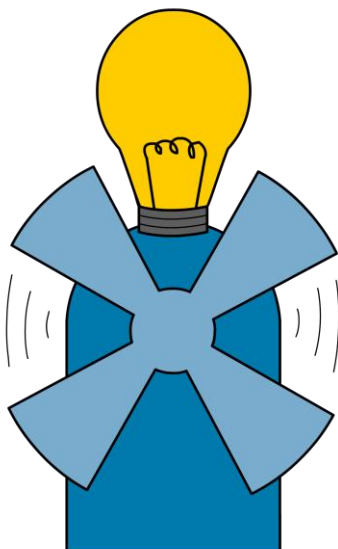


studirko.com.hr

predstavlja:



C# za FPZ - kolokviji (28.04.2025.)

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 - kolokviji“, 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 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 kolokvijima, ali ima i manjih izmjena koje sam radio radi očuvanja konzistentnosti pisanja zadataka.

Skripta će se mijenjati i dopunjavati s kolokvijima 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

Priprema za prvi kolokvij – Zadatci	1
Kolokvij #1-1 (19. 04. 2016. – grupa A)	1
Kolokvij #1-2 (19. 04. 2016. – grupa B)	2
Kolokvij #1-3 (19. 04. 2016. – grupa C).....	3
Kolokvij #1-4 (19. 04. 2016. – grupa D)	4
Kolokvij #1-5 (19. 04. 2016. – grupa E).....	5
Kolokvij #1-6 (25. 04. 2017. – grupa A)	6
Kolokvij #1-7 (25. 04. 2017. – grupa B)	7
Kolokvij #1-8 (25. 04. 2017. – grupa C).....	8
Kolokvij #1-9 (25. 04. 2017. – grupa D)	9
Kolokvij #1-10 (25. 04. 2017. – grupa E).....	10
Kolokvij #1-11 (24. 04. 2018. – grupa A)	11
Kolokvij #1-12 (24. 04. 2018. – grupa B)	12
Kolokvij #1-13 (24. 04. 2018. – grupa C).....	13
Kolokvij #1-14 (24. 04. 2018. – grupa D)	14
Kolokvij #1-15 (24. 04. 2018. – grupa E).....	15
Kolokvij #1-16 (13. 04. 2021. – grupa A)	16
Kolokvij #1-17 (13. 04. 2021. – grupa B)	19
Kolokvij #1-18 (13. 04. 2021. – grupa C).....	21
Kolokvij #1-19 (12. 04. 2022. – grupa A)	23
Kolokvij #1-20 (12. 04. 2022. – grupa B)	28
Kolokvij #1-21 (12. 04. 2022. – grupa C).....	30
Kolokvij #1-22 (12. 04. 2022. – grupa D)	32
Kolokvij #1-23 (11. 04. 2023. – grupa A)	34
Kolokvij #1-24 (11. 04. 2023. – grupa B)	36
Kolokvij #1-25 (11. 04. 2023. – grupa C).....	38
Kolokvij #1-26 (11. 04. 2023. – grupa D)	39
Kolokvij #1-27 (23. 04. 2024. – grupa A)	41
Kolokvij #1-28 (23. 04. 2024. – grupa B)	45
Kolokvij #1-29 (23. 04. 2024. – grupa C).....	50
Kolokvij #1-30 (23. 04. 2024. – grupa D)	53
Priprema za drugi kolokvij – Zadatci	57
Kolokvij #2-24 (02. 06. 2020. – grupa A)	57

Kolokvij #2-25 (01. 06. 2021. – grupa A)	62
Kolokvij #2-26 (01. 06. 2021. – grupa B)	65
Kolokvij #2-27 (01. 06. 2021. – grupa C).....	68
Kolokvij #2-28 (06. 06. 2023. – grupa A)	71
Kolokvij #2-29 (06. 06. 2023. – grupa B)	77
Kolokvij #2-30 (06. 06. 2023. – grupa C).....	79
Kolokvij #2-31 (11. 06. 2024. – grupa A)	81
Kolokvij #2-32 (11. 06. 2024. – grupa B)	86
Kolokvij #2-33 (11. 06. 2024. – grupa C).....	91
Kolokvij #2-34 (11. 06. 2024. – grupa D)	96
Priprema za prvi kolokvij – Rješenja	101
Kolokvij #1-1 (19. 04. 2016. – grupa A)	101
Kolokvij #1-2 (19. 04. 2016. – grupa B)	103
Kolokvij #1-3 (19. 04. 2016. – grupa C).....	105
Kolokvij #1-4 (19. 04. 2016. – grupa D)	107
Kolokvij #1-5 (19. 04. 2016. – grupa E).....	109
Kolokvij #1-6 (25. 04. 2017. – grupa A)	111
Kolokvij #1-7 (25. 04. 2017. – grupa B)	112
Kolokvij #1-8 (25. 04. 2017. – grupa C).....	113
Kolokvij #1-9 (25. 04. 2017. – grupa D)	114
Kolokvij #1-10 (25. 04. 2017. – grupa E).....	115
Kolokvij #1-11 (24. 04. 2018. – grupa A)	116
Kolokvij #1-12 (24. 04. 2018. – grupa B)	118
Kolokvij #1-13 (24. 04. 2018. – grupa C).....	119
Kolokvij #1-14 (24. 04. 2018. – grupa D)	120
Kolokvij #1-15 (24. 04. 2018. – grupa E).....	121
Kolokvij #1-19 (12. 04. 2022. – grupa A)	122
Kolokvij #1-20 (12. 04. 2022. – grupa B)	126
Kolokvij #1-21 (12. 04. 2022. – grupa C).....	132
Kolokvij #1-23 (11. 04. 2023. – grupa A)	137
Kolokvij #1-27 (23. 04. 2024. – grupa A)	144
Kolokvij #1-28 (23. 04. 2024. – grupa B)	154
Kolokvij #1-29 (23. 04. 2024. – grupa C).....	163
Kolokvij #1-30 (23. 04. 2024. – grupa D)	172

Priprema za drugi kolokvij – Rješenja	179
Kolokvij #2-24 (02. 06. 2020. – grupa A)	179
Kolokvij #2-25 (01. 06. 2021. – grupa A)	186
Kolokvij #2-26 (01. 06. 2021. – grupa B)	191
Kolokvij #2-27 (01. 06. 2021. – grupa C).....	195

Priprema za prvi kolokvij – Zadatci

Kolokvij #1-1 (19. 04. 2016. – grupa A)

1. Napišite program koji za učitani polumjer kruga r izračunava površinu i opseg kruga.

Ako polumjer nije ispravno zadan (manji od nule) program ispisuje poruku "Niste unijeli dobar polumjer!" te program završava.

Ako je unesen ispravan polumjer kruga potrebno je ispisati poruku: "Površina i opseg kruga s polumjerom r iznosi: $P = ?$, $O = ?!$ ". Za vrijednost π koristiti 3.14. Broj decimalnih mjesta u ispisu nije bitan.

Npr. ako korisnik preko tipkovnice za polumjer učita vrijednost 2.5 potrebno je ispisati: "Površina i opseg kruga s polumjerom 2.5 iznosi: $P = 19.625$, $O = 15.7!$ ".

2. Napišite metodu ZbrojKvadrata koja kao argument prima dva prirodna broja te vraća zbroj kvadrata predanih brojeva.

Učitavanje dva prirodna broja, poziv metode ZbrojKvadrata te ispis rezultata potrebno je napraviti unutar Main metode.

Za kvadriranje je potrebno koristiti ugrađene matematičke metode.

Npr. ako korisnik preko tipkovnice unese brojeve 5 i 6 potrebno je ispisati: "Zbroj kvadrata brojeva 5 i 6 iznosi: 61!".



Kolokvij #1-2 (19. 04. 2016. – grupa B)

1. Napišite program koji učitava tri realna broja a, b i c.

Ako je broj a veći od broja b program ispisuje zbroj učitanih brojeva, inače ako uvjet ne vrijedi program ispisuje "Nisu učitani ispravni brojevi!", te tu program završava.

Ispis je potrebno formatirati tako da se ispisuju i operandi i operatori. Broj decimalnih mjesta u ispisu nije bitan.

Npr. ako se učitaju redom brojevi $a = 2.5$, $b = 1.5$ i $c = 2$ za zbroj učitanih brojeva (jer je $a > b$) potrebno je ispisati " $2.5 + 1.5 + 2 = 6$ ".

2. Napišite metodu KvadratZbroja koja kao argument prima dva prirodna broja te vraća kvadrat zbroja predanih brojeva.

Učitavanje dva prirodna broja, poziv metode KvadratZbroja te ispis rezultata potrebno je napraviti unutar Main metode.

Za kvadriranje je potrebno koristiti ugrađene matematičke metode.

Ispis mora biti u sljedećem formatu: npr. ako korisnik preko tipkovnice unese brojeve 5 i 6 potrebno je ispisati: "Kvadrat zbroja brojeva 5 i 6 iznosi: 121!"



Kolokvij #1-3 (19. 04. 2016. – grupa C)

1. Napišite program koji učitava tri prirodna broja a, b i c.

Ako je broj a paran broj program ispisuje zbroj brojeva b i c, a ako je a neparan broj ispisuje razliku brojeva b i c.

Ispis je potrebno formatirati tako da se ispisuju i operandi i operatori. Npr. ako se učitaju redom brojevi a = 2, b = 1 i c = 3, budući da je a paran broj potrebno je ispisati: "1 + 3 = 4".

2. Napišite program koji omogućuje korisniku unos lozinke sve dok ona ne bude jednaka vrijednosti lozinke pohranjena u varijabli lozinka, npr. "a1b2". Lozinka može biti bilo koja riječ ili broj ili kombinacija navedenog.

U trenutku kada korisnik upiše ispravnu lozinku treba mu se ispisati poruka: "Lozinka je ispravna!". Za svaku neispravno unesenu lozinku npr. 568 program ispisuje "Unesena lozinka 568 nije ispravna!".



Kolokvij #1-4 (19. 04. 2016. – grupa D)

1. Napišite program koji učitava dva realna broja a i b koji predstavljaju katete pravokutnog trokuta.

Potrebno je provjeriti da li je umnožak učitanih brojeva veći od nule. Ako nije, program ispisuje "Nisu uneseni ispravni brojevi!" te nakon toga program završava. Ako je, potrebno je ispisati površinu pravokutnog trokuta u formatu: npr. ako su učitani brojevi $a = 2.5$ i $b = 2$, potrebno je ispisati "Površina pravokutnog trokuta određenog stanicama $a = 2.5$ i $b = 2$ iznosi $P = 2.5!$ ".

Broj decimalnih mjesta u ispisu nije bitan.

2. Napišite program koji učitava prirodne brojeve u jednodimenzionalno polje s 4 elementa.

Nakon što su učitani svi elementi polja potrebno je ispisati sve elemente polja te njihovu prosječnu vrijednost.

Ispis mora biti u sljedećem formatu: npr. ako su redom učitani brojevi 5 6 7 2, potrebno je ispisati:

"Elementi polja su: 5 6 7 2

Prosječna vrijednost elemenata polja iznosi: 5"

Broj decimalnih mjesta u ispisu nije bitan.



Kolokvij #1-5 (19. 04. 2016. – grupa E)

1. Napišite program koji učitava tri realna broja a, b i c.

Ako je broj c manji od broja a potrebno je ispisati umnožak učitanih brojeva, inače ako uvjet ne vrijedi potrebno je ispisati "Nisu uneseni ispravni brojevi!". Brojevi mogu imati decimalnu točku.

Ispis je potrebno formatirati tako da se ispisuju i operandi i operatori. Broj decimalnih mjesta u ispisu nije bitan.

Npr. ako se učitaju redom brojevi $a = 2.5$, $b = 1.5$ i $c = 2$ za umnožak učitanih brojeva potrebno je ispisati $"2.5 * 1.5 * 2 = 7.5"$.

2. Napišite program koji učitava prirodne brojeve u jednodimenzionalno polje s 4 elementa.

Nakon što su učitani svi elementi polja potrebno je ispisati sve elemente polja te njihov zbroj.

Ispis mora biti u sljedećem formatu: npr. ako su redom učitani brojevi 5 6 7 2, potrebno je ispisati:

"Elementi polja su: 5 6 7 2

Zbroj elemenata polja iznosi: 20"



Kolokvij #1-6 (25. 04. 2017. – grupa A)

1. Napišite metodu ZbrojiOduzmi koja preko argumenata prima dva cijela broja a i b te vraća rezultat operacije zbrajanja i oduzimanja cijelih brojeva unutar intervala $[a, b]$ (uključujuće granice).

Za svaki broj unutar intervala $[a, b]$ provjerava se da li je djeljiv ili s 3 ili s 4. Ako je broj djeljiv s 3 onda se broj pribraja rezultatu, a ako je broj djeljiv s 4 onda se broj oduzima od rezultata. Početna vrijednost rezultata je 0.

Učitavanje dva prirodna broja, poziv metode ZbrojiOduzmi i ispis rezultata potrebno je napraviti unutar Main metode.

Npr. ako korisnik preko tipkovnice unese brojeve $a = 2$ i $b = 7$ potrebno je ispisati: "Rezultat operacije ZbrojiOduzmi unutar intervala $[2, 7]$ iznosi: 5!". Budući da su unutar intervala $[2, 7]$ brojevi djeljivi s tri 3 i 6, a broj djeljiv s četiri je samo 4, rezultat je $0 + 3 - 4 + 6 = 5$.



Kolokvij #1-7 (25. 04. 2017. – grupa B)

1. Napišite program koji učitava 5 realnih brojeva u jednodimenzionalno polje, pritom je potrebno koristiti petlju po izboru.

Nakon što su učitani svi elementi polja potrebno je ispisati sve elemente polja te vrijednost najvećeg elementa polja.

Format ispisa mora biti kao u primjeru. Npr. ako su redom učitani brojevi 5.5, 7.8, 6.2, 4.7 i 9 potrebno je ispisati:

"Elementi polja su:

5.5

7.8

6.2

4.7

9

Vrijednost najvećeg elementa polja iznosi: 9!"



Kolokvij #1-8 (25. 04. 2017. – grupa C)

1. Napišite program koji učitava 9 prirodnih brojeva u dvodimenzionalno polje (matricu 3×3), pritom je potrebno koristiti petlje po izboru. Nakon što su učitani svi elementi matrice potrebno je ispisati samo drugi stupac matrice prema formatu u primjeru.

Npr. ako su redom učitani brojevi po redcima

1,2,3

4,5,6

7,8,9

potrebno je ispisati:

"Elementi drugog stupca matrice su:

2

5

8"



Kolokvij #1-9 (25. 04. 2017. – grupa D)

1. Napišite metodu Izracun koja preko argumenta prima tri prirodna broja a, b i c, te ispisuje rezultat matematičkog izraza:

$$c^8 + \sum_{i=a}^b i$$

Granice a i b su uključujuće. Učitavanje tri prirodna broja i poziv metode Izracun potrebno je napraviti unutar Main metode.

Npr. ako korisnik preko tipkovnice unese brojeve a = 2, b = 5 i c = 1 potrebno je ispisati: "Rezultat zadanog matematičkog izraza iznosi: 15!", budući da je:

$$c^8 = 1^8 = 1$$

$$\sum_{i=a}^b 2 + 3 + 4 + 5 = 14$$



Kolokvij #1-10 (25. 04. 2017. – grupa E)

1. Napišite program koji učitava 3 cijela broja u jednodimenzionalno polje (koristiti petlju po izboru). Nakon što su učitani svi elementi polja potrebno je ispisati sve elemente polja s dvostruko manjom vrijednošću te indeks najmanjeg elementa u polju.

Format ispisa mora biti kao u primjeru. Npr. ako su redom učitani brojevi 5, 7 i -6 potrebno je ispisati:

"Dvostruko manji elementi polja su:

2.5

3.5

-3

Indeks najmanjeg elementa polja iznosi: 2!"



Kolokvij #1-11 (24. 04. 2018. – 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 $P(x, y) = P((x_1 + x_2) / 2, (y_1 + y_2) / 2)$.

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)“.



Kolokvij #1-12 (24. 04. 2018. – grupa B)

1. Napišite program koji učitava imena 5 studenata s tipkovnice u jednodimenzionalno polje koristeći petlju po vlastitom izboru.

Nakon što su učitana sva imena studenata potrebno je ispisati sva unesena imena, te ime koje je najduže (ima najviše slova) i najkraće (ima najmanje slova).

Format ispisa treba biti kao u primjeru. Primjerice ako se redom učitaju imena: Marko, Ana, Luka, Tomislav i Ante potrebno je ispisati:

„Unesena imena studenata su:

Marko

Ana

Luka

Tomislav

Ante

Najduže ime je: Tomislav

Najkraće ime je: Ana“.



Kolokvij #1-13 (24. 04. 2018. – grupa C)

1. Napišite program koji učitava 5 cjelobrojnih vrijednosti bodova s kolokvija u jednodimenzionalno polje.

Potrebno je ispisati koliko je studenata pristupilo kolokviju i koja je prosječna ocjena u formatu kao u primjeru.

Izračun prosječne vrijednosti i ispis rezultata potrebno je obaviti pozivom metode Izracun koja kao argument prima učitano polje te NE vraća rezultat, već ga ispisuje u konzolu.

Primjerice ako se redom učitaju vrijednosti: 10, 15, 20, 0 i 5 potrebno je ispisati: „Kolokviju je pristupilo 5 studenata, a prosječan broj bodova je 10!“.



Kolokvij #1-14 (24. 04. 2018. – grupa D)

1. 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, od zadnjeg prema prvom elementu.

Format ispisa mora biti kao u primjeru. Primjerice ako su redom učitane vrijednosti po redcima:

9,8,7

6,5,4

3,2,1

potrebno je ispisati:

„Elementi dijagonale matrice obrnutim redoslijedom su:

1

5

9“.



Kolokvij #1-15 (24. 04. 2018. – grupa E)

1. Napišite program koji učitava 6 prezimena studenata u jednodimenzionalno polje koristeći petlju po vlastitom izboru.

Nakon što su učitana sva prezimena potrebno je ispisati koliko prezimena ima broj slova manji od 5, koliko ih ima jednak 5, te koliko ih ima većih od 5.

Format ispisa mora biti kao u primjeru. Primjerice, ako se redom učitaju vrijednosti: Čičak, Ivić, Palčić, Braovac, Lovrić, Guja potrebno je ispisati:

„Broj prezimena s manje od 5 slova: 2

Broj prezimena s točno 5 slova: 1

Broj prezimena s više od 5 slova: 3“.



Kolokvij #1-16 (13. 04. 2021. – grupa A)

Napomena: kolokvij je u potpunosti isti kao i ispit koji je pisan 17.04.2021. (grupa A). Postoji mogućnost da su i ostale grupe pisane na ispit, ali to nije potvrđeno 100%.

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.



Kolokvij #1-17 (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.



Kolokvij #1-18 (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 IzracunProsjeaka 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
}
```



Kolokvij #1-19 (12. 04. 2022. – grupa A)

1. Napišite C# kod konzolne aplikacije koja preko tipkovnice učitava brzine na prometnici (cijeli broj) sve dok je vrijednost brzine unutar intervala [0, 200].

U svakoj iteraciji potrebno je učitavanu brzinu dodati u listu *brzine* (dodaju se samo valjane brzine).

Nakon učitavanja potrebno je u konzolu ispisati brzine u listi *brzine* čije su vrijednosti veće od 100 zajedno s njihovim indeksima u formatu *Brzina vrijednosti x veća je od 100, a nalazi se na indeksu y u listi brzine!* (znakove x i y potrebno je zamijeniti odgovarajućim vrijednostima).

Dodatno, na kraju je potrebno ispisati i koliko brzina ima u listi *brzine* čija je vrijednost veća od 100 u formatu *Ukupno ima z brzina u listi brzine čija je vrijednost veća od 100!* (znak z potrebno je zamijeniti odgovarajućom vrijednošću).

2. Napišite C# kod klase *Auto* koja mora imati tri atributa: *marka* (niz znakova, privatna), *OIB* vlasnika (niz znakova, privatna) i *cijena* (cijeli broj, javna).

Potrebno je napisati konstruktor klase koja prima argumente za sve atribute klase i postavlja ih, te se dodatno unutar konstruktora poziva i metoda *Ispis*.

Javna metoda *Ispis* unutar klase *Auto* treba ispisati sadržaj instance klase u formatu *Auto marke MARKA ima vlasnika kojemu je oib OIB i kojem je cijena CIJENA!* (znakove *MARKA*, *OIB* i *CIJENA* potrebno je zamijeniti s odgovarajućim vrijednostima).

Unutar *Main* dijela programa klase *Program* potrebno je napraviti dvije instance klase *auto* s proizvoljnim vrijednostima (**nije potrebno učitavanje s konzole**) te onoj instanci (autu) koji ima veću cijenu smanjiti iznos cijene za 5% i ispisati njegov sadržaj u konzolu koristeći metodu *Ispis*.



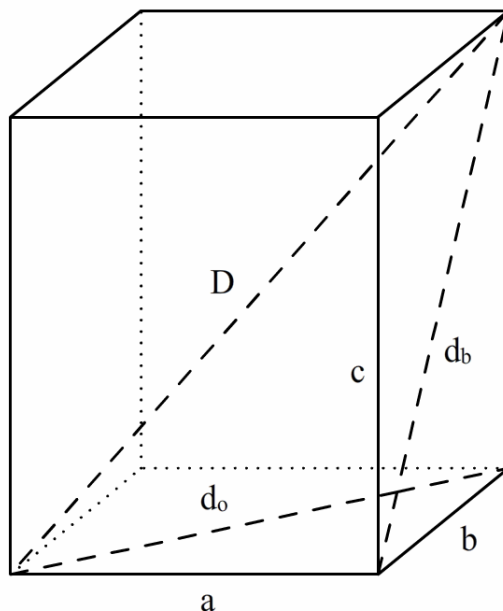
3. Napišite C# kod metode *dijagonaleKvadra* koja preko parametara prima tri brida kvadra a , b i c (brojevi s plivajućim zarezom).

Ukoliko su sve vrijednosti bridova veće od 0 metoda u konzolu ispisuje duljine dijagonala osnovne d_o i bočne strane d_b zaokružene na 2 decimale i **vraća** duljinu prostorne dijagonale D zaokružene na 3 decimale. Inače ništa ne ispisuje, a **vraća** vrijednost -1.

Format ispisa dijagonala osnovne i bočne strane treba biti *Za kvadar zadan bridovima $a=A$, $b=B$ i $c=C$ osnovna dijagonala iznosi $d_o=DO$, a bočna dijagonala iznosi $d_b=DB$* (znakove A , B , C , DO i DB potrebno je zamijeniti s odgovarajućim vrijednostima).

Jednadžbe za izračun dijagonala dane su izrazima navedenim uz sliku kvadra.

Unutar *Main* metode potrebno je 5 puta petljom po vlastitom izboru učitati vrijednosti bridova kvadra (a , b i c) s konzole i svaki put u proizvoljnom formatu ispisati prostornu dijagonalu kvadra za učitane bridove (ono što vrati metoda *dijagonaleKvadra*).



$$d_o = \sqrt{a^2 + b^2}$$

$$d_b = \sqrt{a^2 + c^2}$$

$$D = \sqrt{d_o^2 + c^2}$$

4. Što će se ispisati nakon izvršavanja sljedećeg programskog odsječka?

```
1 int[,] matrica = new int[4, 3] { { 1, 2, 3 }, { 3, 4, 2 },
2                                     { 5, 6, 7 }, { 1, 2, 3 }, };
3
4 for (int j = matrica.GetLength(1) + 1; j > 2; j--)
5 {
6     for (int i = -1; i <= matrica.GetLength(0) - 2; i++)
7     {
8         if (i + 2 == j - 1)
9         {
10             Console.WriteLine(matrica[i + 1, j - 2] + " ");
11         }
12     }
13 }
14 Console.WriteLine(matrica.Length);
```



5. Što će se ispisati nakon izvršavanja sljedećeg programskog odsječka?

```
1 static void Main(string[] args)
2 {
3     int a = (24 % 15) / 2;
4     int b = a + 3 * a;
5     M1(b, a);
6     Console.WriteLine(a + "," + b);
7 }
8 static void M1(int a, int b)
9 {
10     a = b + 1;
11     Console.WriteLine(a + "," + b);
12     M3(a, b);
13 }
14 static void M2(int a, int b)
15 {
16     a = 3;
17     b = a;
18     Console.WriteLine(a + "," + b);
19 }
20 static void M3(int a, int b)
21 {
22     Console.WriteLine(b + "," + a);
23     Console.WriteLine(b + "," + ++a);
24 }
```



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

```
1 int a = 7 % 4;
2 int b = 5 + a;
3 int c = 0, d = 1;
4 do
5 {
6     if (a >= 2 && (b != 0 ^ d == 1))
7     {
8         c -= a;
9         d = 2 * 10;
10        b -= b;
11    }
12    else
13    {
14        c += a;
15        d = a * 3 + 2;
16        a--;
17    }
18    Console.WriteLine(a + "," + b + "," + c + "," + d);
19 } while (a + b != 2);
```



7. Što će se ispisati nakon izvršavanja sljedećeg programskog odsječka?

