDNOSTRESTORANA)
    {
        statusDostave = STATUSDOSTAVE;
        vrijednostKupca = VRIJEDNOSTKUPCA;
        vrijednostRestorana = VRIJEDNOSTRESTORANA;
    }

    public double estVrijemeDostave()
    {
        if (statusDostave == "pripremaRobe")
        {
            return Math.Pow(vrijednostRestorana, 2) /
                   Math.Pow(vrijednostKupca, 1.5);
        }
        else
        {
            return vrijednostRestorana / Convert.ToDouble(vrijednostKupca);
        }
    }
}
```



Zadatak 4

Napišite C# kôd programa koji preko tipkovnice popunjava matricu cijelih brojeva **M = 3 x 4** s proizvoljnim vrijednostima (cijelim brojevima).

Nakon uspješnog popunjavanja svih elemenata, potrebno je u konzolu ispisati ukupnu i prosječnu učitane vrijednosti i ukupan broj unesenih vrijednosti. Format ispisa mora biti: *Sum: X, Avg: Y, Count: Z* (znakove **X**, **Y**, i **Z** potrebno je zamijeniti odgovarajućim vrijednostima).

Rješenje:

```
int[,] M = new int[3, 4];

int suma = 0;
for (int i = 0; i < M.GetLength(0); i++)
{
    for (int j = 0; j < M.GetLength(1); j++)
    {
        M[i, j] = Convert.ToInt32(Console.ReadLine());
        suma += M[i, j];
    }
}

double prosjek = suma / Convert.ToDouble(M.Length);
Console.WriteLine("Sum: " + suma + ", Avg: " + prosjek
    + ", Count: " + M.Length);
```



Napomena: nedostaju četvrti i peti zadatak!

Zadatak 1

Što će se ispisati u konzolu nakon izvršavanja sljedećeg programskog odsječka?

```
static void Main(string[] args)
{
    ListaSlova();
}
public static void ListaSlova()
{
    List<string> slova = new List<string>() { "A", "O", "C" };
    slova.AddRange(new List<string> { "B", "O", "O", "R" });
    List<string> listaSlova = new List<string>();
    int i = 0;
    try
    {
        for (i = slova.Count; i > 0; i -= i % 2)
        {
            string tempSlovo = slova[--i];
            listaSlova.Add(tempSlovo);
        }

        Console.WriteLine(listaSlova[--i]);
    }
    catch (Exception ex)
    {
        listaSlova.Insert(listaSlova.Count, "T");
    }
    listaSlova.RemoveAt(4);
    Console.Write(listaSlova[0]);
    for (i = 1; i < listaSlova.Count; i += 2)
    {
        Console.Write("; " + listaSlova[i]);
    }
}
```

Rješenje:

Na ekran će se ispisati:

R; O; O



Zadatak 2

Napišite C# kôd aplikacije koja učitava podatke iz lokalne datoteke **data.txt**. Datoteka sadrži *n* zabilježenih brzina vozila te ID-eve vozila. Format retka datoteke: **ID;Brzina**. Brzine su decimalni brojevi. Prvi zapis u datoteci ima ID **0** te svako sljedeće vozilo ima ID vrijednosti veću za **1** u usporedbi s prethodnim zapisom.

Potrebno je učitati sve brzine u listu ili u polje (proizvoljno).

Nakon učitavanja podataka potrebno je zatvoriti otvorenu datoteku te je potrebno u konzolu ispisati najmanju zabilježenu brzinu te pripadni ID vozila. Format ispisa mora biti: *Najmanja brzina: X, ID: Y!* (znakove **X** i **Y** potrebno je zamijeniti s odgovarajućim vrijednostima).

Potrebno je koristiti **try-catch** blok.

Rješenje:

```
try
{
    StreamReader reader = new StreamReader("data.txt");