```
1 static void Main(string[] args)
2 {
3     PravokutanTrokut p = new PravokutanTrokut(8, 10);
4     Console.WriteLine(p.izracunajPovrsinu(4, 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)
14     {
15         A = a;
16         B = b;
17         povrsina = -1;
18         ispis();
19     }
20     public double izracunajPovrsinu(double b, double a)
21     {
22         double POVRSINA = (b * a) / 2;
23         ispis();
24         return POVRSINA;
25     }
26     public void izracunajPovrsinu()
27     {
28         povrsina = (A * B) / 2;
29         ispis();
30     }
31     public void ispis()
32     {
33         Console.WriteLine(povrsina);
34     }
35 }
```



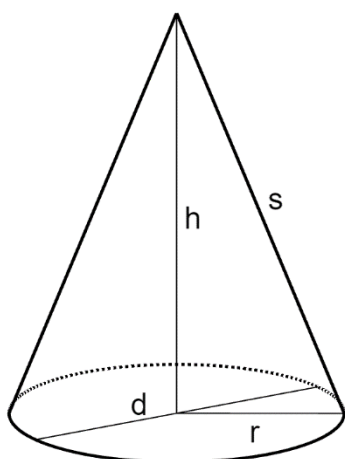
Kolokvij #1-20 (12. 04. 2022. – grupa B)

1. Napišite C# kod metode *IzracunZaStozac* koja kao argumente prima izbor unesen od korisnika (oplošje, volumen ili polumjer plašta) (niz znakova), polumjer stošca r i visinu stošca h (brojevi s plivajućim zarezom).

Ovisno što je predano u parametru *izbor* metoda treba u konzolu ispisati poruku u formatu kao u primjeru. Ukoliko je za izbor predana vrijednost volumen, a za polumjer i visinu su predane vrijednosti 2.5 i 5.2, metoda u konzolu treba ispisat *Volumen stošca određenog polumjerom r=2.5 i visinom h=5.2 iznosi V=34.033920413889426!*.

Jednadžbe potrebne za izračun prikazane su uz sliku stošca.

U *Main* metodi potrebno je 6 puta s tipkovnice učitati izbor (oplošje, volumen ili polumjer plašta) i dimenzije stošca (polumjer i visinu) koristeći petlju po vlastitom izboru, te svaki put pozvati metodu *IzracunZaStozac* za učitane vrijednosti.



$$V = \frac{1}{3}\pi r^2 h$$

$$O = \pi r(r + s)$$

$$s = \sqrt{r^2 + h^2}$$

V – obujam

O – oplošje

r – polumjer

s – polumjer plašta

h – visina



2. Napišite C# kod klase *Dostavljac* koja sadrži privatne attribute ime (niz znakova), prezime (niz znakova) i listu masa paketa koje dostavljač treba dostaviti (lista cijelih brojeva).

Konstruktor klase prima argumente imena i prezimena, postavlja odgovarajuće vrijednosti atributa imena i prezimena, a listu masa paketa inicijalizira (u početku je lista prazna).

Unutar klase *Dostavljac* potrebno je napisati dvije javne metode: *DodajPaket* i *Ispis*.

Metoda *DodajPaket* preko argumenata prima masu paketa koju je potrebno dodati u listu mase paketa.

Metoda *Ispis* ispisuje sadržaj instance klase u formatu kao primjeru. Primjerice, neka je kreiran dostavljač čije je ime *Pero*, prezime *Perić* i u listi masa paketa ima vrijednosti 1, 5, 2 i 3, u konzolu je potrebno ispisati *Dostavljač Pero Perić je dostavio ukupno 4 paketa čije su mase redom: 1 5 2 3!*.

Unutar *Main* dijela programa potrebno je s konzole učitati ime i prezime dostavljača, kreirati instancu klase dostavljač s odgovarajućim vrijednostima, te potom tom dostavljaču (njegovoj listi masa paketa) dodati 3 mase paketa proizvoljnih iznosa (mase paketa **ne moraju** se učitati s konzole). Na kraju je potrebno pozvati metodu *Ispis* za kreiranog dostavljača.

3. Napišite C# kod konzolne aplikacije koja učitava n (broj redaka) i m (broj stupaca) vrijednosti s konzole koji određuju matricu $A = n \times m$.

S konzole je potrebno popuniti matricu A s ID-evima vozila (**niz znakova**).

Potom je potrebno ispisati sadržaj matrice unazad, od zadnjeg elementa u matrici prema prvom elementu redak po redak, u formatu kao u primjeru. Sadržaj koji se treba ispisati u svakom koraku je vrijednost elementa matrice (ID), pozicija elemenata u matrici (indeks retka i indeks stupca) i koliko ima znakova u ID-u.

Na kraju je u proizvoljnom formatu potrebno ispisati i koliko ima ID-eva u matrici čiji je prvi znak jednak znaku 1 (jedan znak).

Primjerice, ako se učitava matrica:

1	3	4	15
18	23	6	999
36	90	81	7
877	123459	44	11

Potrebno je ispisati:

ID 11 nalazi se na indeksu retka 3 i indeksu stupca 3 te ukupno ima 2 znaka!

ID 44 nalazi se na indeksu retka 3 i indeksu stupca 2 te ukupno ima 2 znaka!

ID 123459 nalazi se na indeksu retka 3 i indeksu stupca 1 te ukupno ima 6 znakova!

ID 977 nalazi se na indeksu retka 3 i indeksu stupca 0 te ukupno ima 3 znaka!

ID 7 nalazi se na indeksu retka 2 i indeksu stupca 3 te ukupno ima 1 znaka!

...

ID 1 nalazi se na indeksu retka 0 i indeksu stupca 0 te ukupno ima 1 znaka!

Ukupno ima 5 ID-eva koji počinju znakom 1!



Kolokvij #1-21 (12. 04. 2022. – grupa C)

1. Napišite C# kod metode *sumaZnamenkiBroja* koja kao argument prima cijeli broj, u konzolu ispisuje znamenku po znamenku predanog broja (svaku u zasebnom retku) i kao rezultat **vraća** zbroj znamenki predanog broja.

U *Main* metodi potrebno je učitavati brojeve sve dok se učitavaju brojevi čija je suma znamenki unutar intervala $[0, 15]$. Pomoć: ostatak dijeljenja s 10 i cjelobrojno dijeljenje s 10.

Primjerice, ukoliko se u nekoj iteraciji unese broj 6784, metoda treba ispisati:

Znamenke broja 6784 su:

4
8
7
6

2. Napišite C# kod programa koji prebrojava koliko puta ste se nalazili na pojedinom mjestu matrice. Prvo je potrebno kreirati matricu $M = 10 \times 10$ (cijelih brojeva).

S tipkovnice je potrebno kontinuirano učitavati znak po znak, pri čemu se očekuje da se mogu unositi samo znakovi *U*, *D*, *L* i *R* (pomoć: učitani znak spremite u varijablu tipa niza znakova).

Početna pozicija unutar matrice je $(i, j) = (5, 5)$ (indeks retka i indeks stupca).

U svakoj iteraciji pozicija se pomiče za vrijednost 1 (jedan). *U* predstavlja pomicanje gore, *D* pomicanje dolje, *L* pomicanje u lijevo, a *R* pomicanje u desno. Učitavanje znakova s konzole traje dok god se trenutna pozicija nalazi unutar matrice.

Prilikom pomicanja pozicije potrebno je bilježiti koliko smo puta bili na pojedinoj poziciji u matrici (početna zadana pozicija se ne broji kao posjeta).

Na kraju je potrebno u konzolu ispisati cijelu matricu u formatu kao u primjeru. Primjerice, ako se redom učitavaju vrijednosti U, U, U, D, L, L, R, R, R, R, U, U, U, U, treba se ispisati matrica (početna pozicija označena je crvenom bojom).

0	0	0	0	0	0	0	1	0	0
0	0	0	0	0	0	0	1	0	0
0	0	0	0	0	0	0	1	0	0
0	0	0	1	2	3	1	1	0	0
0	0	0	0	0	1	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0



3. Napišite C# kod konzolne aplikacije koja računa dobit aviokompanije.

Potrebno je napraviti klasu *AvioKompanija* koja sadrži atribute ime prijevoznika (niz znakova, javan), broj sjedala (cijeli broj, javan), cijena karte (broj s plivajućim zarezom, privatan).

Klasa *AvioKompanija* **nema konstruktor**, ali ima dvije javne metode *dohvatiCijenuKarte* i *postaviCijenuKarte*. Metoda *dohvatiCijenuKarte* **vraća** vrijednost atributa cijena karte, dok metoda *postaviCijenuKarte* preko parametra koji joj se predaje postavlja vrijednost atributa cijena karte.

Unutar *Main* dijela programa klase *Program* potrebno je kreirati 2 proizvoljna avio prijevoznika (instance klase *AvioKompanija*), i popuniti sve atribute klasa na temelju vrijednosti koje se učitavaju s konzole. Nakon učitavanja vrijednosti i punjenja atributa klasa, izračunajte i ispišite ostvarenu dobit za svakog prijevoznika prema sljedećem izrazu:

$$\text{ostvarenaDobit} = \text{BrojSjedala} * \text{CijenaKarte}$$

Format ispisa treba biti *Dobit prijevoznika X iznosi Y!* (znakove X (X predstavlja ime prijevoznika) i Y potrebno je zamijeniti s odgovarajućim vrijednostima).



Kolokvij #1-22 (12. 04. 2022. – grupa D)

1. Napišite C# kod konzolne aplikacije koja učitava n (broj redaka) i m (broj stupaca) vrijednosti s konzole koji određuju matricu $M = n \times m$. S konzole je potrebno popuniti matricu M brojevima s plivajućim zarezom.

Nakon što se učitava matrica potrebno je pronaći najmanju i najveću vrijednost na glavnoj dijagonali matrice, te dodatno provjeriti da li je matrica simetrična. Format ispisa treba biti kao u primjeru.

Matrica je simetrična ako vrijedi da je element na indeksu matrice (i, j) jednak elementu na indeksu matrice (j, i) **i to za sve elemente matrice**, odnosno da je donja trokutasta matrica jednaka gornjoj trokutastoj matrici.

Pomoć: početno se može pretpostaviti da je matrica simetrična, te onda ukoliko se ispostavi da neke odgovarajuće vrijednosti u donjoj i gornjoj trokutastoj matrici nisu iste, postavlja se da matrica nije simetrična. Ako se primjerice učitava matrica:

1.1	2.3	4.7	5.6
2.3	2.7	3.8	4.2
4.7	3.8	1.2	7.0
5.6	4.2	7.0	8.3

Potrebno je ispisati:

Najveća vrijednost na glavnoj dijagonali matrice je 8.3, dok je najmanja vrijednosti 1.1!

Matrica $M=4 \times 4$ je simetrična!



2. Napišite C# kod klase *PravokutanTrokut* koja sadrži privatne attribute stranice *a*, stranice *b* i površine (sve brojevi s plivajućim zarezom). Stranice *a* i *b* predstavljaju katete pravokutnog trokuta.

Konstruktor klase prima argumente stranica *a* i *b*, postavlja odgovarajuće vrijednosti stranica *a* i *b*, a atribut površine postavlja na vrijednost -1.

Unutar klase *PravokutanTrokut* potrebno je napisati tri javne metode.

Prva metoda *izracunajPovrsinu* kao parametre prima stranice trokuta *a* i *b* te računa površinu trokuta i **vraća** izračunatu površinu.

Druga metoda *izracunajPovrsinu* (jednako se zove kao i prva), ne prima parametre, te na temelju vrijednosti atributa stranica *a* i *b* računa površinu pravokutnog trokuta i rezultat sprema u atribut površine.

Treća metoda *Ispis* ispisuje površinu pravokutnog trokuta u formatu *Povrsina pravkuntog trokuta zadanog katetama a=A i b=B iznosi POVRŠINA!* (znakove *A*, *B* i *POVRŠINA* potrebo je zamijeniti s odgovarajućim vrijednostima).

Unutar *Main* dijela programa potrebno je napraviti sljedeće korake:

1. s konzole učitati stranice trokuta *a* i *b*,
2. kreirati instancu klase *PravokutanTrokut* s odgovarajućim učitanim vrijednostima,
3. pozvati metodu *Ispis* nad kreiranom instancom klase *PravokutanTrokut* – unutar komentara napisati koja će se vrijednost ispisati za površinu,
4. pozvati metodu *izracunajPovrsinu* nad kreiranom instancom klase *PravokutanTrokut* koja kao parametre prima proizvoljne vrijednosti stranica *a* i *b* (vrijednosti se **ne moraju** učitati s konzole) i rezultat koji vrati metoda ispisati u konzolu u proizvoljnom formatu,
5. pozvati metodu *Ispis* nad kreiranom instancom klase *PravokutanTrokut* – unutar komentara napisati koja će se vrijednost ispisati,
6. pozvati metodu *izracunajPovrsinu* nad kreiranom instancom klase *PravokutanTrokut* koja ne prima parametre.

3. Napišite C# kod metode *provjeraSamoglasnika* koja kao argument prima rečenicu (niz znakova) i u konzolu ispisuje koliko ima pojavljivanja svakog od znakova (samoglasnika) *a*, *e*, *i*, *o* i *u* (**samo mala slova**), te dodatno koliko ima ukupno znakova u rečenici i koliko ima ukupno samoglasnika u rečenici. Format ispisu trebaju biti kao u primjeru.

U *Main* metodi potrebno je napraviti polje naziva *recenice* koje se sastoji od 3 elementa (za 3 rečenice – nizovi znakova), te preko konzole učitati sadržaj polja s petljom po vlastitom izboru i prilikom svakog učitavanja pozvati metodu *provjeraSamoglasnika* nad učitanim nizom znakova (rečenicom).

Primjerice, ukoliko se u nekoj od iteracija učita rečenica mi volimo programiranje! Potrebno je ispisati *U rečenici ##mi volimo programiranje!## ima ukupno 24 znaka od čega su 9 samoglasnici i to redom a(2), e(1), i(3), o(3) i u(0)!*.



Kolokvij #1-23 (11. 04. 2023. – grupa A)

1. Napišite C# kod programa koji preko tipkovnice učitava broj redaka **n** i broj stupaca **m** matrice **M**. Potom program treba stvoriti matricu **M** brojeva s plivajućim zarezom i popuniti ju proizvoljnim vrijednostima s konzole.

Program treba u jednodimenzionalno polje brojeva **P** spremiti rezultat operacija zbrajanja po recima ili stupcima matrice **M**. Ako je matrica **M** kvadratna izvršava se operacija zbrajanja vrijednosti u stupcima, inače ako matrica **M** nije kvadratna izvršava se operacija zbrajanja vrijednosti po recima. Prilikom operacije zbrajanja u obzir se uzimaju samo vrijednosti matrice koje se nalaze u intervalu [0, 200>.

Na kraju je potrebno ispisati sadržaj polja **P** u proizvoljnom formatu.

2. Napišite C# kod klase **Nasreca** koja mora imati tri javna atributa: *dužina* (broj s plivajućim zarezom), *sirina* (broj s plivajućim zarezom) i *policajska uprava* (niz znakova).

Klasa dodatno treba imati 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 **2** proizvoljne instance klase **Nasreca** koristeći **petlju** po vlastitom izboru, dodajte ih u listu proizvoljnog naziva i ispišite sadržaj pojedine instance klase u konzolu koristeći napisanu metodu **ispis**.

3. Napišite C# kod 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!*.

4. Što će se ispisati nakon izvršavanja sljedećeg programskog odsječka?

```
1 List<int> lista = new List<int>();
2 for (int i = 32; i > 0; i = i / 2)
3 {
4     lista.Add(i);
5 }
6 lista.RemoveAt(lista.Count - 3);
7 lista.Insert(0, lista.Count);
8 lista.AddRange(new int[] { -2, 0, 3 });
9 for (int i = 1; i < lista.Count; i+=2)
10 {
11     Console.WriteLine(lista[i]);
12 }
```



5. Što će se ispisati nakon izvršavanja sljedećeg programskog odsječka?

```
1 string s1 = " Marko ";
2 string s2 = "Mirko";
3 string s = s1 + " " + s2;
4 int x1 = 0;
5 int x2 = 0;
6 int x3 = 0;
7 int x4 = 0;
8 for (int i = 0; i < s.Length; i++)
9 {
10     if (s[i] == ' ')
11     {
12         x1++;
13     }
14     else if (s[i] == 'O')
15     {
16         x2++;
17     }
18     if (s[i] == 'M')
19     {
20         x3++;
21     }
22     else
23     {
24         x4++;
25     }
26 }
27 Console.WriteLine(x1);
28 Console.WriteLine(x2);
29 Console.WriteLine(x3);
30 Console.WriteLine(x4);
```



Kolokvij #1-24 (11. 04. 2023. – grupa B)

1. Napišite C# kod konzolne aplikacije koja učitava cijeli broj **n** s tipkovnice, te potom u listu naziva **taksiji** dodaje **n** instanci klase **Taxi** pri čemu se vrijednost atributa pojedine instance učitavaju s konzole.

Nakon učitavanja taksija u listu **taksiji**, potrebno je pronaći taksi koji je najbliži poziciji (10, 10) i ispisati njegovog vozača u konzolu u proizvoljnom formatu. Za izračun udaljenosti koristiti euklidsku udaljenost danu sljedećim izrazom:

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

```
1 class Taxi
2 {
3     public string REGISTRACIJA;
4     public string VOZAC;
5     private double POZICIJA_X;
6     private double POZICIJA_Y;
7     public void postaviX(double x)
8     {
9         POZICIJA_X = x;
10    }
11    public void postaviY(double y)
12    {
13        POZICIJA_Y = y;
14    }
15    public double dohvatiX()
16    {
17        return POZICIJA_X;
18    }
19    public double dohvatiY()
20    {
21        return POZICIJA_Y;
22    }
23 }
```

2. Napišite C# kod programa koji preko tipkovnice popunjava matricu niza znakova **M = 2 x 3** s proizvoljnim vrijednostima (riječima).

Program treba prebrojati koliko puta se u riječima u matrici pojavljuje bilo koji od znakova **b**, **l** ili **a**, uz uvjet da je riječ kraća od **6** znakova.

Za svaku riječ matrice treba ispisati koliko znakova zadovoljava navedeni kriterij u formatu: *Riječ RIJEC sadrži X znakova b, l ili a.* (znakove **RIJEC** i **X** potrebno je zamijeniti odgovarajućim vrijednostima). Pomoć: ukoliko riječ ima više ili jednako 6 znakova, broj pojavljivanja znakova **b**, **l** ili **a** je 0.



3. Napišite C# kod programa koji s konzole kontinuirano učitava tri volumena paketa u proizvodnom pogonu u m³: v_1 , v_2 i v_3 (cijeli brojevi).

Potom se na paletu maksimalne zapremnine 100 m³ dodaje samo onaj paket koji od trenutno tri učitana paketa ima najveći volumen. Pritom je u konzolu potrebno ispisati volumen dodanog paketa i koliko je preostalo slobodnog mjesta na paleti u formatu: *Volumen dodanog paketa iznosi X, a preostalo je Y mjesta!* (znakova X i Y potrebno je zamijeniti odgovarajućim vrijednostima).

Učitavanje je potrebno ponavljati sve dok se najveći paket može dodati na paletu, odnosno dok ima mjesta na paleti da se doda najveći paket.

Za odluku o tome koji je paket najveći potrebno je koristiti metodu **najveci** koja kao argument prima **volumene tri paketa** i **vraća** onaj paket koji je **najveci**.

Na kraju je u proizvoljnom formatu potrebno ispisati koliko je paketa stavljeno na paletu.



Kolokvij #1-25 (11. 04. 2023. – grupa C)

1. Napišite C# kod programa koji kontinuirano učitava podatke vozila: **duljinu** (broj s plivajućim zarezom) i **brzinu** (cijeli broj) izmjerene radarom koji je postavljen na prometnici. Učitavanje podataka o vozilu treba ponavljati dok god je duljina vozila **veća od 0** ili dok je brzina **veća od 0**.

U svakom koraku učitavanja, potrebno je podatke vozila spremati u dvije liste i to: duljine vozila u listu **duljine**, a brzine vozila u listu **brzine**.

Nakon završetka učitavanja svih vozila potrebno je u konzolu ispisati koliko ima elemenata (vozila) u polju **duljine** u proizvoljnom formatu.

Na kraju je potrebno s konzole učitati proizvoljan indeks **i** te je potrebno ispisati duljinu i brzinu vozila određenog indeksom **i** samo ako je vrijednost indeksa **i** ispravna.

2. Napišite C# kod metode **Prebroji** koja kao argument prima dva polja cijelih brojeva **brojevi** i **dozvoljeni**. Program treba prebrojati koliko ima elemenata u polju **brojevi** koji su dozvoljeni – element je validan ako se pojavljuje u polju **dozvoljeni**. Metoda treba **vratiti** koliko takvih elemenata ima.

Unutar glavnog dijela programa potrebno je učitati vrijednost veličina polja **n** i **m** te stvoriti polje **brojevi** veličine **n** i polje **dozvoljeni** veličine **m**.

Potom je potrebno popuniti polja proizvoljnim vrijednostima, iskoristiti metodu **Prebroji** i na kraju u formatu kao u primjeru ispisati rezultat.

Primjerice neka polje **brojevi** ima vrijednosti 4, 5, 4, 2, i 4, a polje **dozvoljeni** vrijednosti 4, 9 i 2, u konzolu se treba ispisati *Broj dozvoljenih vrijednosti u polju brojevi iznosi: 4!*.

3. Napišite 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 javnu metodu **izracunajDuzinu**.

Metoda **izracunajDuzinu** **ne prima argumente**, 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 metodi **izracunajDuzinu**. Format ispisa mora biti: *Duljina prometnice kategorije X iznosi Y!* (znakove X i Y potrebno je zamijeniti odgovarajućim vrijednostima).



Kolokvij #1-26 (11. 04. 2023. – grupa D)

1. Napišite C# kod dvije klase **Lokacija** i **PrometnaNesreca**.

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

Klasa *PrometnaNesreca* sadrži javne atribute **šifra** (cijeli broj) koji predstavlja jedinstveni identifikator nesreće i atribut **LokacijaNesrece** (tip klase *Lokacija*) koji predstavlja mjesto (lokaciju) gdje se nesreća dogodila. Klasa *PrometnaNesreca* treba imati i javnu metodu **ispis** koja ispisuje opis prometne nesreće u formatu: *Prometna nesreća pod šifrom Sifra dogodila se na lokaciji (X, Y)* (znakove *Sifra*, *X* i *Y* potrebno je dohvatiti iz odgovarajućih atributa klase). Unutar glavnog dijela programa potrebno je s tipkovnice učitati podatke **dvije** proizvoljne nesreće (instance klase nesreća) i za svaku u konzolu ispisati njen sadržaj koristeći metodu *ispis*.

2. Napišite metodu **NajvecaApsolutnaRazlika** koja preko argumenta prima matricu **M** cijelih brojeva, te pronalazi i ispisuje najveću apsolutnu razliku između dva elementa matrice – od svih mogućih kombinacija elemenata i njihovih apsolutnih razlika pronalazi i ispisuje onu najveću u sljedećem formatu: *Najveća apsolutna razlika između pojedinih elemenata matrice iznosi Z!, a čine je element x=X i element =Y* (znakove *Z*, *X* i *Y* potrebno je zamijeniti s odgovarajućim vrijednostima).

Apsolutna razlika između dva broja računa se sljedećim izrazom:

$$au = |x - y|$$

Unutar glavnog dijela programa potrebno je stvoriti matricu **M = 3 x 3** i popuniti ju proizvoljnim vrijednostima s tipkovnice. Nakon toga potrebno je iskoristiti metodu *NajvecaApsolutnaRazlika* kako bi se ispisala najveća apsolutna razlika.

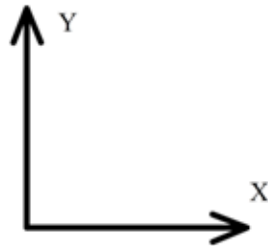


3. Napišite C# kod konzolne aplikacije koja simulira kretanje točke (objekta) na ekranu. Početna pozicija točke je (x_0, y_0) (brojevi s plivajućim zarezom) učitava se preko tipkovnice.

U svakoj iteraciji točka se pomiče za određenu vrijednost prema sljedećim pravilima:

- *W* predstavlja pomicanje gore za 3
- *S* pomicanje dolje za 4.5
- *A* pomicanje u lijevo za 1.5
- *D* pomicanje u desno za 1.5
- ukoliko je iteraciju prije nego što je učitani neko od znakova *W*, *S*, *A* i *D* bio učitani znak *E*, odgovarajuće vrijednosti pomaka su udvostručene samo u trenutnoj iteraciji
- ukoliko je iteraciju prije nego što je učitani neko od znakova *W*, *S*, *A* i *D* bio učitani znak *P*, odgovarajuće vrijednosti pomaka su prepolovljene samo u trenutnoj iteraciji
- ukoliko se učitati neko od preostalih znakova ne događa se ništa

Učitavanje znakova traje dok dva puta za redom nije učitani znak *E*. Na kraju je potrebno u konzolu u proizvoljnom formatu ispisati konačnu poziciju točke i koliko je ukupno pomaka točke bilo. Razmatra se klasično postavljen koordinatni sustav kao na slici:



Kolokvij #1-27 (23. 04. 2024. – grupa A)

1. Zadatak riješite unutar projekta naziva **Zadatak1**.

Napišite C# kôd programa koji preko tipkovnice popunjava matricu niza cijelih brojeva **M = 2 x 2** s proizvoljnim vrijednostima (cijelim brojevima).

Nakon unosa brojeva 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 znamenki sadrži u formatu: Broj *BROJ* sadrži *X* znamenki. (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):

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. Zadatak riješite unutar projekta naziva **Zadatak2**.

Napišite C# kôd programa koji s konzole kontinuirano učitava brzine (brojevi s plivajućim zarezom). Brzine se učitavaju sve dok se ne unese neispravna brzina. Prilikom unosa neispravne brzine, program izlazi iz petlje.

U suprotnom, ukoliko je brzina ispravna, brzina se dodaje na kraj liste naziva „**brzine**“. Brzina je ispravna ukoliko se nalazi u intervalu **[0, 200>**.

Nakon izlaska iz petlje, program ispisuje u proizvoljnom formatu najveću i najmanju brzinu spremljenu u listi „**brzine**“. Za dohvat traženih vrijednosti brzina trebate izraditi metode naziva „**MaxBrzina()**“ i „**MinBrzina()**“. Navedene dvije metode primaju kao ulazni parametar cijelu listu brzina, te nad tom listom izvršavaju potrebne operacije i na kraju vraćaju tražene vrijednosti. „**MaxBrzina()**“ kao rezultat vraća najveću brzinu prisutnu u listi, a „**MinBrzina()**“ kao rezultat vraća najmanju brzinu prisutnu u listi!

Primjer unosa i ispisa (napomena: format unosa/ispisa je proizvoljan):

1. Unesite 0. brzinu: 59
2. Unesite 1. brzinu: 85
3. Unesite 2. brzinu: 36
4. Unesite 3. brzinu: 47
5. Unesite 4. brzinu: 85
6. Unesite 5. brzinu: -1
7. Najveća unesena brzina u listi: 85
8. Najmanja unesena brzina u listi: 36



3. Zadatak riješite unutar projekta naziva **Zadatak3**.

Napišite C# kôd konzolne aplikacije koja učitava cijeli broj **n** s tipkovnice, te potom u listu naziva **taksiji** dodaje **n** instanci klase *Taxi* pri čemu se vrijednosti atributa pojedine instance učitavaju s konzole.

Nakon učitavanja taksija u listu **taksiji**, potrebno je pronaći taksi koji je najbliži poziciji (10, 10) i ispisati njegovog vozača u konzolu u proizvoljnom formatu. Za izračun udaljenosti koristiti euklidsku udaljenost danu sljedećim izrazom:

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

Primjer unosa i ispisa (napomena: format unosa/ispisa je proizvoljan):

1. Unesite broj taksija n: 2
2. Unesite registraciju taksija na indeksu 0: 1337
3. Unesite vozača taksija na indeksu 0: Mufasa
4. Unesite x koordinatu taksija na indeksu 0: 1
5. Unesite y koordinatu taksija na indeksu 0: 2
6. Unesite registraciju taksija na indeksu 1: 185
7. Unesite vozača taksija na indeksu 1: Pumba
8. Unesite x koordinatu taksija na indeksu 1: 6
9. Unesite y koordinatu taksija na indeksu 1: 9
10. Vozač najbližeg taksija poziciji (10, 10) je: Mufasa

```
1 class Taxi
2 {
3     private string REGISTRACIJA;
4     public string VOZAC;
5     private double POZICIJA_X;
6     private double POZICIJA_Y;
7     public Taxi(string registracija, string vozac)
8     {
9         REGISTRACIJA = registracija;
10        VOZAC = vozac;
11    }
12    public void postaviX(double x)
13    {
14        POZICIJA_X = x;
15    }
16    public void postaviY(double y)
17    {
18        POZICIJA_Y = y;
19    }
20    public double dohvatiX()
21    {
22        return POZICIJA_X;
23    }
24    public double dohvatiY()
25    {
26        return POZICIJA_Y;
27    }
28 }
```



4. Što će se ispisati nakon izvršavanja sljedećeg programskog odsječka?

```
1 int n = 8 % 5;
2 int m = 2 * n;
3 while (m > 1)
4 {
5     if (m % 2 == 0)
6     {
7         m++;
8         n = m;
9         m = m / 2;
10    }
11    else
12    {
13        n = m - 2;
14        m = m - 2;
15    }
16    Console.WriteLine(n + "," + m);
17 }
```

5. Što će se ispisati nakon izvršavanja sljedećeg programskog odsječka?

```
1 static void Main(string[] args)
2 {
3     int a = 7 % 4;
4     int b = a + 3;
5     M1(b, a);
6     Console.WriteLine(a + "," + b);
7 }
8 static void M1(int a, int b)
9 {
10    a = b;
11    Console.WriteLine(a + "," + b);
12    M3(a, b);
13 }
14 static void M2(int a, int b)
15 {
16    a = 3;
17    b = a;
18    Console.WriteLine(a + "," + b);
19 }
20 static void M3(int b, int a)
21 {
22    b = Convert.ToInt32(Math.Pow(a, 3));
23    Console.WriteLine(b + "," + a);
24 }
```



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

```
1 int[,] X = new int[4, 4] { { 1, 4, 5, 6 },
2   { 3, 8, 4, 6 },
3   { -5, 11, 8, 18 },
4   { 7, -8, 10, 15 } };
5 for (int j = 0; j < X.GetLength(0) - 1; j++)
6 {
7     int v = 0;
8     for (int i = 1; i < 4; i++)
9     {
10        if (X[i, j] % 3 == 0 ^ X[i, j] > 4)
11        {
12            v++;
13        }
14    }
15    Console.WriteLine(v);
16 }
```



Kolokvij #1-28 (23. 04. 2024. – grupa B)

1. Zadatak riješite unutar projekta naziva **Zadatak1**.

Napišite C# kôd programa koji preko tipkovnice učitava broj redaka **n** i broj stupaca **m** matrice **M**. Potom program treba stvoriti matricu **M** brojeva s plivajućim zarezom i popuniti ju s proizvoljnim vrijednostima s konzole.

Program treba u listu brojeva **ZbrojPoRetcima** spremiti rezultat operacija zbrajanja svih elemenata matrice **M** tako da svaki element liste predstavlja ukupan zbroj vrijednosti po promatranom retku matrice. Prilikom operacije zbrajanja u obzir se uzimaju samo vrijednosti matrice koje se nalaze u intervalu **[0, 200]**.

Na kraju je potrebno ispisati sadržaj liste **ZbrojPoRetcima** u proizvoljnom formatu.

Primjer unosa i ispisa (napomena: format unosa/ispisa je proizvoljan):

Primjer zbrajanja:

[2 5 6] -> 2 + 5 + 6 = 13

[3 6 7] -> 3 + 6 + 7 = 16

Primjer izvođenja programa:

1. Unesite broj redaka n: 2
2. Unesite broj stupaca m: 3
3. Unesite vrijednost 0,0: 2
4. Unesite vrijednost 0,1: 5
5. Unesite vrijednost 0,2: 6
6. Unesite Vrijednost 1,0: 3
7. Unesite vrijednost 1,1: 6
8. Unesite vrijednost 1,2: 7
9. Zbroj 0. retka iznosi 13.
10. Zbroj 1. retka iznosi 16.



2. Zadatak riješite unutar projekta naziva **Zadatak2**.

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

3. Zadatak riješite unutar projekta naziva **Zadatak3**.

Napišite C# kôd metode **ProvjeraBroja** koja vraća **istinu ili laž** ovisno o tome da li je broj (cijeli broj) predan kao argument ispravan ili ne. Broj je ispravan ako je broj znakova u broju strogo veći od **3** (pomoć: cijeli broj se može pretvoriti u tip podataka za niz znakova).

U glavnom dijelu programa potrebno je kontinuirano učitavati proizvoljne brojeve s tipkovnice sve dok se ne učitava neispravan broj. Nakon unosa neispravnog broja, program izlazi iz petlje. Za provjeru ispravnosti koristiti metodu **ProvjeraBroja**.

Nakon učitavanja pojedinog broja potrebno je ispisati u konzolu da li je ispravan ili ne u proizvoljnom formatu.

Primjer unosa i ispisa (napomena: format unosa/ispisa je proizvoljan):

Primjer ispravnog broja:

1. Unesite 1. broj: 1337
2. Učitani broj 1337 je ispravan!

Primjer neispravnog broja:

3. Unesite 2. broj: 133
4. Učitani broj 133 je neispravan!



4. Što će se ispisati nakon izvršavanja sljedećeg programskog odsječka?

```
1 string s1 = " Marko ";
2 string s2 = "Mirko";
3 string s = s1 + s2;
4 int x1 = 0;
5 int x2 = 0;
6 int x3 = 0;
7 int x4 = 0;
8 for (int i = 0; i < s.Length; i++)
9 {
10     if (s[i] == ' ')
11     {
12         x1++;
13     }
14     else if (s[i] == 'o')
15     {
16         x2++;
17     }
18
19     if (s[i] == 'k')
20     {
21         x3++;
22     }
23     else
24     {
25         x4++;
26     }
27 }
28 Console.WriteLine(x1);
29 Console.WriteLine(x2);
30 Console.WriteLine(x3);
31 Console.WriteLine(x4);
```



5. Što će se ispisati nakon izvršavanja sljedećeg programskog odsječka?

```
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 }
```



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

```
1 List<int> lista = new List<int>();
2 for (int i = 32; i > 0; i = i / 2)
3 {
4     lista.Add(i);
5 }
6 lista.RemoveAt(lista.Count - 2);
7 lista.Insert(0, lista.Count);
8 lista.AddRange(new int[] { -2, 0, 3 });
9 for (int i = 0; i < lista.Count; i += 3)
10 {
11     Console.WriteLine(lista[i]);
12 }
```



Kolokvij #1-29 (23. 04. 2024. – grupa C)

1. Zadatak riješite unutar projekta naziva **Zadatak1**.

Napišite C# kôd programa koji kontinuirano učitava podatke vozila: duljinu (broj s plivajućim zarezom) i brzinu (cijeli broj) izmjerene radarom koji je postavljen na prometnici. Učitavanje podataka o vozilu treba ponavljati dok god je duljina vozila **veća od 0** i dok je brzina **veća od 0** (petlja po vlastitom izboru, s tim da se preporučuje korištenje petlje koja barem jednom izvrši naredbe prije provjere uvjeta).

U svakom koraku učitavanja, potrebno je podatke vozila spremati u dvije liste, i to: duljine vozila u listu **duljine**, a brzine vozila u listu **brzine**.

Nakon završetka učitavanja svih vozila potrebno je s konzole učitati proizvoljan indeks **i** te je potrebno ispisati duljinu i brzinu vozila određenog indeksom **i** samo ako je vrijednost indeksa **i** ispravna (indeks postoji u promatranoj listi).

Primjer unosa i ispisa (napomena: format unosa/ispisa je proizvoljan):

Primjer izvođenja programa:

1. Unesite duljinu 0. vozila: 5
2. Unesite brzinu 0. vozila: 70
3. Unesite duljinu 1. vozila: 7
4. Unesite brzinu 1. vozila: 0
5. Unesite indeks: 0
6. Na indeksu 0 duljina vozila je 5, a brzina vozila je 70!

2. Zadatak riješite unutar projekta naziva **Zadatak2**.

Napišite C# kôd metode **Prebroji** koja kao argument prima polje cijelih brojeva **brojevi**.

Program treba prebrojati koliko ima elemenata u polju **brojevi** koji su dozvoljeni – element je validan ako ima strogo više od **4** znamenke.

Metoda treba **vratiti** koliko takvih elemenata ima (pomoć: za provjeru broja znamenki cijelog broja moguće je cijeli broj pretvoriti u tip podataka za niz znakova).

Unutar glavnog dijela programa potrebno je učitati vrijednosti veličine polja **n** te stvoriti polje **brojevi** veličine **n**. Potom je potrebno popuniti polje proizvoljnim vrijednostima, iskoristiti metodu **Prebroji** i, na kraju, u proizvoljnom formatu ispisati rezultat.

Primjer unosa i ispisa (napomena: format unosa/ispisa je proizvoljan):

Primjer izvođenja programa:

1. Unese veličinu polja brojevi n: 4
2. Unesite broj na 0. indeks u polje brojevi: 10
3. Unesite broj na 1. indeks u polje brojevi: 3242
4. Unesite broj na 2. indeks u polje brojevi: 13
5. Unesite broj na 3. indeks u polje brojevi: 34545
6. Broj dozvoljenih vrijednosti u polju brojevi iznosi: 1!



3. Zadatak riješite unutar projekta naziva **Zadatak3**.

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 argumente**, ali izračunava i **vraća** euklidski 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 metodi **izracunajDuzinu**. Format ispisa mora biti: *Duljina prometnice kategorije x iznosi Y!* (znakove X i Y potrebno je zamijeniti odgovarajućim vrijednostima).

4. Što će se ispisati nakon izvršavanja sljedećeg programskog odsječka?

```
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    Console.WriteLine(z + "," + y);
18 }
19 while (x > 0);
```



5. Što će se ispisati nakon izvršavanja sljedećeg programskog odsječka?

```
1 double[] p = new double[7] { 1.2346, 3.257, 8.3678, 4.23345,  
2                               9.7865, 5.68974, 11.4678 };  
3 int i = 1;  
4 for (i = 0; i < p.Length - 1; i += 2)  
5 {  
6     Console.WriteLine(Math.Round(p[(i + 1) / 2], 2));  
7 }  
8 Console.WriteLine(p[p.Length - 2]);
```

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

```
1 static void Main(string[] args)  
2 {  
3     PravokutanTrokut p = new PravokutanTrokut(4, 8);  
4     p.izracunajPovrsinu();  
5     Console.WriteLine(p.izracunajPovrsinu(1, 3));  
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)  
14     {  
15         A = a;  
16         B = b;  
17         povrsina = 0;  
18         ispis();  
19     }  
20     public double izracunajPovrsinu(int a, int b)  
21     {  
22         double povrsina = (b * a) / 2;  
23         ispis();  
24         return povrsina;  
25     }  
26     public void izracunajPovrsinu()  
27     {  
28         povrsina = (A * B) / 2;  
29         ispis();  
30     }  
31     public void ispis()  
32     {  
33         Console.WriteLine(povrsina);  
34     }  
35 }
```



Kolokvij #1-30 (23. 04. 2024. – grupa D)

1. Zadatak riješite unutar projekta naziva **Zadatak1**.

Napišite C# kôd klase **Lokacija**.

Klasa *Lokacija* sadrži javne attribute **x** i **y** (brojevi s plivajućim zarezom) koji predstavljaju geografsku dužinu, odnosno širinu.

Konstruktor klase *Lokacija* prima argumente za **sve** attribute klase. Klasa *Lokacija* treba imati i javnu metodu **Ispis** koja ispisuje sve attribute objekta u formatu: *Lokacija (X,Y)* (znakove X i Y potrebno je dohvatiti iz odgovarajućih atributa klase).

Unutar glavnog dijela programa potrebno je s tipkovnice učitati podatke **dvije** proizvoljne lokacije (instance klase *Lokacija*) i za svaku u konzolu ispisati njen sadržaj koristeći metodu *Ispis*.

Primjer unosa i ispisa (napomena: format unosa/ispisa je proizvoljan):

Primjer izvođenja programa:

1. Unesite x 1. lokacije: 4
2. Unesite y 1. lokacije: 6
3. (ispis) Lokacija (4,6)
4. Unesite x 2. lokacije: 3
5. Unesite y 2. lokacije: 2
6. (ispis) Lokacija (3,2)

2. Zadatak riješite unutar projekta naziva **Zadatak2**.

Napišite C# kôd programa koji preko tipkovnice proizvoljnim vrijednostima popunjava matricu cijelih brojeva dimenzije **M = 2 x 2**.

Nakon popunjavanja cijele matrice, potrebno je ispisati u konzolu najveću vrijednost spremljenu u svakom retku matrice (svaki redak zasebno). Ispis se vrši u proizvoljnom formatu.

Primjer unosa i ispisa (napomena: format unosa/ispisa je proizvoljan):

Primjer izvođenja programa:

1. Unesite (0,0) element: 2
2. Unesite (0,1) element: 4
3. Unesite (1,0) element: 5
4. Unesite (1,1) element: 11
5. (ispis) Najveća vrijednost 0. retka je 4!
6. (ispis) Najveća vrijednost 1. retka je 11!



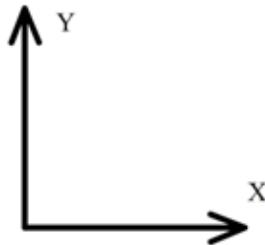
3. Zadatak riješite unutar projekta naziva **Zadatak3**.

Napišite C# kôd konzolne aplikacije koja simulira kretanje točke (objekta) na ekranu. Početna pozicija točke je (x_0, y_0) (brojevi s plivajućim zarezom) učitava se preko tipkovnice.

U svakoj iteraciji točka se pomiče za određenu vrijednost prema sljedećim pravilima:

- *W* predstavlja pomicanje gore za 3 (y-os)
- *S* pomicanje dolje za 4.5 (y-os)
- *A* pomicanje u lijevo za 1.5 (x-os)
- *D* pomicanje u desno za 1.5 (x-os)
- ukoliko se učitava neki od preostalih znakova ne događa se ništa.

Učitavanje znakova traje dok nije učitani znakovni niz **kraj**. Na kraju je potrebno u konzolu u proizvoljnom formatu ispisati konačnu poziciju točke. Razmatra se klasično postavljan koordinatni sustav prikazan na sljedećoj slici. Početna pozicija iznosi $x_0 = 0, y_0 = 0$.



Primjer unosa i ispisa (napomena: format unosa/ispisa je proizvoljan):

Primjer izvođenja programa:

1. Unesite znak: W
(ispis) Nova pozicija: (0,3)
2. Unesite znak: W
(ispis) Nova pozicija: (0,6)
3. Unesite znak: D
(ispis) Nova pozicija: (1.5,6)
4. Unesite znak: A
(ispis) Nova pozicija (0,6)
5. Unesite znak: S
(ispis) Nova pozicija: (0,1.5)
6. Unesite znak: A
(ispis) Nova pozicija: (-1.5,1.5)
7. Unesite znak: kraj
8. (ispis) Konačna pozicija točke je (-1.5,1.5)!



4. Što će se ispisati nakon izvršavanja sljedećeg programskog odsječka?

```
1 int n = 8 % 5;
2 int m = n / 2;
3 int x = 7 + 3 * 2;
4 int y = 2 - 9 / 2;
5 if (m == 1 && (n != 3 ^ true))
6 {
7     m = Convert.ToInt32(Math.Pow(n, 3));
8     n = m;
9     m--;
10    if (x / 2 >= 6 && !(y < 0))
11    {
12        n++;
13        x = n;
14        y = Convert.ToInt32(Math.Sqrt(m + 10));
15    }
16    else
17    {
18        y = Convert.ToInt32(Math.Sqrt(n + 9));
19        x = y;
20        y++;
21    }
22 }
23 else
24 {
25     n = n * 2;
26     m = Convert.ToInt32(Math.Pow(n, 2));
27     if (x / 2 < 6 && !(y >= 0))
28     {
29         x = n;
30         n++;
31         y = Convert.ToInt32(Math.Sqrt(m));
32     }
33     else
34     {
35         y = Convert.ToInt32(Math.Sqrt(n + 10));
36         x = y;
37         y++;
38     }
39 }
40 Console.WriteLine(x + "," + y + "," + n + "," + m);
```



5. Što će se ispisati nakon izvršavanja sljedećeg programskog odsječka?

```
1 static void Main(string[] args)
2 {
3     int[] p = new int[6] { 5, 4, 3, 8, -1, 2 };
4     p[p.Length - 1] = 1;
5     M(p);
6     for (int i = 0; i < p.Length; i += 2)
7     {
8         Console.WriteLine(p[i]);
9     }
10 }
11 private static void M(int[] q)
12 {
13     for (int i = 0; i < q.Length; i++)
14     {
15         if (q[i] % 2 == 1)
16         {
17             q[i] = q[i] - 1;
18         }
19     }
20 }
```

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

```
1 List<string> list = new List<string>();
2 list.AddRange(new string[] { "Ana", "Marko", "Miro" });
3 list.InsertRange(list.Count - 2, new string[] { "Tomo", "Orogoro" });
4 list.Remove("Ana");
5 list.RemoveAt(list.Count - 1);
6 for (int i = 0; i < list.Count; i++)
7 {
8     Console.WriteLine(list[i].Length);
9 }
```



Priprema za drugi kolokvij – Zadatci

Kolokvij #2-24 (02. 06. 2020. – 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 `Šifra` (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.

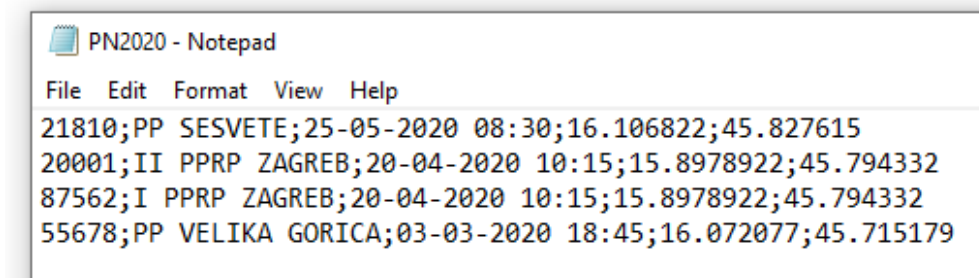
Unutar klase potrebno je napisati i javnu metodu `Ispis` koja ispisuje opis prometne nesreće u formatu: „Prometnu nesreću pod šifrom `Sifra` koja se dogodila `DatumVrijeme` na lokaciji (`X`, `Y`) evidentirala je policijska postaja `Postaja`!“. Vrijednosti `Sifra`, `DatumVrijeme`, `X`, `Y` i `Postaja`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



```
PN2020 - Notepad
File Edit Format View Help
21810;PP SESVETE;25-05-2020 08:30;16.106822;45.827615
20001;II PPRP ZAGREB;20-04-2020 10:15;15.8978922;45.794332
87562;I PPRP ZAGREB;20-04-2020 10:15;15.8978922;45.794332
55678;PP VELIKA GORICA;03-03-2020 18:45;16.072077;45.715179
```

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:

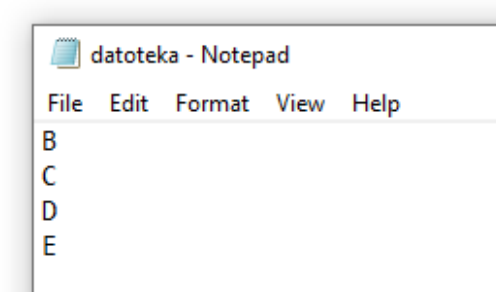
Udaljenost	Tekst
[0, 0.025]	blizu
<0.025, 0.2]	daleko
<0.2, ∞]	jako daleko

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;
    }
}
```



Kolokvij #2-25 (01. 06. 2021. – grupa A)

1. Što će biti zapisano u datoteci data.txt nakon što se izvrši naredba Flush na kraju sljedećeg programskog odsječka.

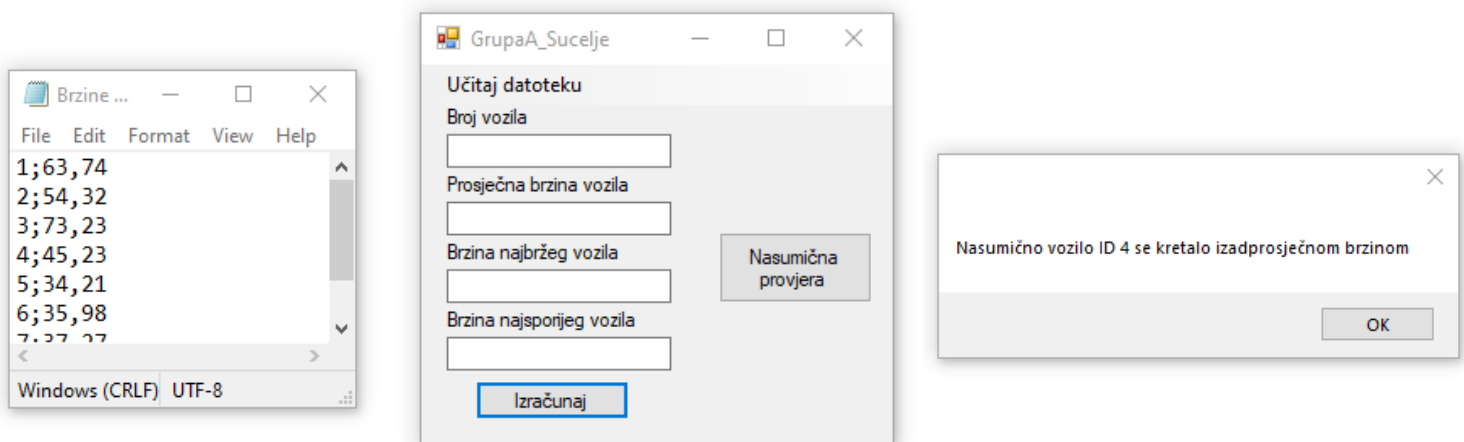
Kao točno rješenje priznaje se samo ono u kojem je vidljiv postupak i sadržaj varijabli u svakoj iteraciji, te u kojima je naznačen svaki trenutak poziva naredbe Flush i sadržaj upisan u datoteku nakon izvršavanja naredbe.

```
static void Main(string[] args)
{
    StreamWriter izlaz = new StreamWriter("data.txt");
    bool[] A = new bool[10];
    for (int i = 2; i <= 9; i++) A[i] = true;
    for (int i = 2; i <= 9; i++)
    {
        if (A[i])
        {
            izlaz.WriteLine(i);
            for (int j = i * 2; j <= 9; j = j + i)
            {
                A[j] = false;
            }
        }
    }
    izlaz.Flush();
    izlaz.Close();
}
```



2. Nadopunite C# kod sučelja GrupaA_Sucelje u nastavku, koje je detaljno prikazno na slici. U rješenju je potrebno naznačiti koji dio koda se odnosi na koju nadopunu (1, 2, 3 ili 4).

- Pritiskom na *Učitaj datoteku* potrebno je učitati datoteku *Brzine.txt*, koja se nalazi u mapi izvršne datoteke programa, u dvodimenzionalno polje, gdje će prvi stupac označavati identifikator vozila ID, a drugi stupac brzinu kretanja. Datoteka sadrži podatke o izmjerenim brzinama vozila u zasebnom retku u sljedećem formatu: ID;BRZINA, pri čemu ID predstavlja identifikator vozila (cijeli broj), BRZINA (broj s plivajućim zarezom) predstavlja izmjerenu brzinu vozila (km/h). Potrebno je koristiti *try-catch* blok te nije potrebno dodati imenski prostor *System.IO*. Unaprijed nije poznato koliko ima redaka u datoteci.
- Pritiskom na gumb *Izračunaj* potrebno je izračunati ukupan broj vozila, prosječnu brzinu svih vozila, najmanju i najveću izmjerenu brzinu. Navedene podatke potrebno je prikazati unutar odgovarajućih TextBoxova, *txtBrojVozila*, *txtProsjecnaBrzina*, *txtNajvecaBrzina* i *txtNajmanjaBrzina*. Nije dozvoljeno ponovno učitavanje podataka iz datoteke već je potrebno koristiti prethodno stvorene liste.
- Pritiskom na gumb *Nasumična provjera* iz polja se odabire nasumično vozilo te se uspoređuje njegova brzina s izračunatim prosjekom te se ispisuje pripadna poruka ukoliko je vozilo brže ili sporije od prosječne brzine. Za nasumičan odabir koristiti klasu *Random*.



```
public partial class Form1 : Form
{
    //Nadopuni - oznaka 1. dio
    public Form1()
    {
        InitializeComponent();
    }
    private void ucitajDatotekuToolStripMenuItem_Click(object sender, EventArgs e)
    {
        //Nadopuni - oznaka 2. dio
    }
    private void btnIzracunaj_Click(object sender, EventArgs e)
    {
        //Nadopuni - oznaka 3. dio
    }
    private void btnNasumicnaProvjera_Click(object sender, EventArgs e)
    {
        //Nadopuni - oznaka 4. dio
    }
}
```



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

$$S(n) = \sum_{i=0}^n (i + 2), \forall n \geq 0$$

Unutar *Main* metode potrebno je pozvati rekurzivnu metodu s argumentom $n = 5$ te ispisati rezultat u formatu: $S(n) = x!$, gdje umjesto x treba ispisati odgovarajuću vrijednost.

4. Kako će izgledati polje 90, 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 iteracija.



Kolokvij #2-26 (01. 06. 2021. – grupa B)

1. Što će biti zapisano u datoteci *data.txt* neposredno prije nego što se pojedini put izvrši naredba *Flush* unutar *if* grananja u sljedećem programskom odsječku.

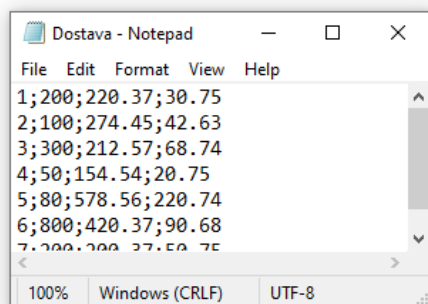
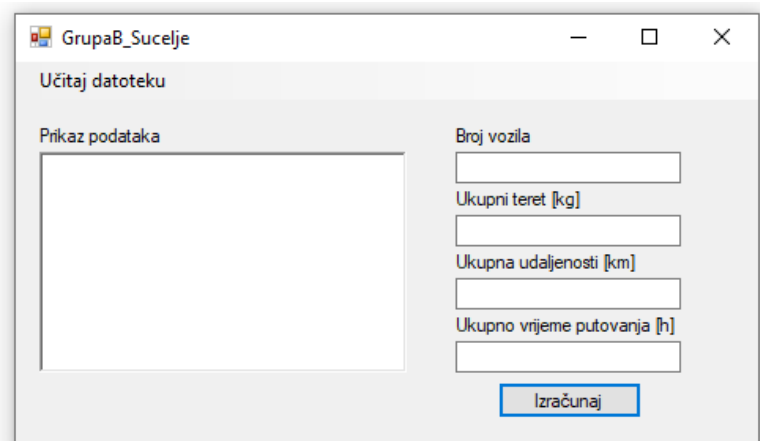
Kao točno rješenje priznaje se samo ono u kojem je vidljiv postupak i sadržaj varijabli u svakoj iteraciji, te u kojima je naznačen svaki trenutak poziva naredbe *Flush* i sadržaj upisan u datoteku neposredno prije izvršavanja naredbe.

```
static void Main(string[] args)
{
    StreamWriter izlaz = new StreamWriter("data.txt");
    for (int i = 6; i < 10; i++)
    {
        for (int j = i / 2; j < i; j += 2)
        {
            if (i - j == 2)
            {
                izlaz.Flush();
            }
            else {
                izlaz.WriteLine(i + j);
            }
        }
    }
    izlaz.Close();
}
```



2. Nadopunite C# kod sučelja *GrupaB_Sucelje* u nastavku, koje je detaljno prikazano na slici. U rješenju je potrebno naznačiti koji dio koda se odnosi na koju nadopunu (1, 2 ili 3).

- Pritiskom na *Učitaj datoteku* potrebno je učitati podatke iz datoteke *Dostava.txt* koja se nalazi na D disku u liste *udaljenosti*, *brzine* i *tereti* deklarirane na razini klase. Datoteka sadrži podatke o dostavnim vozilima u zasebnom retku u sljedećem formatu: ID;TERET;UDALJENOST;BRZINA, pri čemu ID predstavlja identifikator vozila (cijeli broj), TERET (cijeli broj) predstavlja ukupnu prijeđenu udaljenost u kilometrima koje je vozilo dostavilo taj dan, UDALJENOST (realan broj) predstavlja ukupnu prijeđenu udaljenost u kilometrima, dok BRZINA (realan broj) predstavlja prosječnu brzinu vozila (km/h). Učitano datoteku potrebno je liniju po liniju prikazivati unutar kontrole RichTextBox pod nazivom *richTxtPrikaz* (naredba za prelazak u novi red je niz znakova `\n`). Potrebno je koristiti *try-catch* blok te nije potrebno dodati imenski prostor *System.IO*. Unaprijed nije poznato koliko ima redaka u datoteci.
- Pritiskom na gumb *Izračunaj* potrebno je izračunati korišteni broj vozila za dostavu, ukupnu sumu prevezenog tereta (kg), ukupnu sumu prijeđenih udaljenosti svih vozila (km), te ukupno vrijeme trajanja dostave svih vozila (h) (za izračun vremena koristiti jednoliko gibanje). Navedene podatke potrebno je prikazati unutar odgovarajućih TextBox-ova, *txtBrojVozila*, *txtTeret*, *txtUdaljenost* i *txtVrijeme*. Nije dozvoljeno ponovno učitavanje podataka iz datoteke već je potrebno koristiti prethodno stvorene liste.

```
public partial class Form1 : Form
{
    //Nadopuni - oznaka 1. dio
    public Form1()
    {
        InitializeComponent();
    }

    private void ucitajDatotekuToolStripMenuItem_Click(object sender, EventArgs e)
    {
        //Nadopuni - oznaka 2. dio
    }

    private void btnIzracunaj_Click(object sender, EventArgs e)
    {
        //Nadopuni - oznaka 3. dio
    }
}
```



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

$$R(n) = \prod_{i=1}^n i, \forall n \geq 1$$

Unutar *Main* metode potrebno je pozvati rekurzivnu metodu s argumentom $n = 4$ te ispisati rezultat u formatu: „Umnožak brojeva od 1 do 4 iznosi: $x!$ “, gdje umjesto x treba ispisati odgovarajuću vrijednost.

4. Kako će izgledati polje 11, 8, 6, -2, 4, -7 nakon cijele prve iteracije bubble sort algoritma, ukoliko se provodi sortiranje od najmanjeg do najvećeg člana. Potrebno je naznačiti svaku zamjenu elemenata polja zasebno u svim koracima prve iteracije te objasniti postupak.



Kolokvij #2-27 (01. 06. 2021. – grupa C)

1. Što će biti zapisano u datoteci *data.txt* neposredno prije nego što se pojedini put izvrši naredba *Flush* unutar *if* grananja u sljedećem programskom odsječku.

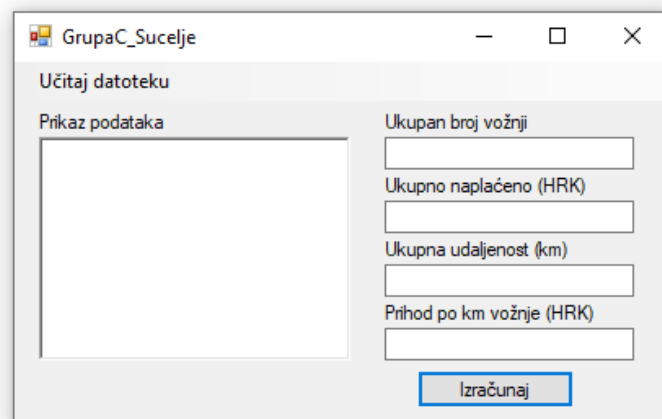
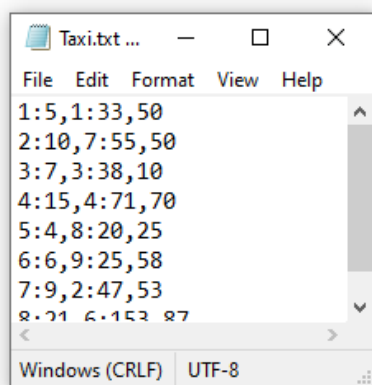
Kao točno rješenje priznaje se samo ono u kojem je vidljiv postupak i sadržaj varijabli u svakoj iteraciji, te u kojima je naznačen svaki trenutak poziva naredbe *Flush* i sadržaj upisan u datoteku neposredno prije izvršavanja naredbe.

```
static void Main(string[] args)
{
    StreamWriter izlaz = new StreamWriter("data.txt");
    for (int i = 6; i < 10; i++)
    {
        for (int j = i / 3; j < i; j += 2)
        {
            if (i - j == 2)
            {
                izlaz.Flush();
            }
            else
            {
                izlaz.WriteLine(i + j);
            }
        }
    }
    izlaz.Close();
}
```



2. Nadopunite C# kod sučelja *GrupaC_Sucelje* u nastavku, koje je detaljno prikazano na slici. U rješenju je potrebno naznačiti koji dio koda se odnosi na koju nadopunu (1, 2 ili 3).

- Pritiskom na *Dohvati podatke* potrebno je dohvatiti podatke iz datoteke *Taxi.txt* koja se nalazi na D disku u liste *voznja*, *udaljenosti* i *naplata* deklarirane na razini klase. Datoteka sadrži podatke o određenim taxi vožnjama u zasebnom retku u sljedećem formatu: ID:UDALJENOST:IZNOS, pri čemu ID predstavlja broj vožnje (cijeli broj), UDALJENOST (realan broj) predstavlja prijeđenu udaljenost u km za pripadnu vožnju, dok IZNOS (realan broj) predstavlja naplaćeni iznos u HRK za odrađenu vožnju. Učitano datoteku potrebno je liniju po liniju prikazivati unutar kontrole RichTextBox pod nazivom *richTxtPrikaz* (naredba za prelazak u novi red je niz znakova `\n`). Potrebno je koristiti try-catch blok te nije potrebno dodati imenski prostor *System.IO*. Unaprijed nije poznato koliko ima redaka u datoteci.
- Pritiskom na gumb *Izračunaj* potrebno je izračunati ukupan broj odrađenih vožnji, ukupan naplaćen iznos (HRK) za sve vožnje, ukupnu sumu prijeđenih udaljenosti svih vožnji (km), te ostvaren prihod u HRK po km vožnje. Navedene podatke potrebno je prikazati unutar odgovarajućih TextBox-ova *txtUkBrVoznji*, *txtNaplata*, *txtUdaljenost* i *txtPrihod*. Nije dozvoljeno ponovno učitavanje podataka iz datoteke već je potrebno koristiti prethodno stvorene liste.



```
public partial class Form1 : Form
{
    //Nadopuni - oznaka 1. dio

    public Form1()
    {
        InitializeComponent();
    }

    private void ucitajDatotekuToolStripMenuItem_Click(object sender, EventArgs e)
    {
        //Nadopuni - oznaka 2. dio
    }

    //3. dio
    private void btnIzracunaj_Click(object sender, EventArgs e)
    {
        //Nadopuni - oznaka 3. dio
    }
}
```



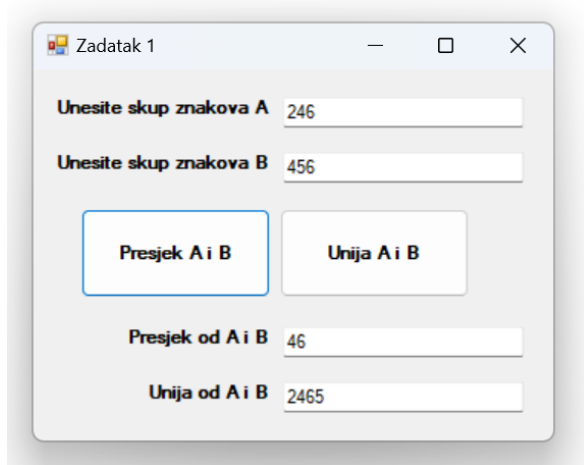
3. Potrebno je napisati kod C# programa koji će za dva zadana cijela broja pronaći njihovu najveću zajedničku mjeru korištenjem Euklidovog rekurzivnog algoritma.
4. Polje 92, 34, 55, 47, 89, 67, 7 potrebno je sortirati od najmanjeg do najvećeg člana *insertion sort* algoritmom. Objasnite postupak sortiranja iteraciju po iteraciju do konačno sortiranog polja.



Kolokvij #2-28 (06. 06. 2023. – grupa A)

1. Zadatak riješite unutar projekta naziva **Zadatak1**.

Napišite C# kôd windows forme prikazane na sljedećoj slici.



Forma sadrži 4 *Label*-a s tekstovima prikazanim na slici, 4 *TextBox*-a naziva *TxtSkupA*, *TxtSkupB*, *TxtPresjek* i *TxtUnija* i dva gumba (*Button*-a) naziva *BtnPresjek* i *BtnUnija* s tekstovima prikazanim na slici.

U *TextBox*-ove *TxtSkupA* i *TxtSkupB* potrebno je unijeti nizove znakova A i B.

Pritiskom na gumb *BtnPresjek* potrebno je presjek skupa A i B ispisati unutar *TextBox*-a *TxtPresjek*.

Pritiskom na gumb *BtnUnija* potrebno je uniju skupa A i B ispisati unutar *TextBox*-a *TxtUnija*.

Provjera ispravnosti: A = 245; B = 456; Presjek A i B = 46; Unija A i B = 2465.

2. Zadatak riješite unutar projekta naziva **Zadatak2**.

Napišite C# kôd rekurzivne metode naziva *matemackilzraz* koja računa sljedeći matematički izraz:

$$r = \sum_{i=0}^n (2)^i$$

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.

Provjera ispravnosti: n = 3; r = 15.



3. Zadatak riješite unutar projekta naziva **Zadatak3.**

Napišite C# kôd konzolne aplikacije koja u polje *brzine* sprema 500 slučajnih brojeva s plivajućim zarezom (brzine) unutar intervala [0, 200>.

Nakon učitavanja potrebno je u datoteku *brzine.txt* na relativnoj putanji spremiti 50 slučajno odabranih vrijednosti iz polja *brzine*, svaku u **zasebnom retku**.

Nakon spremanja vrijednosti u datoteku, ili paralelno s tim, potrebno je sve odabrane brzine spremiti u listu *odabraneBrzine* te ispisati u konzolu prosječnu brzinu, najveću te najmanju spremljenu brzinu. Ispis se izvršava u proizvoljnom formatu.

Potrebno je koristiti *System.IO*, *try-catch* blok te naredbu zatvaranja otvorene datoteke.



4. Zadatak riješite unutar projekta naziva **ZadatakBonusA**.

Napišite C# kôd windows forme prikazane na slici ispod zadatka.

Pritiskom na *ToolStripMenuItem* objekt **Pokreni** aktivira se objekt klase *Timer*, koji izvršava kôd svakih **500 ms**.

Prilikom prvog pokretanja, sve prethodno spremljene lokacije korisnika treba obrisati.

Pritiskom na *ToolStripMenuItem* objekt **Stani** zaustavlja se objekt klase *Timer*.

Potrebno je postaviti vrijednosti širine i dužine objekta klase *PictureBox* na **660** i **280**.

Ako je korisnik pokrenuo objekt klase *Timer*, potrebno je u svakom intervalu **generirati slučajnu lokaciju** novog korisnika, pri čemu x i y vrijednosti ne smiju prelaziti granice objekta klase *PictureBox*.

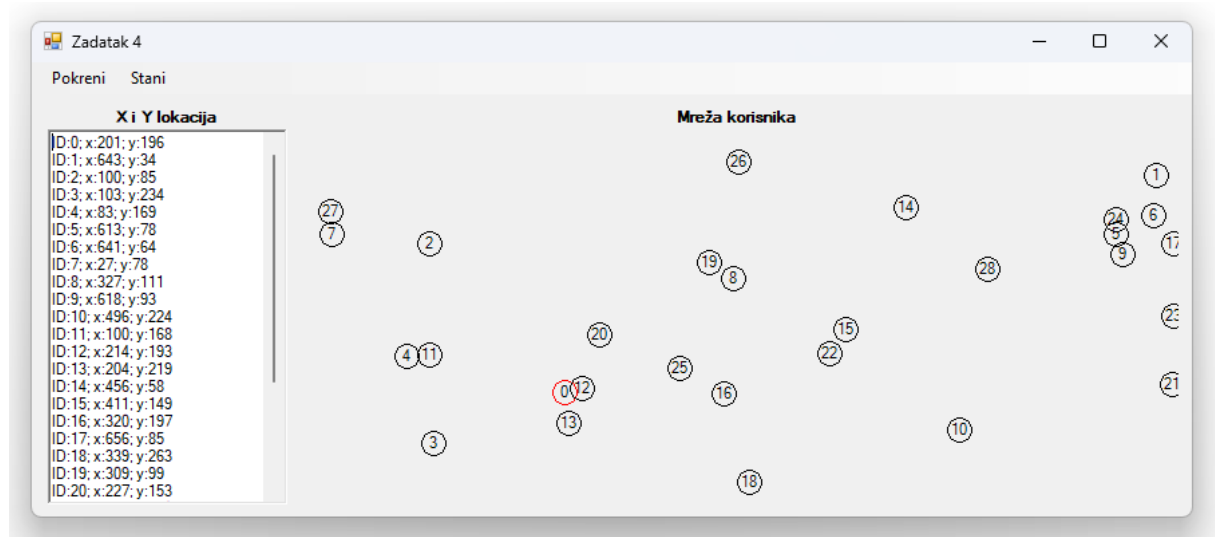
Generirane lokacije treba prikazati unutar objekta klase *RichTextBox* u formatu prikazanom na slici ispod zadatka. Primjer: ID: 0; x: 201; y: 196

Uz tekstualni prikaz, svaku lokaciju treba iscrtati unutar objekta klase *PictureBox*.

Specifikacije grafičkog prikaza lokacija:

- postavite radijus kruga na 18
- prva lokacija (ID = 0) predstavlja skladište i iscrtava se crvenom bojom
- sve ostale lokacije iscrtajte crnom bojom.
- unutar svake lokacije ispišite njezinu vrijednost ID-a.

Za iscrtavanje krugova koristite naredbu **DrawEllipse(...)**. Za ispisivanje vrijednosti ID-a koristite naredbu **DrawString(ID, this.Font, ...)**. Za osvježavanje *PictureBox*-a koristiti metodu **Refresh()**.



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

```
static void Main(string[] args)
{
    ListaBrojeva();
    Console.ReadKey();
}
public static void ListaBrojeva()
{
    List<int> lista = new List<int>();
    int i;
    for (i = 32; i > 0; i /= 2)
    {
        lista.Add(i--);
    }
    lista[0] = lista[i] * lista[lista.Count - 1];
    lista.RemoveAt(3);
    lista.InsertRange(i + 2, new int[] { lista[i + 1] / 2,
        lista[lista.Count - 1 - i] / 2 });
    lista.RemoveRange(lista.Count % 3, 2);
    Console.Write(lista[0]);
    for (i = 1; i < lista.Count; i++)
    {
        Console.Write("; " + lista[i]);
    }
}
```



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

```
static void Main(string[] args)
{
    PerformTimeCalculations();
    Console.ReadKey();
}
public static void PerformTimeCalculations()
{
    DateTime startTime = DateTime.Now;
    SimulateOperation(1000);
    DateTime endTime = DateTime.Now;
    SimulateOperation(500);
    TimeSpan elapsedTime = endTime - startTime;
    double totalSeconds = elapsedTime.TotalSeconds;
    DisplayRoundedTime(totalSeconds);
}
public static void SimulateOperation(int milliseconds)
{
    DateTime startTime = DateTime.Now;
    Thread.Sleep(1000);
    Thread.Sleep(500);
    DateTime endTime = DateTime.Now;
    TimeSpan elapsedTime = endTime - startTime;
    Thread.Sleep(milliseconds);
}
public static void DisplayRoundedTime(double seconds)
{
    int roundedSeconds = (int)Math.Ceiling(seconds);
    Procrastinator(roundedSeconds);
    Console.WriteLine(roundedSeconds);
}
public static void Procrastinator(int seconds)
{
    Thread.Sleep(seconds * 1000);
    DateTime procrastinationEndTime = DateTime.Now;
    double procrastinationDuration =
        (procrastinationEndTime - DateTime.Now).TotalSeconds;
    Console.WriteLine(Math.Floor(procrastinationDuration));
}
```



7. Koji algoritam je implementiran sljedećim programskim odsječkom?

```
static void SomeAlgorithm(int[] polje, int min, int max)
{
    if (min < max)
    {
        int q = Partition(polje, min, max);

        SomeAlgorithm(polje, min, q - 1);
        SomeAlgorithm(polje, q + 1, max);
    }
}

private static int Partition(int[] polje, int min, int max)
{
    int pivot = polje[max];
    int i = min - 1;

    for (int j = min; j < max; j++)
    {
        if (polje[j] <= pivot)
        {
            i++;
            int temp = polje[i];
            polje[i] = polje[j];
            polje[j] = temp;
        }
    }

    int temp1 = polje[i + 1];
    polje[i + 1] = polje[max];
    polje[max] = temp1;
    return i + 1;
}
```

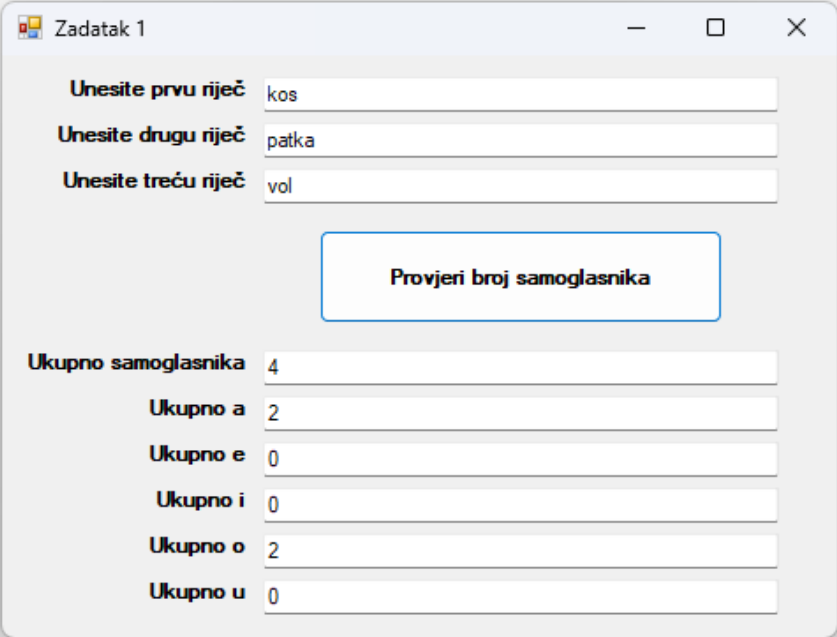


Kolokvij #2-29 (06. 06. 2023. – grupa B)

Napomena: zadaci 4-7 su isti kao i za grupu A istog roka.

1. Zadatak riješite unutar projekta naziva **Zadatak1**.

Napišite C# kôd windows forme prikazane na sljedećoj slici.



Forma sadrži 9 *Label*-a s tekstovima prikazanim na slici, 9 *TextBox*-ova i jedan gumb (*Button*) naziva *btnProvjeri* s tekstom prikazanim na slici.

U pripadne *TextBox*-ove za unos podataka korisnik unosi željene riječi.

Pritiskom na gumb *btnProvjeri* potrebno je izračunati ukupan broj samoglasnika koji se pojavljuju u sva tri unosa te ukupan broj pojavljivanja svakog samoglasnika zasebno. Rezultate provjere potrebno je ispisati u pripadne *TextBox*-ove za ispis rezultata.

2. Zadatak riješite unutar projekta naziva **Zadatak2**.

Napišite C# kôd rekurzivne metode naziva *matematickilzraz* koja računa sljedeći matematički izraz:

$$r = \sum_{i=1}^n ((i)^i + 2)$$

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.

Provjera ispravnosti: *n* = 3; *r* = 38.



3. Zadatak riješite unutar projekta naziva **Zadatak3.**

Napišite C# kôd konzolne aplikacije koja učitava sadržaj datoteke *data.txt* u polje ili listu (strukturu odaberite po želji) **p** (broj elemenata koji je potrebno učitati nije unaprijed poznat).

Datoteka *data.txt* dostupna je na **merlinu** unutar teme 15. Potrebno je spremiti datoteku na način da joj se pristupa preko relativne putanje. Vrijednosti u datoteci su zapisane decimalnim zarezom. Svaki redak datoteke predstavlja jednu zabilježenu brzinu.

U datoteku *data2.txt* potrebno je upisati 3 slučajno odabrana elementa iz polje/liste **p**, zajedno s indeksom elementa u polju/listi. Format zapisa slučajno odabranih podataka: **ID;Brzina**.

Potrebno je koristiti *System.IO*, *try-catch* blok te naredbu zatvaranja/spremanja otvorene datoteke.

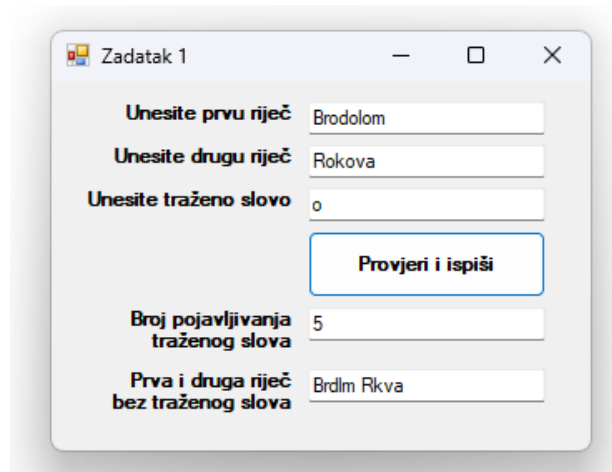


Kolokvij #2-30 (06. 06. 2023. – grupa C)

Napomena: zadaci 4-7 su isti kao i za grupu A istog roka.

1. Zadatak riješite unutar projekta naziva Zadatak1.

Napišite C# kôd windows forme prikazane na sljedećoj slici.



Forma sadrži 5 *Label*-a s tekstovima prikazanim na slici, 5 *TextBox*-ova i jedan gumb (*Button*) naziva *btnProvjeri* s tekstom prikazanim na slici.

U pripadne *TextBox*-ove za unos podataka korisnik unosi željene riječi te traženo slovo.

Pritiskom na gumb *btnProvjeri* potrebno je izračunati ukupan broj pojavljivanja traženog slova u unesenim riječima. Broj pojavljivanja je potrebno ispisati u pripadnu *TextBox*.

Uz ispis broja pojavljivanja traženog slova, potrebno je spojiti izvorne dvije riječi tako da ih povežete razmakom te ih ispišete bez traženog slova.

2. Zadatak riješite unutar projekta naziva Zadatak2.

Napišite C# kôd rekursivne metode naziva *matemackiIzraz* koja računa sljedeći matematički izraz:

$$r = \prod_{i=1}^n \frac{2^i}{2}$$

Unutar *Main* metode potrebno je učitati prirodan broj *n*, te pozvati rekursivnu metodu za izračun navedenog matematičkog izraza s argumentom *n*. Ispis rezultata napraviti unutar *Main* metode u proizvoljnom formatu.

Provjera ispravnosti: $n = 3$; $r = 8$.



3. Zadatak riješite unutar projekta naziva Zadatak3.

Napišite C# kôd konzolne aplikacije koja generira datoteku *data.txt* unutar projekta.

Datoteka sadrži 100 slučajno generiranih brzina vozila te ID-eve vozila. Format retka datoteke: **ID;Brzina**. Brzine su decimalni brojevi u rasponu od 0.0 do 180.0 ([0.0, 180.0>).

Vrijednost ID-a vozila kreću od 0 te svako novo vozilo ima vrijednost ID-a veću za 1 u usporedbi s prethodnim vozilo.

Nakon popunjavanja, potrebno je datoteku spremiti te zatvoriti. Kao rezultat program korisniku u konzoli ispisuje statistiku generiranih brzina. Potrebno je ispisati najveću generiranu brzinu te pripadni ID vozila, te je potrebno ispisati prosječnu generiranu brzinu. Format ispisa je proizvoljan.

Potrebno je koristiti *System.IO*, *try-catch* blok te naredbu zatvaranja/spremanja otvorene datoteke.



Kolokvij #2-31 (11. 06. 2024. – grupa A)

1. Zadatak riješite unutar projekta naziva **Zadatak1**.

Napišite C# kôd konzolne aplikacije koja učitava sadržaj datoteke *brzine.txt* koja bilježi dan zapisa (datum) i brzine u satnom intervalu (broj s plivajućim zarezom). Format zapisa jednog retka je *datum;brzina_1h;brzina_2h;...;brzina_23h;brzina_24h* (vrijednosti su odvojene točka-zarezom).

Datoteku je potrebno preuzeti preko *Merlina* unutar teme 15 i spremiti ju na način da joj se pristupa preko relativne putanje.

Potrebno je pronaći maksimalnu zabilježenu brzinu i dan (datum) na koji je zabilježena. Format ispisa treba biti *Najveća brzina u iznosu od X km/h postignuta je Y!*, (znakove X i Y potrebno je zamijeniti odgovarajućim vrijednostima).

Potrebno je koristiti *try-catch* blok nad čitavim kôdom te koristiti *Using System.IO*. Za provjeru ispravnosti program treba ispisati *Najveća brzina u iznosu od 249.98 km/h postignuta je 4/9/2016!*.

```
brzine.txt
Datoteka    Uređivanje    Prikaži
1/1/2015;5.87;136.46;39.51;152.2;26.97;27.31;145.83;61.02;166.18;65.6;60.95;82.64;2.36;125.34;123.7;153.21;189.65;111.23;36.81;206.13;187.29;193.04;212.37;153.92
1/2/2015;135.04;213.31;200.05;117.62;4.02;73.78;154.39;234.8;194.6;244.88;210.65;192.24;230.4;64.44;66.87;182.3;87.65;129.07;65.92;162.67;231.49;164.73;11.47;177.82
1/3/2015;2.45;217.11;53.31;226.82;2.24;122.81;215.25;62.84;174.1;135.59;17.16;63.66;77.26;28.22;57.92;4.161.21;76.15;138.04;7.49;164.69;181.46;172.82;125.21
1/4/2015;44.36;104.51;118.49;58.21;127.12;49.69;172.43;220.31;201.84;22.23;115.17;36.91;20.47;177.57;232.57;71.79;29.63;147.81;18.14;94.8;47.29;3.21;105.04;59.43
1/5/2015;72.07;223.52;1.46;155.34;26.69;3.98;13.13;70.98;44.29;178.11;182.46;147.73;4.32;157.03;185.72;124.4;213.16;46.84;111.82;55.04;163.04;20.09;71.92;233.64
1/6/2015;93.58;66.25;227.86;16.88;163.32;78.02;77.92;41.15;249.47;59.06;236.14;153.59;48.23;17.09;193.62;197.86;9.09;44.19;242.62;92.36;245.11;84.84;67.47;122.6
1/7/2015;212.09;143.74;131.64;0.45;141.4;50.01;146.39;51.98;151.6;17.82;61.76;210.45;26.12;246.26;157.67;216.28;100.19;141.31;148.26;195.26;170.89;32.13;76.18;196.07
1/8/2015;103.22;163.96;45.94;118.86;27.47;229.56;238.54;8.74;248.78;105.26;54.78;19.58;196.37;77.47;149.75;199.46;162.67;184.16;2.30.41;205.34;233.17;171.74;12.83
1/9/2015;136.52;26.34;242.17;103.79;186.58;122.2;201.71;179.97;67.56;185.58;147.23;227.44;4.06;27.53;24.51;172.04;29.29;53.02;211.67;170.86;191.48;138.09;19.9;22.72
1/10/2015;241.57;198.79;32.59;236.72;30.13;45.77;240.73;120.05;242.22;33.11;232.33;1.13;237.39;134.75;72.16;126.58;153.55;124.81;202.01;10.8;180.24;172.31;195.4;135.14
1/11/2015;159.65;79.96;1.12;152.32;21.51;0.87;71.35;248.43;6.44;219.45;112.23;237.79;89.61;214.98;149.84;21.79;194.85;27.18;235.22;220.95;45.2;23.38;184.64;144.54
1/12/2015;230.86;85.09;17.31;66.66;186.08;9.1;92.48;69.26;3.39;147.45;77.07;200.17;44.65;88.41;98.54;241.35;211.76;233.9;244.69;17.94;22.52;84.37;36.97;188.57
1/13/2015;224.98;180.22;65.95;153.07;187.09;224.45;89.94;136.27;145.32;106.58;171.47;186.42;233.56;4.69;62.36;247.34;126.97;42.97;234.4;192.16;137.91;199.66;227.14;106.44
1/14/2015;178.64;243.87;9.19;61.3;28.69;166.7;122.02;146.49;48.12;78.74;91.68;5.86;193.15;189.41;132.17;28.94;57.51;190.8;54.85;188.07;166.94;162.12;7.78;207.07
1/15/2015;171.54;247.35;140.97;229.55;99.97;94.81;50.66;25.32;203.07;209.52;224.43;193.21;215.26;28.64;107.58;228.62;234.56;24.98;138.29;154.82;163.01;56.5;54.19;187.56
1/16/2015;160.8;205.45;134.64;197.11;144.81;39.07;189.49;16.51;236.09;52.13;139.75;31.35;25.72;142.46;46.51;203.31;28.08;66.37;52.78;233.63;214.98;188.96;63.69;28.87
1/17/2015;83.23;76.24;203.51;141.96;23.83;102.96;44.06;115.05;193.16;203.42;68.76;144.52;210.17;103.55;130.51;164.20;207.92;207.12;163.13;138.89;76.23;202.89;92.1
1/18/2015;228.47;184.98;126.74;96.64;3.71;70.56;222.58;221.84;141.77;105.77;113.89;68.57;85.57;47.53;242.68;133.13;8.07;24.71;88.18;72.31;73.71;151.99;243.14;174.53
1/19/2015;185.86;138.78;213.61;7.07;231.84;150.55;139.98;7;123.34;111.41;163.5;88.85;226.21;197.42;111.33;122.6;230.84;96.92;2.75;104.4;239.66;61.93;78.43;222.49
1/20/2015;215.23;73.99;155.06;114.19;238.05;130.58;18.39;76.47;191.83;202.21;160.78;114.93;106.95;215.51;240.241.21;215.8;106.89;195.23;224.16;134.03;153.83;156.92;60.65
1/21/2015;163.37;215.58;91.47;226.79;149.24;138.75;184.87;123.93;60.35;241.11;41.83;157.85;245.49;25.79;18.95;242.76;93.02;87.02;147.02;24.01;36.64;246.4;15.65;15.33
1/22/2015;106.92;11.54;138.6;132.77;116.7;104.27;88.46;61.4;167.07;94.41;200.81;22.46;39.12;41.6;177.24;53.09;17.34;36.85;54.58;115.84;173.68;82.15;245.72;190.01
1/23/2015;87.94;202.47;131.14;47.01;6.81;132.91;24.01;166.97;199.94;76.14;119.87;137.7;0.15;52.1;7.23;206.08;152.65;230.43;240.78;151.08;74.97;246.49;203.64;51.42
1/24/2015;159.79;93.93;6.67;249.79;191.83;149.81;91.65;24.76;15.82;198.59;44.83;164.23;223.13;41.45;54.59;34.16;70.4;33.84;72.52;212.98;171.72;39.53;52.94;215.23
1/25/2015;29.62;98.5;213.19;193.25;91.38;94.65;115.03;91.45;248.83;79.72;137.23;162.88;126.24;24.22;225.15;50.13;234.49;95.11;121.88;51.56;7.27;93.23;92.95;111.21
1/26/2015;175.84;206.62;80.68;41.13;173.96;240.66;129.7;120.25;40.99;41.44;220.17;93.33;186.62;148.4;183.38;171.17;83.56;203.38;165.4;143.41;154.22;141.7;157.92;46.18
1/27/2015;58.7;22.39;228.49;76.61;167.65;1.38;207.96;186.62;115.55;101.98;17.42;134.76;13.97;73.9;167.49;8.17;200.02;16.97;121.89;84.8;54.06;131.82;113.51;86.09
1/28/2015;161.73;200.71;218;53.9;177.6;89.54;206.99;34.13;48.7;34.5;232.43;151.57;12.17;53.09;202.6;39.62;83.48;33.54;227.78;84.64;130.98;48.63;157.2;9.65
```

2. Zadatak riješite unutar projekta naziva **Zadatak2**.

Napišite C# kôd rekursivne metode naziva *F* koja kao argument prima cijeli broj *n* te računa matematički izraz:

$$R(n) = \prod_{i=1}^n i, \forall n \geq 1$$

$$F(0) = 0, F(1) = 1$$

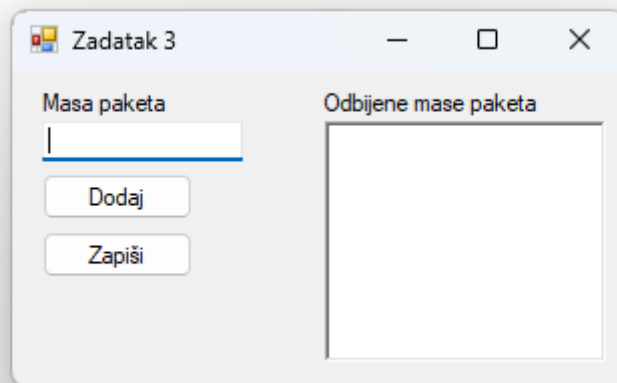
Unutar *Main* metode potrebno je s konzole učitati cijeli broj *n* te koristeći metodu *F* u konzolu ispisati poruku u formatu *Rezultat funkcije F za argument n iznosi z!* (znakove *n* i *z* potrebno je zamijeniti odgovarajućim vrijednostima).

Za provjeru ispravnosti, ukoliko se unese vrijednost 5 potrebno je ispisati *Rezultat funkcije F za argument 5 iznosi 4105!*.



3. Zadatak riješite unutar projekta naziva **Zadatak3.**

Napišite C# kôd windows forme prikazane na sljedećoj slici.



Forma sadrži dva Labela s tekstovima *Masa paketa* i *Odbijene mase paketa*; jedan *TextBox* naziva *txtMasaPaketa*; jedan *RichTextBox* naziva *rtxtOdbijeni*; i dva gumba (*Button-a*) naziva *btnDodaj* i *btnZapisi*) s tekstovima *Dodaj* i *Zapiši*.

U *TextBox txtMasaPaketa* potrebno je unijeti masu paketa (broj s plivajućim zarezom). Pritiskom na gumb *Dodaj* potrebno je provjeriti da li je masa paketa unutar intervala $[0, 20]$, te ako je dodati učitane masu paketa u listu paketa, inače je potrebno dodati učitane masu paketa u zasebno retku u *RichTextBox* naziva *rtxtOdbijeni*.

Pritiskom na gumb s tekstom *Zapiši* potrebno je u datoteku naziva *ispravniPaketi* smještenu na relativnoj putanji zapisati sve mase paketa iz liste paketa, svaku u zasebno retku.



4. Zadatak riješite unutar projekta naziva **ZadatakBonusA**.

Napišite C# kôd windows forme prikazane na slici ispod zadatka.

Pritiskom na *ToolStripMenuItem* objekt **Pokreni** aktivira se objekt klase *Timer*, koji izvršava kôd svakih **500 ms**.

Prilikom prvog pokretanja, sve prethodno spremljene lokacije korisnika treba obrisati.

Pritiskom na *ToolStripMenuItem* objekt **Stani** zaustavlja se objekt klase *Timer*.

Potrebno je postaviti vrijednosti širine i dužine objekta klase *PictureBox* na **660** i **280**.

Ako je korisnik pokrenuo objekt klase *Timer*, potrebno je u svakom intervalu **generirati slučajnu lokaciju** novog korisnika, pri čemu x i y vrijednosti ne smiju prelaziti granice objekta klase *PictureBox*.

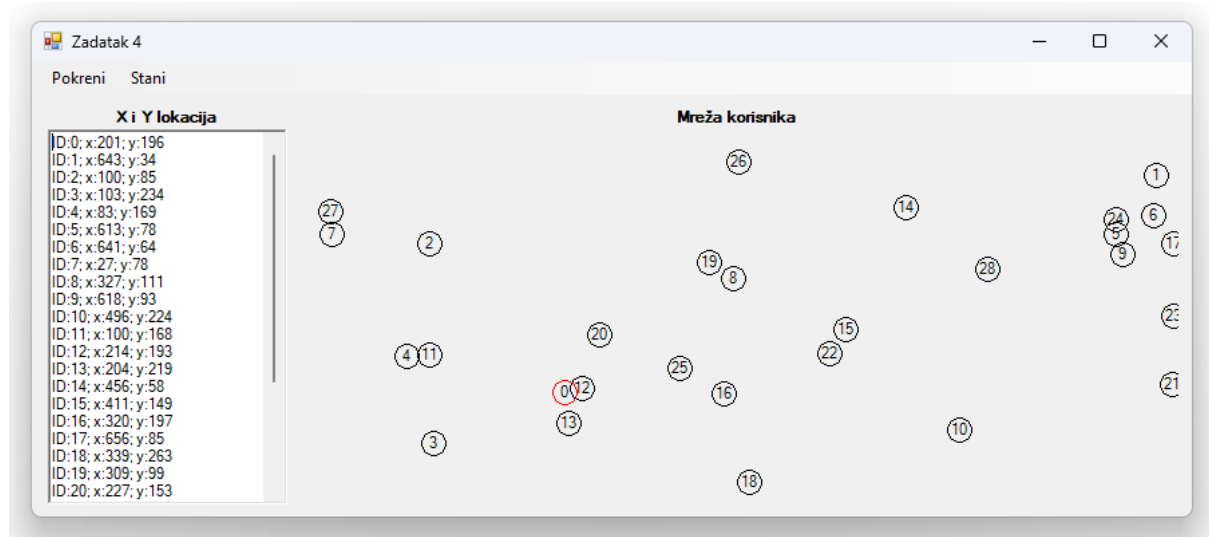
Generirane lokacije treba prikazati unutar objekta klase *RichTextBox* u formatu prikazanom na slici ispod zadatka. Primjer: ID: 0; x: 201; y: 196

Uz tekstualni prikaz, svaku lokaciju treba iscrtati unutar objekta klase *PictureBox*.

Specifikacije grafičkog prikaza lokacija:

- postavite radijus kruga na 18
- prva lokacija (ID = 0) predstavlja skladište i iscrtava se crvenom bojom
- sve ostale lokacije iscrtajte crnom bojom.
- unutar svake lokacije ispišite njezinu vrijednost ID-a.

Za iscrtavanje krugova koristite naredbu **DrawEllipse(...)**. Za ispisivanje vrijednosti ID-a koristite naredbu **DrawString(ID, this.Font, ...)**. Za osvježavanje *PictureBox*-a koristiti metodu **Refresh()**.



5. Koji je algoritam prikazan C# metodom Sortiranje?

```
1 private static void Sortiranje(int[] polje)
2 {
3     int n = polje.Length;
4
5     for (int i = 1; i < n; i++)
6     {
7         int temp = polje[i];
8         int j;
9
10        for (j = i; j >= 1 && polje[j - 1] > temp; j--)
11        {
12            polje[j] = polje[j - 1];
13        }
14        polje[j] = temp;
15    }
16 }
```

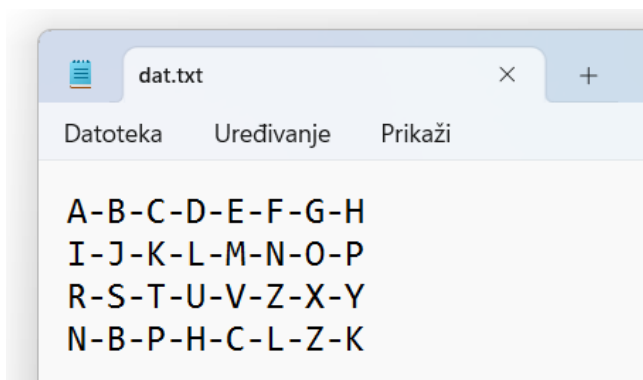
6. Što će se ispisat nakon izvršavanja sljedećeg programskog odsječka?

```
1 static void Main(string[] args)
2 {
3     Ispis(27);
4 }
5 static void Ispis(int n)
6 {
7     if (n <= 1)
8     {
9         Console.WriteLine(n);
10        return;
11    }
12    else
13    {
14        Console.WriteLine(n);
15        Ispis(n / 3);
16        Console.WriteLine(n);
17    }
18 }
```



7. Što će se ispisati nakon izvršavanja sljedećeg programskog odsječka, ako datoteka dat.txt postoji na relativnoj putanji, a čiji je sadržaj prikazan slikom ispod.

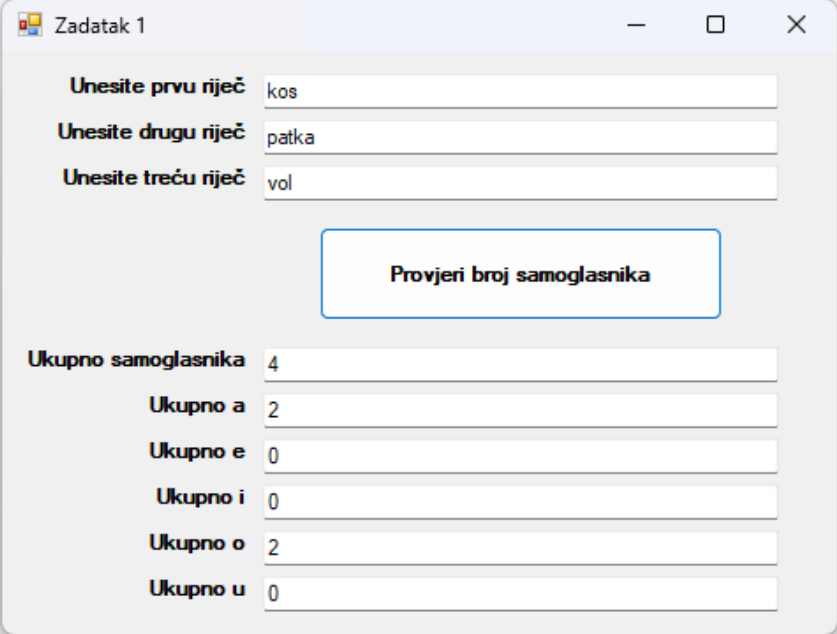
```
1 try
2 {
3     StreamReader citanje = new StreamReader("dat.txt");
4     int i = 5;
5     while (!citanje.EndOfStream)
6     {
7         string linija = citanje.ReadLine();
8         string[] elementi = linija.Split('-');
9         Console.WriteLine(elementi[i]);
10        i -= 2;
11    }
12    citanje.Close();
13 }
14 catch (Exception ex)
15 {
16     Console.WriteLine("A");
17 }
```



Kolokvij #2-32 (11. 06. 2024. – grupa B)

1. Zadatak riješite unutar projekta naziva **Zadatak1**.

Napišite C# kôd windows forme prikazane na sljedećoj slici.



Unesite prvu riječ	Unesite drugu riječ	Unesite treću riječ
kos	patka	vol

Provjeri broj samoglasnika

Ukupno samoglasnika	4
Ukupno a	2
Ukupno e	0
Ukupno i	0
Ukupno o	2
Ukupno u	0

Forma sadrži 9 *Label*-a s tekstovima prikazanim na slici, 9 *TextBox*-ova i jedan gumb (*Button*) naziva *btnProvjeri* s tekstom prikazanim na slici.

U pripadne *TextBox*-ove za unos podataka korisnik unosi željene riječi.

Pritiskom na gumb *btnProvjeri* potrebno je izračunati ukupan broj samoglasnika koji se pojavljuju u sva tri unosa te ukupan broj pojavljivanja svakog samoglasnika zasebno. Rezultate provjere potrebno je ispisati u pripadne *TextBox*-ove za ispis rezultata.



2. Zadatak riješite unutar projekta naziva **Zadatak2.**

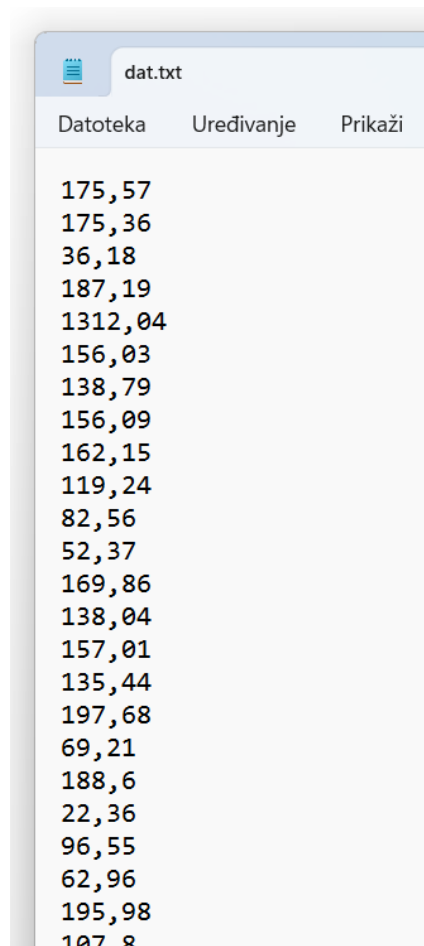
Napišite C# kôd rekurzivne metode naziva *F* koja kao argument prima cijeli broj *n* i računa sljedeći matematički izraz:

$$F(n) = \sum_{i=1}^n ((i)^i + 2)$$

Unutar *Main* metode potrebno je učitati prirodan broj *n*, te koristeći metodu *F* u konzolu ispisati poruku u formatu *Rezultat funkcije F za argument n iznosi z!* (znakove *n* i *z* potrebno je zamijeniti odgovarajućim vrijednostima). Za provjeru ispravnosti, ukoliko se unese vrijednost 5, potrebno je ispisati *Rezultat funkcije F za argument 5 iznosi 3423!*.

3. Zadatak riješite unutar projekta naziva **Zadatak3.**

Napišite C# kôd konzolne aplikacije koja učitava sadržaj datoteke *data.txt* prikazane na sljedećoj slici u strukturu po izboru.



Broj redaka u datoteci nije unaprijed poznat. Datoteka *data.txt* dostupna je na **Merlinu** unutar teme 15. Potrebno je spemiti datoteku na način da joj se pristupa preko relativne putanje. Vrijednosti u datoteci su zapisane decinmalnim zarezom. U datoteku *data2.txt* potrebno je upisati 10 slučajno odabranih elemenata iz odabrane strukture zajedno s pripadajućim indeksom elementa. Potrebno je koristiti *System.IO*, *try-catch* blok te naredbu zatvaranja/spremanja otvorene datoteke.



4. Zadatak riješite unutar projekta naziva **ZadatakBonusB**.

Napišite C# kôd windows forme prikazane na slici ispod zadatka.

Pritiskom na *ToolStripMenuItem* objekt **Pokreni** aktivira se objekt klase *Timer*, koji izvršava kôd svakih **500 ms**.

Prilikom prvog pokretanja, sve prethodno spremljene lokacije korisnika treba obrisati.

Pritiskom na *ToolStripMenuItem* objekt **Stani** zaustavlja se objekt klase *Timer*.

Potrebno je postaviti vrijednosti širine i dužine objekta klase *PictureBox* na **660** i **280**.

Ako je korisnik pokrenuo objekt klase *Timer*, potrebno je u svakom intervalu **generirati slučajnu lokaciju** novog korisnika, pri čemu x i y vrijednosti ne smiju prelaziti granice objekta klase *PictureBox*.

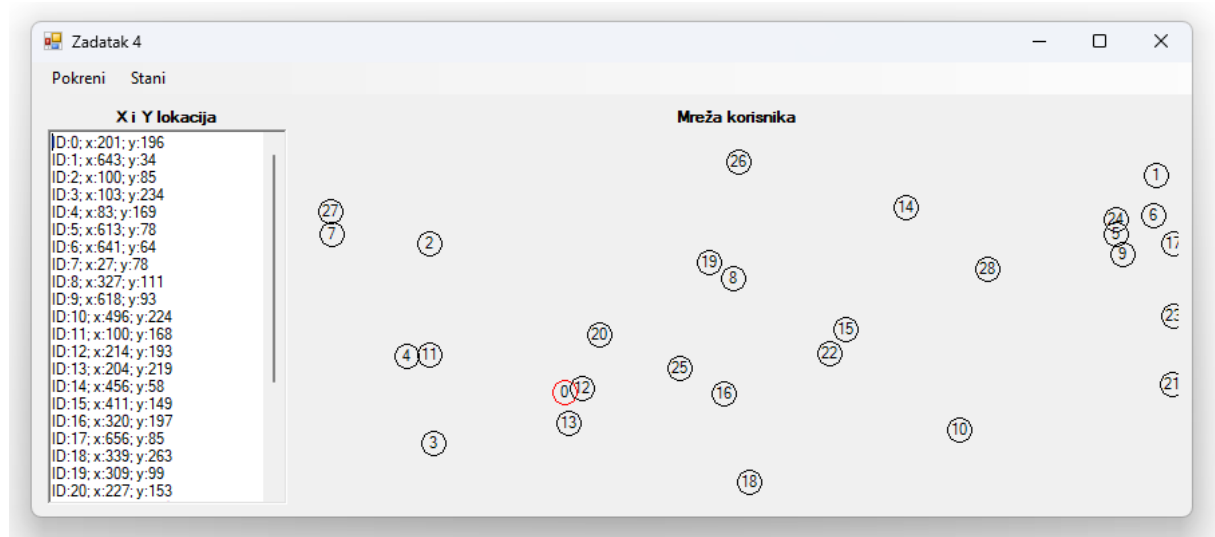
Generirane lokacije treba prikazati unutar objekta klase *RichTextBox* u formatu prikazanom na slici ispod zadatka. Primjer: ID: 0; x: 201; y: 196

Uz tekstualni prikaz, svaku lokaciju treba iscrtati unutar objekta klase *PictureBox*.

Specifikacije grafičkog prikaza lokacija:

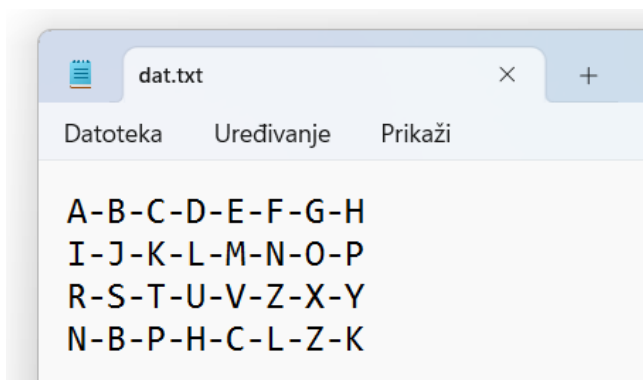
- postavite radijus kruga na 18
- prva lokacija (ID = 0) predstavlja skladište i iscrtava se crvenom bojom
- sve ostale lokacije iscrtajte crnom bojom.
- unutar svake lokacije ispišite njezinu vrijednost ID-a.

Za iscrtavanje krugova koristite naredbu **DrawEllipse(...)**. Za ispisivanje vrijednosti ID-a koristite naredbu **DrawString(ID, this.Font, ...)**. Za osvježavanje *PictureBox*-a koristiti metodu **Refresh()**.



5. Što će se ispisati nakon izvršavanja sljedećeg programskog odsječka, ako datoteka dat.txt postoji na relativnoj putanji, a čiji je sadržaj prikazan slikom ispod.

```
1 try
2 {
3     StreamReader citanje = new StreamReader("dat.txt");
4     int i = 1;
5     while (!citanje.EndOfStream)
6     {
7         string linija = citanje.ReadLine();
8         string[] elementi = linija.Split('-');
9         Console.WriteLine(elementi[i]);
10        i += 2;
11    }
12    citanje.Close();
13 }
14 catch (Exception ex)
15 {
16     Console.WriteLine("A");
17 }
```



6. Koji je algoritam prikazan C# metodom Sortiranje?

```
1 static void Sortiranje(int[] polje)
2 {
3     int n = polje.Length;
4
5     for (int i = 0; i < n; i++)
6     {
7         int minIndex = i;
8         for (int j = i + 1; j < n; j++)
9         {
10            if (polje[j] < polje[minIndex])
11                minIndex = j;
12        }
13        int temp = polje[minIndex];
14        polje[minIndex] = polje[i];
15        polje[i] = temp;
16    }
17 }
```



7. Što će se ispisat nakon izvršavanja sljedećeg programskog odsječka?

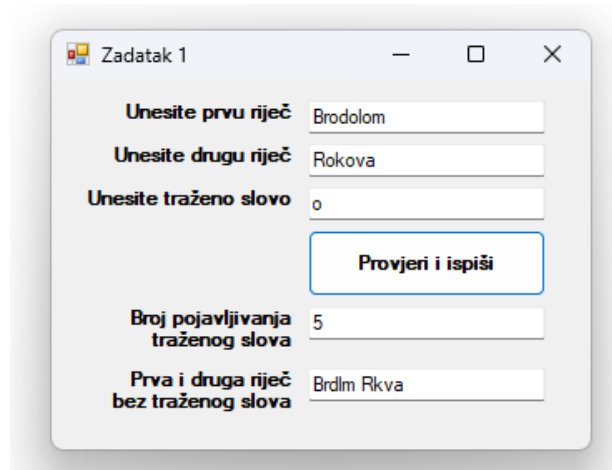
```
1 static void Main(string[] args)
2 {
3     Ispis(81);
4 }
5 static void Ispis(int n)
6 {
7     if (n <= 3)
8     {
9         Console.WriteLine(n);
10        return;
11    }
12    else
13    {
14        Console.WriteLine(n);
15        Ispis(n / 3);
16        Console.WriteLine(n);
17    }
18 }
```



Kolokvij #2-33 (11. 06. 2024. – grupa C)

1. Zadatak riješite unutar projekta naziva **Zadatak1**.

Napišite C# kôd windows forme prikazane na sljedećoj slici.



Forma sadrži 5 *Label*-a s tekstovima prikazanim na slici, 5 *TextBox*-ova i jedan gumb (*Button*) naziva *btnProvjeri* s tekstom prikazanim na slici.

U pripadne *TextBox*-ove za unos podataka korisnik unosi željene riječi te traženo slovo.

Pritiskom na gumb *btnProvjeri* potrebno je izračunati ukupan broj pojavljivanja traženog slova u unesenim riječima. Broj pojavljivanja je potrebno ispisati u pripadnu *TextBox*.

Uz ispis broja pojavljivanja traženog slova, potrebno je spojiti izvorne dvije riječi tako da ih povežete razmakom te ih ispišete bez traženog slova.

2. Zadatak riješite unutar projekta naziva **Zadatak2**.

Napišite C# kôd rekurzivne metode naziva *F* koja kao argument prima cijeli broj *n* i računa sljedeći matematički izraz:

$$F(n) = \prod_{i=1}^n \frac{2^i}{2}$$

Unutar *Main* metode potrebno je učitati prirodan broj *n*, te koristeći metodu *F* u konzolu ispisati poruku u formatu *Rezultat funkcije F za argument n iznosi z!* (znakove *n* i *z* potrebno je zamijeniti odgovarajućim vrijednostima). Za provjeru ispravnosti, ukoliko se unese vrijednost 5, potrebno je ispisati *Rezultat funkcije F za argument 5 iznosi 31!*.



3. Zadatak riješite unutar projekta naziva **Zadatak3**.

Napišite C# kôd konzolne aplikacije koja generira datoteku *data.txt* unutar projekta na relativnoj putanji.

Datoteka treba sadržavati 100 slučajno generiranih brzina vozila te ID-eve vozila. Format retka datoteke treba biti: **ID;Brzina**. Brzine su decimalni brojevi u rasponu od 0.0 do 180.0 ([0.0, 180.0>).

Vrijednost ID-a vozila kreću od 0 te svako novo vozilo ima vrijednost ID-a veću za 1 u usporedbi s prethodnim vozilo.

Nakon popunjavanja, potrebno je datoteku spremiti te zatvoriti. Kao rezultat program korisniku u konzoli ispisuje statistiku generiranih brzina. Potrebno je ispisati najveću generiranu brzinu te pripadni ID vozila u formatu *Najveća zabilježena brzina u iznosu od X zabilježena je na ID-u Y!* (znakove X i Y potrebno je zamijeniti odgovarajućim vrijednostima).

Potrebno je koristiti *System.IO*, *try-catch* blok te naredbu zatvaranja/spremanja otvorene datoteke.



4. Zadatak riješite unutar projekta naziva **ZadatakBonusC**.

Napišite C# kôd windows forme prikazane na slici ispod zadatka.

Pritiskom na *ToolStripMenuItem* objekt **Pokreni** aktivira se objekt klase *Timer*, koji izvršava kôd svakih **500 ms**.

Prilikom prvog pokretanja, sve prethodno spremljene lokacije korisnika treba obrisati.

Pritiskom na *ToolStripMenuItem* objekt **Stani** zaustavlja se objekt klase *Timer*.

Potrebno je postaviti vrijednosti širine i dužine objekta klase *PictureBox* na **660** i **280**.

Ako je korisnik pokrenuo objekt klase *Timer*, potrebno je u svakom intervalu **generirati slučajnu lokaciju** novog korisnika, pri čemu x i y vrijednosti ne smiju prelaziti granice objekta klase *PictureBox*.

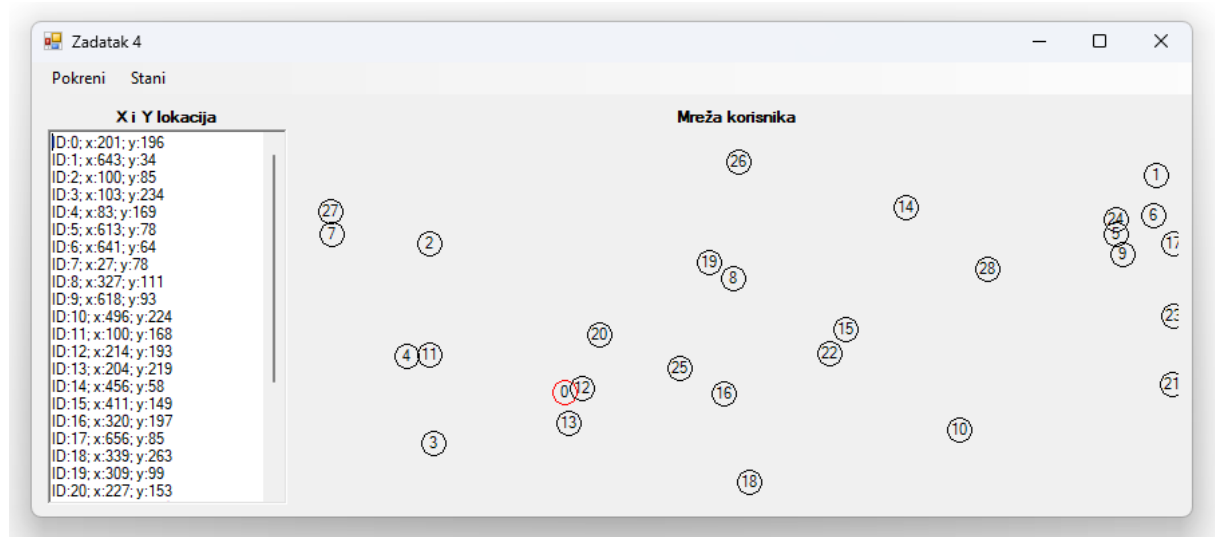
Generirane lokacije treba prikazati unutar objekta klase *RichTextBox* u formatu prikazanom na slici ispod zadatka. Primjer: ID: 0; x: 201; y: 196

Uz tekstualni prikaz, svaku lokaciju treba iscrtati unutar objekta klase *PictureBox*.

Specifikacije grafičkog prikaza lokacija:

- postavite radijus kruga na 18
- prva lokacija (ID = 0) predstavlja skladište i iscrtava se crvenom bojom
- sve ostale lokacije iscrtajte crnom bojom.
- unutar svake lokacije ispišite njezinu vrijednost ID-a.

Za iscrtavanje krugova koristite naredbu **DrawEllipse(...)**. Za ispisivanje vrijednosti ID-a koristite naredbu **DrawString(ID, this.Font, ...)**. Za osvježavanje *PictureBox*-a koristiti metodu **Refresh()**.



5. Koji je algoritam prikazan C# metodom Sortiranje?

```
1 private static void Sortiranje(int[] polje)
2 {
3     bool imaPromjena = true;
4     int temp, velicinaPolja = polje.Length;
5
6     for (int i = 1; (i <= (velicinaPolja - 1)) && imaPromjena; i++)
7     {
8         imaPromjena = false;
9         for (int j = 0; j < (velicinaPolja - 1); j++)
10        {
11            if (polje[j + 1] < polje[j])
12            {
13                temp = polje[j];
14                polje[j] = polje[j + 1];
15                polje[j + 1] = temp;
16                imaPromjena = true;
17            }
18        }
19    }
20 }
```

6. Koliko puta će sljedeći programski odsječak ispisati poruku „Pozdrav!“?

```
1 static void Main(string[] args)
2 {
3     rekMetoda(100);
4 }
5 static void rekMetoda(int x)
6 {
7     Console.WriteLine("Pozdrav!");
8     if (x > 0 && x < 10)
9     {
10        Console.WriteLine("Pozdrav!");
11        return;
12    }
13    else
14    {
15        Console.WriteLine("Pozdrav!");
16        rekMetoda(x / 10);
17        Console.WriteLine("Pozdrav!");
18    }
19 }
```



7. Što će se ispisati nakon izvršavanja sljedećeg programskog odsječka?

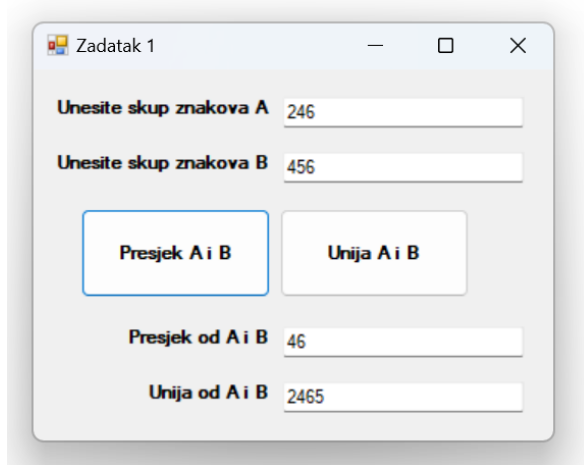
```
1 static void Main(string[] args)
2 {
3     WriteRead();
4 }
5 static void WriteRead()
6 {
7     try
8     {
9         StreamWriter sw = new StreamWriter("dat.txt");
10        sw.WriteLine("A;B;C");
11        sw.WriteLine("E;F;G");
12        sw.WriteLine("H;I;J");
13        sw.WriteLine("B;C;A");
14        sw.WriteLine("G;F;E");
15        sw.WriteLine("J;H;I");
16        sw.Flush();
17        sw.Close();
18
19        StreamReader ulaz = new StreamReader("dat.txt");
20        string linija;
21        while (ulaz.EndOfStream)
22        {
23            linija = ulaz.ReadLine();
24            linija = ulaz.ReadLine();
25            Console.WriteLine(linija);
26        }
27        ulaz.Close();
28        Console.WriteLine("A");
29        linija = ulaz.ReadLine();
30    }
31    catch (Exception ex)
32    {
33        Console.WriteLine("F");
34    }
35    Console.WriteLine("K");
36 }
```



Kolokvij #2-34 (11. 06. 2024. – grupa D)

1. Zadatak riješite unutar projekta naziva **Zadatak1**.

Napišite C# kôd windows forme prikazane na sljedećoj slici.



Forma sadrži 4 *Label*-a s tekstovima prikazanim na slici, 4 *TextBox*-a naziva *TxtSkupA*, *TxtSkupB*, *TxtPresjek* i *TxtUnija* i dva gumba (*Button*-a) naziva *BtnPresjek* i *BtnUnija* s tekstovima prikazanim na slici.

U *TextBox*-ove *TxtSkupA* i *TxtSkupB* potrebno je unijeti nizove znakova A i B.

Pritiskom na gumb *BtnPresjek* potrebno je presjek skupa A i B ispisati unutar *TextBox*-a *TxtPresjek*.

Pritiskom na gumb *BtnUnija* potrebno je uniju skupa A i B ispisati unutar *TextBox*-a *TxtUnija*.

Provjera ispravnosti: A = 245; B = 456; Presjek A i B = 46; Unija A i B = 2465.

2. Zadatak riješite unutar projekta naziva **Zadatak2**.

Napišite C# kôd rekurzivne metode naziva *F* koja kao argument prima cijeli broj *n* i računa sljedeći matematički izraz:

$$F(n) = \sum_{i=1}^n ((i)^i + 2)$$

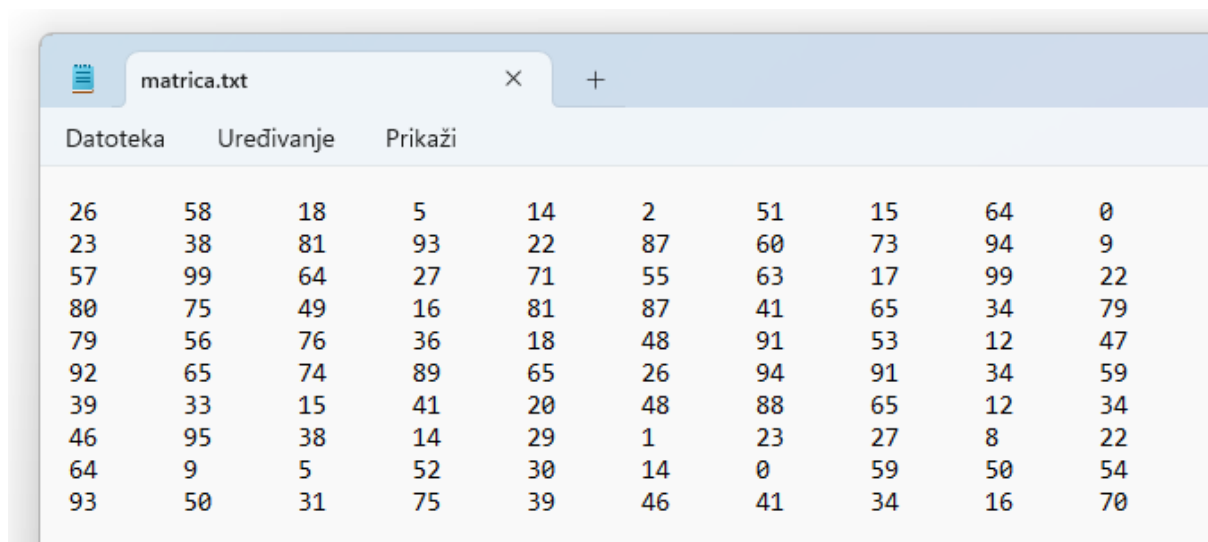
Unutar *Main* metode potrebno je učitati prirodan broj *n*, te koristeći metodu *F* u konzolu ispisati poruku u formatu *Rezultat funkcije F za argument n iznosi z!* (znakove n i z potrebno je zamijeniti odgovarajućim vrijednostima). Za provjeru ispravnosti, ukoliko se unese vrijednost 5, potrebno je ispisati *Rezultat funkcije F za argument 5 iznosi 3423!*.



3. Napišite C# kôd konzolne aplikacije koja u dvodimenzionalno polje (matrica) s 10 redaka i 10 stupaca sprema 100 slučajnih cijelih brojeva unutar intervala [0,99].

Nakon učitavanja potrebno je u datoteku *matrica.txt* na relativnoj putanji spremiti cijelu matricu u formatu kao na sljedećoj slici.

Potrebno je koristiti *try-catch* blok nad čitavim kôdom, te koristiti *Using System.IO*.



The screenshot shows a text editor window titled 'matrica.txt'. The window has a menu bar with 'Datoteka', 'Uređivanje', and 'Prikaži'. The main area displays a 10x10 grid of integers, representing a matrix. The numbers are arranged in 10 rows and 10 columns.

26	58	18	5	14	2	51	15	64	0
23	38	81	93	22	87	60	73	94	9
57	99	64	27	71	55	63	17	99	22
80	75	49	16	81	87	41	65	34	79
79	56	76	36	18	48	91	53	12	47
92	65	74	89	65	26	94	91	34	59
39	33	15	41	20	48	88	65	12	34
46	95	38	14	29	1	23	27	8	22
64	9	5	52	30	14	0	59	50	54
93	50	31	75	39	46	41	34	16	70



4. Zadatak riješite unutar projekta naziva **ZadatakBonusD**.

Napišite C# kôd windows forme prikazane na slici ispod zadatka.

Pritiskom na *ToolStripMenuItem* objekt **Pokreni** aktivira se objekt klase *Timer*, koji izvršava kôd svakih **500 ms**.

Prilikom prvog pokretanja, sve prethodno spremljene lokacije korisnika treba obrisati.

Pritiskom na *ToolStripMenuItem* objekt **Stani** zaustavlja se objekt klase *Timer*.

Potrebno je postaviti vrijednosti širine i dužine objekta klase *PictureBox* na **660** i **280**.

Ako je korisnik pokrenuo objekt klase *Timer*, potrebno je u svakom intervalu **generirati slučajnu lokaciju** novog korisnika, pri čemu x i y vrijednosti ne smiju prelaziti granice objekta klase *PictureBox*.

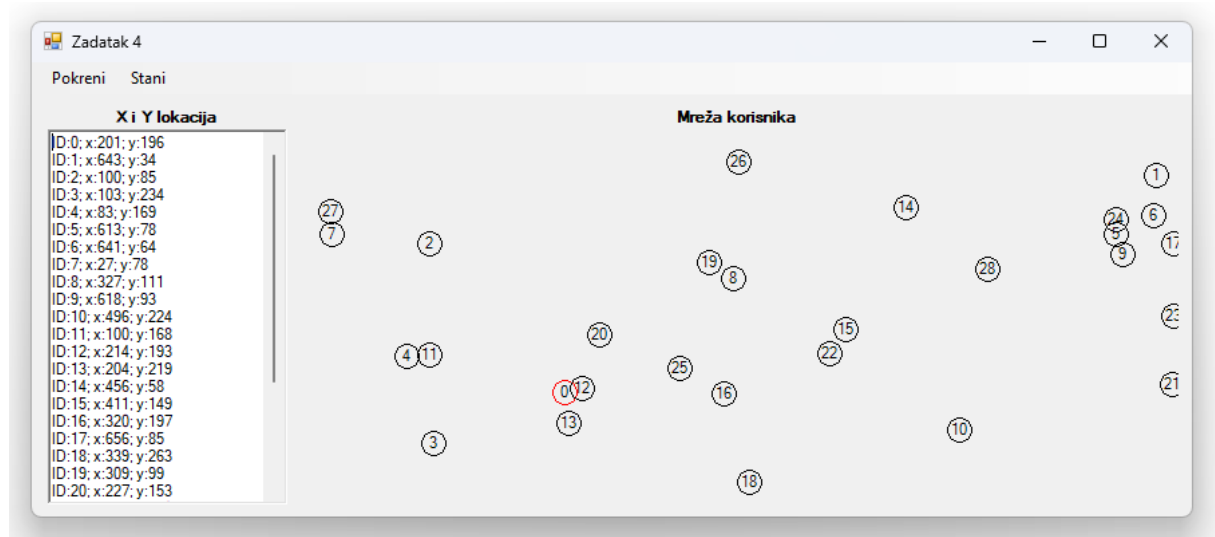
Generirane lokacije treba prikazati unutar objekta klase *RichTextBox* u formatu prikazanom na slici ispod zadatka. Primjer: ID: 0; x: 201; y: 196

Uz tekstualni prikaz, svaku lokaciju treba iscrtati unutar objekta klase *PictureBox*.

Specifikacije grafičkog prikaza lokacija:

- postavite radijus kruga na 18
- prva lokacija (ID = 0) predstavlja skladište i iscrtava se crvenom bojom
- sve ostale lokacije iscrtajte crnom bojom.
- unutar svake lokacije ispišite njezinu vrijednost ID-a.

Za iscrtavanje krugova koristite naredbu **DrawEllipse(...)**. Za ispisivanje vrijednosti ID-a koristite naredbu **DrawString(ID, this.Font, ...)**. Za osvježavanje *PictureBox*-a koristiti metodu **Refresh()**.



5. Što će se ispisati nakon izvršavanja sljedećeg programskog odsječka?

```
1 static void Main(string[] args)
2 {
3     F(4);
4 }
5 static void F(int n)
6 {
7     if (n <= 0)
8     {
9         Console.WriteLine(n - 1);
10        return;
11    }
12    else
13    {
14        Console.WriteLine(n);
15        F(n - 2);
16        F(n - 4);
17    }
18 }
```

6. Koji je algoritam prikazan C# metodom Sortiranje?

```
1 static void Sortiranje(int[] polje, int min, int max)
2 {
3     if (min < max)
4     {
5         int pivot = polje[max];
6         int i = min - 1;
7
8         for (int j = min; j < max; j++)
9         {
10            if (polje[j] <= pivot)
11            {
12                i++;
13                int temp = polje[i];
14                polje[i] = polje[j];
15                polje[j] = temp;
16            }
17        }
18        int temp1 = polje[i + 1];
19        polje[i + 1] = polje[max];
20        polje[max] = temp1;
21        int q = i + 1;
22        Sortiranje(polje, min, q - 1);
23        Sortiranje(polje, q + 1, max);
24    }
25 }
```



7. Što će se ispisati nakon izvršavanja sljedećeg programskog odsječka, ako datoteka brzine.txt postoji na relativnoj putanji, a čiji je sadržaj prikazan slikom ispod. Svi zapisi u datoteci sadrže decimalni zarez.

```
1 static void Main(string[] args)
2 {
3     UcitavanjePodataka();
4 }
5 public static void UcitavanjePodataka()
6 {
7     try
8     {
9         StreamReader sr = new StreamReader("brzine.txt");
10        string linija = sr.ReadLine();
11        int brojac = 0;
12        double najvecaBrzina = double.MinValue;
13        while (!sr.EndOfStream)
14        {
15            linija = sr.ReadLine();
16            string elementi = linija;
17            double brzina = Convert.ToDouble(elementi);
18
19            if (brzina > najvecaBrzina)
20            {
21                najvecaBrzina = brzina;
22                linija = sr.ReadLine();
23                brojac++;
24            }
25            Console.WriteLine(elementi);
26        }
27        sr.Close();
28        Console.WriteLine(brojac);
29    }
30    catch (Exception ex)
31    {
32        Console.WriteLine("F");
33    }
34 }
```



Priprema za prvi kolokvij – Rješenja

Kolokvij #1-1 (19. 04. 2016. – grupa A)

1. Napišite program koji za učitani polumjer kruga r izračunava površinu i opseg kruga.

Ako polumjer nije ispravno zadan (manji od nule) program ispisuje poruku "Niste unijeli dobar polumjer!" te program završava.

Ako je unesen ispravan polumjer kruga potrebno je ispisati poruku: "Površina i opseg kruga s polumjerom r iznosi: $P = ?$, $O = ?!$ ". Za vrijednost π koristiti 3.14. Broj decimalnih mjesta u ispisu nije bitan.

Npr. ako korisnik preko tipkovnice za polumjer učiata vrijednost 2.5 potrebno je ispisati: "Površina i opseg kruga s polumjerom 2.5 iznosi: $P = 19.625$, $O = 15.7!$ ".

```
static void Main(string[] args)
{
    Console.Write("Unesite polumjer kruga r: ");
    double r = Convert.ToDouble(Console.ReadLine());

    if (r >= 0)
    {
        double P = Math.Pow(r, 2) * Math.PI;
        double O = 2 * r * Math.PI;

        Console.WriteLine("Površina i opseg kruga s polumjerom " + r
            + " iznosi: P = " + P + ", O = " + O + "!");
    }
    else
    {
        Console.WriteLine("Niste unijeli dobar polumjer!");
    }

    Console.ReadKey();
}
```



2. Napišite metodu ZbrojKvadrata koja kao argument prima dva prirodna broja te vraća zbroj kvadrata predanih brojeva.

Učitavanje dva prirodna broja, poziv metode ZbrojKvadrata te ispis rezultata potrebno je napraviti unutar Main metode.

Za kvadriranje je potrebno koristiti ugrađene matematičke metode.

Npr. ako korisnik preko tipkovnice unese brojeve 5 i 6 potrebno je ispisati: "Zbroj kvadrata brojeva 5 i 6 iznosi: 61!".

```
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());

    double zbroj = ZbrojKvadrata(x, y);

    Console.WriteLine("Zbroj kvadrata brojeva " + x + " i " + y + " iznosi: "
        + zbroj + "!");

    Console.ReadKey();
}

static double ZbrojKvadrata(int x, int y)
{
    double rezultat = Math.Pow(x, 2) + Math.Pow(y, 2);
    return rezultat;
}
```



Kolokvij #1-2 (19. 04. 2016. – grupa B)

1. Napišite program koji učitava tri realna broja a, b i c.

Ako je broj a veći od broja b program ispisuje zbroj učitanih brojeva, inače ako uvjet ne vrijedi program ispisuje "Nisu učitani ispravni brojevi!", te tu program završava.

Ispis je potrebno formatirati tako da se ispisuju i operandi i operatori. Broj decimalnih mjesta u ispisu nije bitan.

Npr. ako se učitaju redom brojevi $a = 2.5$, $b = 1.5$ i $c = 2$ za zbroj učitanih brojeva (jer je $a > b$) potrebno je ispisati " $2.5 + 1.5 + 2 = 6$ ".

```
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());

    if (a > b)
    {
        double zbroj = a + b + c;
        Console.WriteLine(a + " + " + b + " + " + c + " = " + zbroj);
    }
    else
    {
        Console.WriteLine("Nisu učitani ispravni brojevi!");
    }

    Console.ReadKey();
}
```



2. Napišite metodu KvadratZbroja koja kao argument prima dva prirodna broja te vraća kvadrat zbroja predanih brojeva.

Učitavanje dva prirodna broja, poziv metode KvadratZbroja te ispis rezultata potrebno je napraviti unutar Main metode.

Za kvadriranje je potrebno koristiti ugrađene matematičke metode.

Ispis mora biti u sljedećem formatu: npr. ako korisnik preko tipkovnice unese brojeve 5 i 6 potrebno je ispisati: "Kvadrat zbroja brojeva 5 i 6 iznosi: 121!"

```
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());

    double kvadrat = KvadratZbroja(x, y);

    Console.WriteLine("Kvadrat zbroja brojeva " + x + " i " + y + " iznosi: "
        + kvadrat + "!");

    Console.ReadKey();
}

static double KvadratZbroja(int x, int y)
{
    double rezultat = Math.Pow(x + y, 2);
    return rezultat;
}
```



Kolokvij #1-3 (19. 04. 2016. – grupa C)

1. Napišite program koji učitava tri prirodna broja a, b i c.

Ako je broj a paran broj program ispisuje zbroj brojeva b i c, a ako je a neparan broj ispisuje razliku brojeva b i c.

Ispis je potrebno formatirati tako da se ispisuju i operandi i operatori. Npr. ako se učitaju redom brojevi a = 2, b = 1 i c = 3, budući da je a paran broj potrebno je ispisati: "1 + 3 = 4".

```
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());

    if (a % 2 == 0)
    {
        int zbroj = b + c;
        Console.WriteLine(b + " + " + c + " = " + zbroj);
    }
    else
    {
        int razlika = b - c;
        Console.WriteLine(b + " - " + c + " = " + razlika);
    }

    Console.ReadKey();
}
```



2. Napišite program koji omogućuje korisniku unos lozinke sve dok ona ne bude jednaka vrijednosti lozinke pohranjena u varijabli lozinka, npr. "a1b2". Lozinka može biti bilo koja riječ ili broj ili kombinacija navedenog.

U trenutku kada korisnik upiše ispravnu lozinku treba mu se ispisati poruka: "Lozinka je ispravna!". Za svaku neispravno unesenu lozinku npr. 568 program ispisuje "Unesena lozinka 568 nije ispravna!".

```
static void Main(string[] args)
{
    string lozinka = "a1b2";
    string unos = "";

    do
    {
        Console.Write("Unesite lozinku: ");
        unos = Console.ReadLine();

        if (lozinka == unos)
        {
            Console.WriteLine("Lozinka je ispravna!");
        }
        else
        {
            Console.WriteLine("Unesena lozinka " + unos + " nije ispravna!");
        }
    } while (lozinka != unos);

    Console.ReadKey();
}
```



Kolokvij #1-4 (19. 04. 2016. – grupa D)

1. Napišite program koji učitava dva realna broja a i b koji predstavljaju katete pravokutnog trokuta.

Potrebno je provjeriti da li je umnožak učitanih brojeva veći od nule. Ako nije, program ispisuje "Nisu uneseni ispravni brojevi!" te nakon toga program završava. Ako je, potrebno je ispisati površinu pravokutnog trokuta u formatu: npr. ako su učitani brojevi a = 2.5 i b = 2, potrebno je ispisati "Površina pravokutnog trokuta određenog stanicama a = 2.5 i b = 2 iznosi P = 2.5!".

Broj decimalnih mjesta u ispisu nije bitan.

```
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());

    double umnozak = a * b;

    if (umnozak > 0)
    {
        double površina = a * b / 2;
        Console.WriteLine("Površina pravokutnog trokuta određenog  
stranicama a = " + a + " i b = " + b  
+ " iznosi P = " + površina + "!");
    }
    else
    {
        Console.WriteLine("Nisu uneseni ispravni brojevi!");
    }

    Console.ReadKey();
}
```



2. Napišite program koji učitava prirodne brojeve u jednodimenzionalno polje s 4 elementa.

Nakon što su učitani svi elementi polja potrebno je ispisati sve elemente polja te njihovu prosječnu vrijednost.

Ispis mora biti u sljedećem formatu: npr. ako su redom učitani brojevi 5 6 7 2, potrebno je ispisati:

"Elementi polja su: 5 6 7 2"

Prosječna vrijednost elemenata polja iznosi: 5"

Broj decimalnih mjesta u ispisu nije bitan.

```
static void Main(string[] args)
{
    int[] polje = new int[4];

    for (int i = 0; i < polje.Length; i++)
    {
        Console.Write("Unesite " + (i + 1) + ". element polja: ");
        int x = Convert.ToInt32(Console.ReadLine());

        polje[i] = x;
    }

    int suma = 0;

    Console.Write("Elementi polja su:");
    for (int i = 0; i < polje.Length; i++)
    {
        Console.Write(" " + polje[i]);

        suma = suma + polje[i];
    }

    double prosjek = suma / Convert.ToDouble(polje.Length);

    Console.WriteLine("\nProsječna vrijednost elemenata polja iznosi: "
        + prosjek);

    Console.ReadKey();
}
```



Kolokvij #1-5 (19. 04. 2016. – grupa E)

1. Napišite program koji učitava tri realna broja a, b i c.

Ako je broj c manji od broja a potrebno je ispisati umnožak učitanih brojeva, inače ako uvjet ne vrijedi potrebno je ispisati "Nisu uneseni ispravni brojevi!". Brojevi mogu imati decimalnu točku.

Ispis je potrebno formatirati tako da se ispisuju i operandi i operatori. Broj decimalnih mjesta u ispisu nije bitan.

Npr. ako se učitaju redom brojevi a = 2.5, b = 1.5 i c = 2 za umnožak učitanih brojeva potrebno je ispisati "2.5 * 1.5 * 2 = 7.5".

```
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());

    if (c < a)
    {
        double umnozak = a * b * c;

        Console.WriteLine(a + " * " + b + " * " + " * " + c + " = "
            + umnozak);
    }
    else
    {
        Console.WriteLine("Nisu uneseni ispravni brojevi!");
    }

    Console.ReadKey();
}
```



2. Napišite program koji učitava prirodne brojeve u jednodimenzionalno polje s 4 elementa.

Nakon što su učitani svi elementi polja potrebno je ispisati sve elemente polja te njihov zbroj.

Ispis mora biti u sljedećem formatu: npr. ako su redom učitani brojevi 5 6 7 2, potrebno je ispisati:

"Elementi polja su: 5 6 7 2"

Zbroj elemenata polja iznosi: 20"

```
static void Main(string[] args)
{
    int[] polje = new int[4];

    for (int i = 0; i < polje.Length; i++)
    {
        Console.Write("Unesite " + (i + 1) + ". element polja: ");
        int x = Convert.ToInt32(Console.ReadLine());

        polje[i] = x;
    }

    int zbroj = 0;

    Console.Write("Elementi polja su:");
    for (int i = 0; i < polje.Length; i++)
    {
        Console.Write(" " + polje[i]);

        zbroj = zbroj + polje[i];
    }

    Console.WriteLine("\nZbroj elemenata polja iznosi: " + zbroj);

    Console.ReadKey();
}
```



Kolokvij #1-6 (25. 04. 2017. – grupa A)

1. Napišite metodu ZbrojiOduzmi koja preko argumenata prima dva cijela broja a i b te vraća rezultat operacije zbrajanja i oduzimanja cijelih brojeva unutar intervala [a, b] (uključujuće granice).

Za svaki broj unutar intervala [a, b] provjerava se da li je djeljiv ili s 3 ili s 4. Ako je broj djeljiv s 3 onda se broj pribraja rezultatu, a ako je broj djeljiv s 4 onda se broj oduzima od rezultata. Početna vrijednost rezultata je 0.

Učitavanje dva prirodna broja, poziv metode ZbrojiOduzmi i ispis rezultata potrebno je napraviti unutar Main metode.

Npr. ako korisnik preko tipkovnice unese brojeve a = 2 i b = 7 potrebno je ispisati: "Rezultat operacije ZbrojiOduzmi unutar intervala [2, 7] iznosi: 5!". Budući da su unutar intervala [2, 7] brojevi djeljivi s tri 3 i 6, a broj djeljiv s četiri je samo 4, rezultat je $0 + 3 - 4 + 6 = 5$.

```
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());

    int rezultat = ZbrojiOduzmi(a, b);

    Console.WriteLine("Rezultat operacije ZbrojiOduzmi unutar intervala ["
        + a + ", " + b + "] iznosi: " + rezultat + "!");

    Console.ReadKey();
}

static int ZbrojiOduzmi(int a, int b)
{
    int rezultat = 0;

    for (int i = a; i <= b; i++)
    {
        if (i % 3 == 0)
        {
            rezultat += i;
        }
        if (i % 4 == 0)
        {
            rezultat -= i;
        }
    }

    return rezultat;
}
```



Kolokvij #1-7 (25. 04. 2017. – grupa B)

1. Napišite program koji učitava 5 realnih brojeva u jednodimenzionalno polje, pritom je potrebno koristiti petlju po izboru.

Nakon što su učitani svi elementi polja potrebno je ispisati sve elemente polja te vrijednost najvećeg elementa polja.

Format ispisa mora biti kao u primjeru. Npr. ako su redom učitani brojevi 5.5, 7.8, 6.2, 4.7 i 9 potrebno je ispisati:

"Elementi polja su:

5.5

7.8

6.2

4.7

9

Vrijednost najvećeg elementa polja iznosi: 9!"

```
static void Main(string[] args)
{
    double[] polje = new double[5];

    double najveci = double.MinValue;

    for (int i = 0; i < polje.Length; i++)
    {
        Console.Write("Unesite " + (i + 1) + ". broj: ");
        double x = Convert.ToDouble(Console.ReadLine());

        polje[i] = x;

        if (x > najveci)
        {
            najveci = x;
        }
    }

    Console.WriteLine("Elementi polja su:");

    for (int i = 0; i < polje.Length; i++)
    {
        Console.WriteLine(polje[i]);
    }

    Console.WriteLine("Vrijednost najvećeg elementa polja iznosi: "
        + najveci + "!");

    Console.ReadKey();
}
```



Kolokvij #1-8 (25. 04. 2017. – grupa C)

1. Napišite program koji učitava 9 prirodnih brojeva u dvodimenzionalno polje (matricu 3 x 3), pritom je potrebno koristiti petlje po izboru. Nakon što su učitani svi elementi matrice potrebno je ispisati samo drugi stupac matrice prema formatu u primjeru.

Npr. ako su redom učitani brojevi po redcima

1,2,3

4,5,6

7,8,9

potrebno je ispisati:

"Elementi drugog stupca matrice su:

2

5

8"

```
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 element matrice [" + i + ", " + j + "]: ");
            matrica[i, j] = Convert.ToInt32(Console.ReadLine());
        }
    }

    Console.WriteLine("Elementi drugog stupca matrice su:");

    for (int i = 0; i < matrica.GetLength(0); i++)
    {
        Console.WriteLine(matrica[i, 1]);
    }

    Console.ReadKey();
}
```



Kolokvij #1-9 (25. 04. 2017. – grupa D)

1. Napišite metodu Izracun koja preko argumenta prima tri prirodna broja a, b i c, te ispisuje rezultat matematičkog izraza:

$$c^8 + \sum_{i=a}^b i$$

Granice a i b su uključujuće. Učitavanje tri prirodna broja i poziv metode Izracun potrebno je napraviti unutar Main metode.

Npr. ako korisnik preko tipkovnice unese brojeve a = 2, b = 5 i c = 1 potrebno je ispisati: "Rezultat zadanog matematičkog izraza iznosi: 15!", budući da je:

$$c^8 = 1^8 = 1$$

$$\sum_{i=2}^5 2 + 3 + 4 + 5 = 14$$

```
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());

    Izracun(a, b, c);

    Console.ReadKey();
}

static void Izracun(int a, int b, int c)
{
    double rezultat = 0.0;

    for (int i = a; i <= b; i++)
    {
        rezultat += i;
    }

    rezultat += Math.Pow(c, 8);

    Console.WriteLine("Rezultat zadanog matematičkog izraza iznosi: "
        + rezultat + "!");
}
```



Kolokvij #1-10 (25. 04. 2017. – grupa E)

1. Napišite program koji učitava 3 cijela broja u jednodimenzionalno polje (koristiti petlju po izboru). Nakon što su učitani svi elementi polja potrebno je ispisati sve elemente polja s dvostruko manjom vrijednošću te indeks najmanjeg elementa u polju.

Format ispisa mora biti kao u primjeru. Npr. ako su redom učitani brojevi 5, 7 i -6 potrebno je ispisati:

"Dvostruko manji elementi polja su:

2.5

3.5

-3

Indeks najmanjeg elementa polja iznosi: 2!"

```
static void Main(string[] args)
{
    int[] polje = new int[3];
    int najmanjiElement = int.MaxValue;
    int indeksNajmanjeg = -1;

    for (int i = 0; i < polje.Length; i++)
    {
        Console.Write("Unesite " + (i + 1) + ". element polja: ");
        polje[i] = Convert.ToInt32(Console.ReadLine());

        if (polje[i] < najmanjiElement)
        {
            najmanjiElement = polje[i];
            indeksNajmanjeg = i;
        }
    }

    Console.WriteLine("Dvostruko manji elementi polja su:");

    for (int i = 0; i < polje.Length; i++)
    {
        Console.WriteLine(polje[i] / 2.0);
    }

    Console.WriteLine("Indeks najmanjeg elementa polja iznosi: "
        + indeksNajmanjeg + "!");

    Console.ReadKey();
}
```



Kolokvij #1-11 (24. 04. 2018. – grupa A)

1. Napišite program koji učitava četiri realne vrijednosti s tipkovnice, redom x1, y1, x2 i y2 koje predstavljaju dvije točke u koordinatnom sustavu T1 (x1, y1) i T2 (x2, y2), odnosno dužinu $\overline{T_1T_2}$.

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 $P(x, y) = P((x1 + x2) / 2, (y1 + y2) / 2)$.

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)“.

```
static void Main(string[] args)
{
    Console.Write("Unesite koordinatu x1 točke T1: ");
    double x1 = Convert.ToDouble(Console.ReadLine());

    Console.Write("Unesite koordinatu y1 točke T1: ");
    double y1 = Convert.ToDouble(Console.ReadLine());

    Console.Write("Unesite koordinatu x2 točke T2: ");
    double x2 = Convert.ToDouble(Console.ReadLine());

    Console.Write("Unesite koordinatu y2 točke T2: ");
    double y2 = Convert.ToDouble(Console.ReadLine());

    if (Provjera(x1) && Provjera(y1) && Provjera(x2) && Provjera(y2))
    {
        double xp = (x1 + x2) / 2;
        double yp = (y1 + y2) / 2;

        Console.WriteLine("Polovište učitane dužine je: P(" + xp + ", "
            + yp + ")");
    }
    else
    {
        Console.WriteLine("Nisu unesene ispravne vrijednosti!");
    }

    Console.ReadKey();
}

static bool Provjera(double broj)
{
    if (broj >= -10 && broj <= 10)
    {
        return true;
    }

    return false;
}
```





Kolokvij #1-12 (24. 04. 2018. – grupa B)

1. Napišite program koji učitava imena 5 studenata s tipkovnice u jednodimenzionalno polje koristeći petlju po vlastitom izboru.

Nakon što su učitana sva imena studenata potrebno je ispisati sva unesena imena, te ime koje je najduže (ima najviše slova) i najkraće (ima najmanje slova).

Format ispisa treba biti kao u primjeru. Primjerice ako se redom učitaju imena: Marko, Ana, Luka, Tomislav i Ante potrebno je ispisati:

„Unesena imena studenata su:

Marko

Ana

Luka

Tomislav

Ante

Najduže ime je: Tomislav

Najkraće ime je: Ana“.

```
static void Main(string[] args)
{
    string[] studenti = new string[5];

    for (int i = 0; i < studenti.Length; i++)
    {
        Console.Write("Unesite ime " + (i + 1) + ". studenta: ");
        studenti[i] = Console.ReadLine();
    }

    string najduzeIme = studenti[0];
    string najkraceIme = studenti[0];

    Console.WriteLine("Unesena imena studenata su:");
    for (int i = 0; i < studenti.Length; i++)
    {
        Console.WriteLine(studenti[i]);

        if (studenti[i].Length > najduzeIme.Length)
        {
            najduzeIme = studenti[i];
        }
        else if (studenti[i].Length < najkraceIme.Length)
        {
            najkraceIme = studenti[i];
        }
    }

    Console.WriteLine("\nNajduže ime je: " + najduzeIme);
    Console.WriteLine("Najkraće ime je: " + najkraceIme);

    Console.ReadKey();
}
```



Kolokvij #1-13 (24. 04. 2018. – grupa C)

1. Napišite program koji učitava 5 cjelobrojnih vrijednosti bodova s kolokvija u jednodimenzionalno polje.

Potrebno je ispisati koliko je studenata pristupilo kolokviju i koja je prosječna ocjena u formatu kao u primjeru.

Izračun prosječne vrijednosti i ispis rezultata potrebno je obaviti pozivom metode Izracun koja kao argument prima učitano polje te NE vraća rezultat, već ga ispisuje u konzolu.

Primjerice ako se redom učitaju vrijednosti: 10, 15, 20, 0 i 5 potrebno je ispisati: „Kolokviju je pristupilo 5 studenata, a prosječan broj bodova je 10!“.

```
static void Main(string[] args)
{
    int[] bodovi = new int[5];

    for (int i = 0; i < bodovi.Length; i++)
    {
        Console.Write("Unesite bodove s kolokvija za " + (i + 1)
            + ". studenta: ");
        bodovi[i] = Convert.ToInt32(Console.ReadLine());
    }

    Izracun(bodovi);

    Console.ReadKey();
}

static void Izracun(int[] bodovi)
{
    int zbroj = 0;

    for (int i = 0; i < bodovi.Length; i++)
    {
        zbroj += bodovi[i];
    }

    double prosjek = zbroj / Convert.ToDouble(bodovi.Length);

    Console.WriteLine("Kolokviju je pristupilo " + bodovi.Length
        + " studenata, a prosječan broj bodova je " + prosjek + "!");
}
```



Kolokvij #1-14 (24. 04. 2018. – grupa D)

1. 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, od zadnjeg prema prvom elementu.

Format ispisa mora biti kao u primjeru. Primjerice ako su redom učitane vrijednosti po redcima:

9,8,7

6,5,4

3,2,1

potrebno je ispisati:

„Elementi dijagonale matrice obrnutim redoslijedom 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 element matrice [" + i + "," + j + "]: ");
            matrica[i, j] = Convert.ToInt32(Console.ReadLine());
        }
    }

    Console.WriteLine("Elementi dijagonale matrice obrnutim redoslijedom su:");

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

    Console.ReadKey();
}
```



Kolokvij #1-15 (24. 04. 2018. – grupa E)

1. Napišite program koji učitava 6 prezimena studenata u jednodimenzionalno polje koristeći petlju po vlastitom izboru.

Nakon što su učitana sva prezimena potrebno je ispisati koliko prezimena ima broj slova manji od 5, koliko ih ima jednak 5, te koliko ih ima većih od 5.

Format ispisa mora biti kao u primjeru. Primjerice, ako se redom učitaju vrijednosti: Čičak, Ivić, Palčić, Braovac, Lovrić, Guja potrebno je ispisati:

„Broj prezimena s manje od 5 slova: 2

Broj prezimena s točno 5 slova: 1

Broj prezimena s više od 5 slova: 3“.

```
static void Main(string[] args)
{
    string[] studenti = new string[6];

    int brojSlovaManjiOd5 = 0;
    int brojSlovaTocno5 = 0;
    int brojSlovaViseOd5 = 0;

    for (int i = 0; i < studenti.Length; i++)
    {
        Console.Write("Unesite prezime " + (i + 1) + ". studenta: ");
        studenti[i] = Console.ReadLine();

        if (studenti[i].Length < 5)
        {
            brojSlovaManjiOd5++;
        }
        else if (studenti[i].Length == 5)
        {
            brojSlovaTocno5++;
        }
        else
        {
            brojSlovaViseOd5++;
        }
    }

    Console.WriteLine("Broj prezimena s manje od 5 slova: "
        + brojSlovaManjiOd5);
    Console.WriteLine("Broj prezimena s točno od 5 slova: "
        + brojSlovaTocno5);
    Console.WriteLine("Broj prezimena s više od 5 slova: "
        + brojSlovaViseOd5);

    Console.ReadKey();
}
```



Kolokvij #1-19 (12. 04. 2022. – grupa A)

1. Napišite C# kod konzolne aplikacije koja preko tipkovnice učitava brzine na prometnici (cijeli broj) sve dok je vrijednost brzine unutar intervala [0, 200].

U svakoj iteraciji potrebno je učitano brzinu dodati u listu *brzine* (dodaju se samo valjane brzine).

Nakon učitavanja potrebno je u konzolu ispisati brzine u listi *brzine* čije su vrijednosti veće od 100 zajedno s njihovim indeksima u formatu *Brzina vrijednosti x veća je od 100, a nalazi se na indeksu y u listi brzine!* (znakove x i y potrebno je zamijeniti odgovarajućim vrijednostima).

Dodatno, na kraju je potrebno ispisati i koliko brzina ima u listi *brzine* čija je vrijednost veća od 100 u formatu *Ukupno ima z brzina u listi brzine čija je vrijednost veća od 100!* (znak z potrebno je zamijeniti odgovarajućom vrijednošću).

Rješenje:

```
static void Main(string[] args)
{
    List<int> brzine = new List<int>();
    int v = 0;

    do
    {
        Console.Write("Unesite " + (brzine.Count + 1) + ". brzinu: ");
        v = Convert.ToInt32(Console.ReadLine());

        if (v >= 0 && v <= 200)
        {
            brzine.Add(v);
        }
    } while (v >= 0 && v <= 200);

    int brojVecihOd100 = 0;

    for (int i = 0; i < brzine.Count; i++)
    {
        if (brzine[i] > 100)
        {
            Console.WriteLine("Brzina vrijednosti " + brzine[i]
                + " veća je od 100, a nalazi se na indeksu " + i
                + " u listi brzine!");
            brojVecihOd100++;
        }
    }

    Console.WriteLine("Ukupno ima " + brojVecihOd100
        + " brzina u listi brzine čija je vrijednost veća od 100!");

    Console.ReadKey();
}
```



2. Napišite C# kod klase *Auto* koja mora imati tri atributa: *marka* (niz znakova, privatna), *OIB* vlasnika (niz znakova, privatna) i *cijena* (cijeli broj, javna).

Potrebno je napisati konstruktor klase koja prima argumente za sve attribute klase i postavlja ih, te se dodatno unutar konstruktora poziva i metoda *Ispis*.

Javna metoda *Ispis* unutar klase *Auto* treba ispisati sadržaj instance klase u formatu *Auto marke MARKA ima vlasnika kojemu je oib OIB i kojem je cijena CIJENA!* (znakove *MARKA*, *OIB* i *CIJENA* potrebno je zamijeniti s odgovarajućim vrijednostima).

Unutar *Main* dijela programa klase *Program* potrebno je napraviti dvije instance klase *auto* s proizvoljnim vrijednostima (**nije potrebno učitavanje s konzole**) te onoj instanci (*autu*) koji ima veću cijenu smanjiti iznos cijene za 5% i ispisati njegov sadržaj u konzolu koristeći metodu *Ispis*.

Rješenje:

```
static void Main(string[] args)
{
    Auto a1 = new Auto("BMW", "01234567891", 100000);
    Auto a2 = new Auto("Mercedes", "98765432109", 150000);

    if (a1.cijena > a2.cijena)
    {
        a1.cijena = Convert.ToInt32(Math.Round(a1.cijena * 0.95));
        a1.Ispis();
    }
    else if (a2.cijena > a1.cijena)
    {
        a2.cijena = Convert.ToInt32(Math.Round(a2.cijena * 0.95));
        a2.Ispis();
    }
    else
    {
        Console.WriteLine("Vozila imaju istu cijenu!");
        a1.Ispis();
        a2.Ispis();
    }

    Console.ReadKey();
}

class Auto
{
    private string marka;
    private string OIB;
    public int cijena;

    public Auto(string MARKA, string oib, int CIJENA)
    {
        marka = MARKA;
        OIB = oib;
        cijena = CIJENA;
        Ispis();
    }

    public void Ispis()
    {
        Console.WriteLine("Auto marke " + marka + ", ima vlasnika kojemu je oib "
            + OIB + " i kojem je cijena " + cijena + "!");
    }
}
```



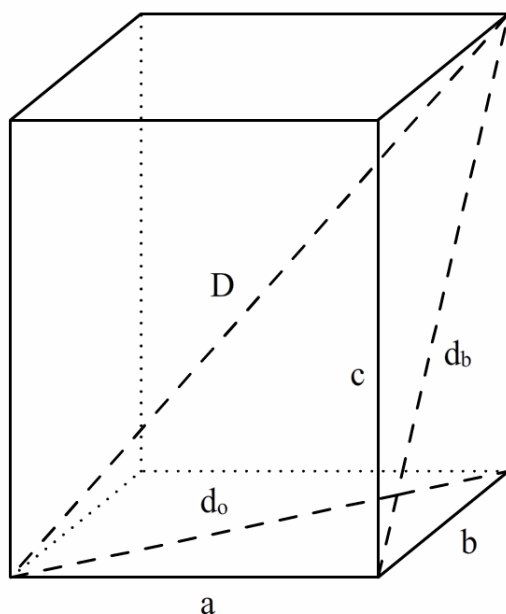
3. Napišite C# kod metode *dijagonaleKvadra* koja preko parametara prima tri brida kvadra a , b i c (brojevi s plivajućim zarezom).

Ukoliko su sve vrijednosti bridova veće od 0 metoda u konzolu ispisuje duljine dijagonala osnovne d_o i bočne strane d_b zaokružene na 2 decimale i **vraća** duljinu prostorne dijagonale D zaokružene na 3 decimale. Inače ništa ne ispisuje, a **vraća** vrijednost -1.

Format ispisa dijagonala osnovne i bočne strane treba biti *Za kvadar zadan bridovima $a=A$, $b=B$ i $c=C$ osnovna dijagonala iznosi $d_o=DO$, a bočna dijagonala iznosi $d_b=DB$* (znakove A , B , C , DO i DB potrebno je zamijeniti s odgovarajućim vrijednostima).

Jednadžbe za izračun dijagonala dane su izrazima navedenim uz sliku kvadra.

Unutar *Main* metode potrebno je 5 puta petljom po vlastitom izboru učitati vrijednosti bridova kvadra (a , b i c) s konzole i svaki put u proizvoljnom formatu ispisati prostornu dijagonalu kvadra za učitane bridove (ono što vrati metoda *dijagonaleKvadra*).



$$d_o = \sqrt{a^2 + b^2}$$

$$d_b = \sqrt{a^2 + c^2}$$

$$D = \sqrt{d_o^2 + c^2}$$



Rješenje:

```
static void Main(string[] args)
{
    for (int i = 1; i <= 5; i++)
    {
        Console.WriteLine("Unesite brid a " + i + ". kvadra: ");
        double a = Convert.ToDouble(Console.ReadLine());

        Console.WriteLine("Unesite brid b " + i + ". kvadra: ");
        double b = Convert.ToDouble(Console.ReadLine());

        Console.WriteLine("Unesite brid c " + i + ". kvadra: ");
        double c = Convert.ToDouble(Console.ReadLine());

        Console.WriteLine("D=" + dijagonaleKvadra(a, b, c));
    }

    Console.ReadKey();
}

static double dijagonaleKvadra(double a, double b, double c)
{
    if (a > 0 && b > 0 && c > 0)
    {
        double d0 = Math.Sqrt(Math.Pow(a, 2) + Math.Pow(b, 2));
        double db = Math.Sqrt(Math.Pow(a, 2) + Math.Pow(c, 2));
        double D = Math.Sqrt(Math.Pow(d0, 2) + Math.Pow(c, 2));

        Console.WriteLine("Za kvadar zadan bridovima a=" + a
            + ", b=" + b + " i c=" + c
            + " osnovna dijagonala iznosi do=" + Math.Round(d0, 2)
            + ", a bočna dijagonala iznosi db=" + Math.Round(db, 2)
            + "!");

        return Math.Round(D, 3);
    }

    return -1;
}
```



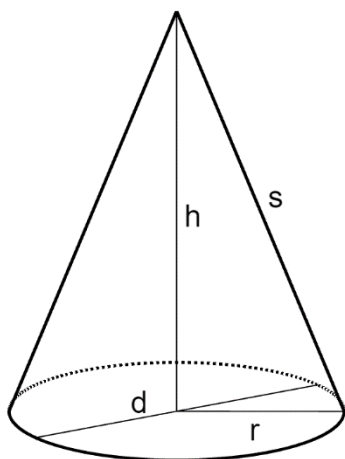
Kolokvij #1-20 (12. 04. 2022. – grupa B)

1. Napišite C# kod metode *IzracunZaStozac* koja kao argumente prima izbor unesen od korisnika (oplošje, volumen ili polumjer plašta) (niz znakova), polumjer stošca r i visinu stošca h (brojevi s plivajućim zarezm).

Ovisno što je predano u parametru *izbor* metoda treba u konzolu ispisati poruku u formatu kao u primjeru. Ukoliko je za izbor predana vrijednost volumen, a za polumjer i visinu su predane vrijednosti 2.5 i 5.2, metoda u konzolu treba ispisat *Volumen stošca određenog polumjerom $r=2.5$ i visinom $h=5.2$ iznosi $V=34.033920413889426!$.*

Jednadžbe potrebne za izračun prikazane su uz sliku stošca.

U *Main* metodi potrebno je 6 puta s tipkovnice učitati izbor (oplošje, volumen ili polumjer plašta) i dimenzije stošca (polumjer i visinu) koristeći petlju po vlastitom izboru, te svaki put pozvati metodu *IzracunZaStozac* za učitane vrijednosti.



$$V = \frac{1}{3}\pi r^2 h$$

$$O = \pi r(r + s)$$

$$s = \sqrt{r^2 + h^2}$$

V – obujam

O – oplošje

r – polumjer

s – polumjer plašta

h – visina



Rješenje:

```
static void Main(string[] args)
{
    for (int i = 1; i <= 6; i++)
    {
        Console.WriteLine("Što želite računati? ");
        string izbor = Console.ReadLine();

        Console.WriteLine("Unesite polumjer stošca: ");
        double r = Convert.ToDouble(Console.ReadLine());

        Console.WriteLine("Unesite visinu stošca: ");
        double h = Convert.ToDouble(Console.ReadLine());

        IzracunZaStozac(izbor, r, h);
    }

    Console.ReadKey();
}

static void IzracunZaStozac(string izbor, double r, double h)
{
    if (izbor == "oplošje")
    {
        double s = Math.Sqrt(Math.Pow(r, 2) + Math.Pow(h, 2));
        double O = Math.PI * r * (r + s);
        Console.WriteLine("Oplošje stošca određenog polumjerom r=" + r + " i visinom h=" + h + " iznosi O=" + O + "!.");
    }
    else if (izbor == "volumen")
    {
        double V = (1 / 3.0) * Math.PI * Math.Pow(r, 2) * h;
        Console.WriteLine("Volumen stošca određenog polumjerom r=" + r + " i visinom h=" + h + " iznosi V=" + V + "!.");
    }
    else
    {
        double s = Math.Sqrt(Math.Pow(r, 2) + Math.Pow(h, 2));
        Console.WriteLine("Polumjer plašta stošca određenog polumjerom r=" + r + " i visinom h=" + h + " iznosi s=" + s + "!.");
    }
}
```



2. Napišite C# kod klase *Dostavljac* koja sadrži privatne attribute ime (niz znakova), prezime (niz znakova) i listu masa paketa koje dostavljač treba dostaviti (lista cijelih brojeva).

Konstruktor klase prima argumente imena i prezimena, postavlja odgovarajuće vrijednosti atributa imena i prezimena, a listu masa paketa inicijalizira (u početku je lista prazna).

Unutar klase *Dostavljac* potrebno je napisati dvije javne metode: *DodajPaket* i *Ispis*.

Metoda *DodajPaket* preko argumenata prima masu paketa koju je potrebno dodati u listu mase paketa.

Metoda *Ispis* ispisuje sadržaj instance klase u formatu kao primjeru. Primjerice, neka je kreiran dostavljač čije je ime *Pero*, prezime *Perić* i u listi masa paketa ima vrijednosti 1, 5, 2 i 3, u konzolu je potrebno ispisati *Dostavljač Pero Perić je dostavio ukupno 4 paketa čije su mase redom: 1 5 2 3!*.

Unutar *Main* dijela programa potrebno je s konzole učitati ime i prezime dostavljača, kreirati instancu klase dostavljač s odgovarajućim vrijednostima, te potom tom dostavljaču (njegovoj listi masa paketa) dodati 3 mase paketa proizvoljnih iznosa (mase paketa **ne moraju** se učitati s konzole). Na kraju je potrebno pozvati metodu *Ispis* za kreiranog dostavljača.



Rješenje:

```
static void Main(string[] args)
{
    Console.Write("Unesite ime dostavljača: ");
    string ime = Console.ReadLine();

    Console.Write("Unesite prezime dostavljača: ");
    string prezime = Console.ReadLine();

    Dostavljac d = new Dostavljac(ime, prezime);

    d.DodajPaket(5);
    d.DodajPaket(10);
    d.DodajPaket(15);

    d.Ispis();

    Console.ReadKey();
}

class Dostavljac
{
    private string ime;
    private string prezime;
    private List<int> masePaketa;

    public Dostavljac(string IME, string PREZIME)
    {
        ime = IME;
        prezime = PREZIME;
        masePaketa = new List<int>();
    }

    public void DodajPaket(int MASA)
    {
        masePaketa.Add(MASA);
    }

    public void Ispis()
    {
        Console.Write("Dostavljač " + ime + " " + prezime
            + " je dostavio ukupno " + masePaketa.Count
            + " paketa čije su mase redom:");
        for (int i = 0; i < masePaketa.Count; i++)
        {
            Console.Write(" " + masePaketa[i]);
        }
        Console.WriteLine("!");
    }
}
```



3. Napišite C# kod konzolne aplikacije koja učitava n (broj redaka) i m (broj stupaca) vrijednosti s konzole koji određuju matricu $A = n \times m$.

S konzole je potrebno popuniti matricu A s ID-evima vozila (**niz znakova**).

Potom je potrebno ispisati sadržaj matrice unazad, od zadnjeg elementa u matrici prema prvom elementu redak po redak, u formatu kao u primjeru. Sadržaj koji se treba ispisati u svakom koraku je vrijednost elementa matrice (ID), pozicija elemenata u matrici (indeks retka i indeks stupca) i koliko ima znakova u ID-u.

Na kraju je u proizvoljnom formatu potrebno ispisati i koliko ima ID-eva u matrici čiji je prvi znak jednak znaku 1 (jedan znak).

Primjerice, ako se učitava matrica:

1	3	4	15
18	23	6	999
36	90	81	7
877	123459	44	11

Potrebno je ispisati:

ID 11 nalazi se na indeksu retka 3 i indeksu stupca 3 te ukupno ima 2 znaka!

ID 44 nalazi se na indeksu retka 3 i indeksu stupca 2 te ukupno ima 2 znaka!

ID 123459 nalazi se na indeksu retka 3 i indeksu stupca 1 te ukupno ima 6 znaka!

ID 977 nalazi se na indeksu retka 3 i indeksu stupca 0 te ukupno ima 3 znaka!

ID 7 nalazi se na indeksu retka 2 i indeksu stupca 3 te ukupno ima 1 znaka!

...

ID 1 nalazi se na indeksu retka 0 i indeksu stupca 0 te ukupno ima 1 znaka!

Ukupno ima 5 ID-eva koji počinju znakom 1!



Rješenje:

```
static void Main(string[] args)
{
    Console.WriteLine("Unesite broj redaka n: ");
    int n = Convert.ToInt32(Console.ReadLine());

    Console.WriteLine("Unesite broj stupaca m: ");
    int m = Convert.ToInt32(Console.ReadLine());

    string[,] A = new string[n, m];

    for (int i = 0; i < A.GetLength(0); i++)
    {
        for (int j = 0; j < A.GetLength(1); j++)
        {
            Console.WriteLine("Unesite ID vozila: ");
            A[i, j] = Console.ReadLine();
        }
    }

    int brojTrazenih = 0;

    for (int i = A.GetLength(0) - 1; i >= 0; i--)
    {
        for (int j = A.GetLength(1) - 1; j >= 0; j--)
        {
            Console.WriteLine("ID " + A[i, j]
                + " nalazi se na indeksu retka " + i
                + " i indeksu stupca " + j + " te ukupno ima "
                + A[i, j].Length + " znaka!");

            if (A[i, j][0] == '1')
            {
                brojTrazenih++;
            }
        }
    }

    Console.WriteLine("Ukupno ima " + brojTrazenih
        + " ID-eva koji počinju znakom 1!");

    Console.ReadKey();
}
```



Kolokvij #1-21 (12. 04. 2022. – grupa C)

1. Napišite C# kod metode *sumaZnamenkiBroja* koja kao argument prima cijeli broj, u konzolu ispisuje znamenku po znamenku predanog broja (svaku u zasebnom retku) i kao rezultat **vraća** zbroj znamenki predanog broja.

U *Main* metodi potrebno je učitavati brojeve sve dok se učitavaju brojevi čija je suma znamenki unutar intervala [0, 15]. Pomoć: ostatak dijeljenja s 10 i cjelobrojno dijeljenje s 10.

Primjerice, ukoliko se u nekoj iteraciji unese broj 6784, metoda treba ispisati:

Znamenke broja 6784 su:

4
8
7
6

Rješenje:

```
static void Main(string[] args)
{
    int suma = 0;
    do
    {
        Console.Write("Unesite broj: ");
        int broj = Convert.ToInt32(Console.ReadLine());

        suma = sumaZnamenkiBroja(broj);
    } while (suma >= 0 && suma <= 15);

    Console.ReadKey();
}

static int sumaZnamenkiBroja(int broj)
{
    string brojKaoRijec = Convert.ToString(broj);

    int zbrojZnamenki = 0;

    Console.WriteLine("Znamenke broja " + broj + " su:");

    do
    {
        int zadnjaZnamenka = broj % 10;
        zbrojZnamenki += zadnjaZnamenka;
        broj /= 10;
        Console.WriteLine(zadnjaZnamenka);
    } while (broj != 0);

    return zbrojZnamenki;
}
```



2. Napišite C# kod programa koji prebrojava koliko puta ste se nalazili na pojedinom mjestu matrice. Prvo je potrebno kreirati matricu $M = 10 \times 10$ (cijelih brojeva).

S tipkovnice je potrebno kontinuirano učitavati znak po znak, pri čemu se očekuje da se mogu unositi samo znakovi U, D, L i R (pomoć: učitani znak spremite u varijablu tipa niza znakova).

Početna pozicija unutar matrice je $(i, j) = (5, 5)$ (indeks retka i indeks stupca).

U svakoj iteraciji pozicija se pomiče za vrijednost 1 (jedan). U predstavlja pomicanje gore, D pomicanje dolje, L pomicanje u lijevo, a R pomicanje u desno. Učitavanje znakova s konzole traje dok god se trenutna pozicija nalazi unutar matrice.

Prilikom pomicanja pozicije potrebno je bilježiti koliko smo puta bili na pojedinoj poziciji u matrici (početna zadana pozicija se ne broji kao posjeta).

Na kraju je potrebno u konzolu ispisati cijelu matricu u formatu kao u primjeru. Primjerice, ako se redom učitavaju vrijednosti $U, U, U, D, L, L, R, R, R, R, U, U, U, U$, treba se ispisati matrica (početna pozicija označena je crvenom bojom).

0	0	0	0	0	0	0	1	0	0
0	0	0	0	0	0	0	1	0	0
0	0	0	0	0	1	0	1	0	0
0	0	0	1	2	3	1	1	0	0
0	0	0	0	0	1	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0



Rješenje:

```
static void Main(string[] args)
{
    int[, ] M = new int[10, 10];

    int redak = 5;
    int stupac = 5;

    do
    {
        Console.WriteLine("Unesite kretanje: ");
        string kretanje = Console.ReadLine();

        if (kretanje == "U")
        {
            redak--;
        }
        else if (kretanje == "D")
        {
            redak++;
        }
        else if (kretanje == "L")
        {
            stupac--;
        }
        else
        {
            stupac++;
        }

        if (redak >= 0 && redak < M.GetLength(0) && stupac >= 0
            && stupac < M.GetLength(1))
        {
            M[redak, stupac] += 1;
        }
        else
        {
            break;
        }
    } while (true);

    for (int i = 0; i < M.GetLength(0); i++)
    {
        for (int j = 0; j < M.GetLength(1); j++)
        {
            Console.Write(M[i, j] + "\t");
        }
        Console.WriteLine();
    }

    Console.ReadKey();
}
```



3. Napišite C# kod konzolne aplikacije koja računa dobit aviokompanije.

Potrebno je napraviti klasu *AvioKompanija* koja sadrži atribute ime prijevoznika (niz znakova, javan), broj sjedala (cijeli broj, javan), cijena karte (broj s plivajućim zarezom, privatan).

Klasa *AvioKompanija* **nema konstruktor**, ali ima dvije javne metode *dohvatiCijenuKarte* i *postaviCijenuKarte*. Metoda *dohvatiCijenuKarte* **vraća** vrijednost atributa cijena karte, dok metoda *postaviCijenuKarte* preko parametra koji joj se predaje postavlja vrijednost atributa cijena karte.

Unutar *Main* dijela programa klase *Program* potrebno je kreirati 2 proizvoljna avio prijevoznika (instance klase *AvioKompanija*), i popuniti sve atribute klase na temelju vrijednosti koje se učitavaju s konzole. Nakon učitavanja vrijednosti i punjenja atributa klase, izračunajte i ispišite ostvarenu dobit za svakog prijevoznika prema sljedećem izrazu:

$$\text{ostvarenaDobit} = \text{BrojSjedala} * \text{CijenaKarte}$$

Format ispisa treba biti *Dobit prijevoznika X iznosi Y!* (znakove X (X predstavlja ime prijevoznika) i Y potrebno je zamijeniti s odgovarajućim vrijednostima).



Rješenje:

```
static void Main(string[] args)
{
    Console.Write("Unesite ime 1. avio kompanije: ");
    string imePrijevoznika1 = Console.ReadLine();

    Console.Write("Unesite broj sjedala 1. avio kompanije: ");
    int brojSjedala1 = Convert.ToInt32(Console.ReadLine());

    Console.Write("Unesite cijenu karte 1. avio kompanije: ");
    double cijenaKarte1 = Convert.ToDouble(Console.ReadLine());

    Console.Write("Unesite ime 2. avio kompanije: ");
    string imePrijevoznika2 = Console.ReadLine();

    Console.Write("Unesite broj sjedala 2. avio kompanije: ");
    int brojSjedala2 = Convert.ToInt32(Console.ReadLine());

    Console.Write("Unesite cijenu karte 2. avio kompanije: ");
    double cijenaKarte2 = Convert.ToDouble(Console.ReadLine());

    AvioKompanija a1 = new AvioKompanija();
    a1.imePrijevoznika = imePrijevoznika1;
    a1.brojSjedala = brojSjedala1;
    a1.postaviCijenuKarte(cijenaKarte1);

    AvioKompanija a2 = new AvioKompanija();
    a2.imePrijevoznika = imePrijevoznika2;
    a2.brojSjedala = brojSjedala2;
    a2.postaviCijenuKarte(cijenaKarte2);

    double ostvarenaDobit1 = a1.brojSjedala * a1.dohvatiCijenuKarte();
    double ostvarenaDobit2 = a2.brojSjedala * a2.dohvatiCijenuKarte();

    Console.WriteLine("Dobit prijevoznika " + a1.imePrijevoznika + " iznosi "
        + ostvarenaDobit1 + "!");
    