    List<double> brzine = new List<double>();
    double najmanjaBrzina = double.MaxValue;
    int idNajmanjeBrzine = -1;

    while (!reader.EndOfStream)
    {
        string linija = reader.ReadLine();
        string[] dijeloviLinije = linija.Split(';');

        double trenutnaBrzina = Convert.ToDouble(dijeloviLinije[1]);
        brzine.Add(trenutnaBrzina);

        if (trenutnaBrzina < najmanjaBrzina)
        {
            najmanjaBrzina = trenutnaBrzina;
            idNajmanjeBrzine = Convert.ToInt32(dijeloviLinije[0]);
        }
    }

    reader.Close();

    Console.WriteLine("Najmanja brzina: " + najmanjaBrzina + ", ID: "
        + idNajmanjeBrzine + "!");
}
catch (Exception ex)
{
    Console.WriteLine(ex.Message);
}
```



Zadatak 3

Napišite C# kôd metode *provjeraSimetricnostiMatrice* koja kao argument prima matricu prirodnih brojeva te vraća *bool* vrijednost u ovisnosti o tome da li je matrica simetrična ili ne.

Nakon provjere potrebno je u konzolu ispisati da li je provjerena matrica simetrična ili ne. Format ispisa je proizvoljan.

```
static void Main(string[] args)
{
    //int[,] matrica = new int[n, m];
    bool simetricna = provjeraSimetricnostiMatrice(matrica);
    //Nadopunite kod

}
public static bool provjeraSimetricnostiMatrice(int[,] m)
{
    //Nadopunite kod

}
```

Rješenje:

```
static void Main(string[] args)
{
    //int[,] matrica = new int[n, m];
    bool simetricna = provjeraSimetricnostiMatrice(matrica);
    //Nadopunite kod
    if (simetricna == true)
    {
        Console.WriteLine("Matrica je simetrična!");
    }
    else
    {
        Console.WriteLine("Matrica nije simetrična!");
    }
}
public static bool provjeraSimetricnostiMatrice(int[,] m)
{
    //Nadopunite kod

    for (int i = 0; i < m.GetLength(0); i++)
    {
        for (int j = 0; j < m.GetLength(1); j++)
        {
            if (m[i, j] != m[j, i])
            {
                return false;
            }
        }
    }

    return true;
}
```



Ispit #84 (31.08.23. - grupa A)

Napomena: uspio sam nabaviti samo drugi zadatak

Zadatak 2

Napišite C# kôd klase *GPS_trajektorija* koja mora imati tri privatna atributa: vrijeme (klasa DateTime), oznaku korisnika (niz znakova) i listu GPS točaka (lista decimalnih brojeva).

Klasa treba imati konstruktor koji prima sve attribute klase i postavlja im vrijednosti, te dvije javne metode: *ispis* i *dodaj*.

Metoda *ispis* ne vraća rezultat i ne prima argumente, a treba u konzolu ispisati podatke instance klase *GPS_trajektorija* u formatu: GPS trajektorija zabilježena u vrijeme=VRIJEME sadrži X GPS točaka! (umjesto znakova VRIJEME i X treba napisati odgovarajuće vrijednosti).

Metoda *dodaj* preko argumenata prima dva decimalna broja te ih dodaje u listu GPS točaka.

Rješenje:

```
class GPS_trajektorija
{
    private DateTime vrijeme;
    private string oznakaKorisnika;
    private List<double> GPStocke;

    public GPS_trajektorija(DateTime VRIJEME, string OZNAKAKORISNIKA,
        List<double> GPSTOCKE)
    {
        vrijeme = VRIJEME;
        oznakaKorisnika = OZNAKAKORISNIKA;
        GPStocke = GPSTOCKE;
    }

    public void ispis()
    {
        Console.WriteLine("GPS trajektorija zabilježena u vrijeme=" + vrijeme
            + " sadrži " + GPStocke.Count + " GPS točaka!");
    }

    public void dodaj(double x, double y)
    {
        GPStocke.Add(x);
        GPStocke.Add(y);
    }
}
```

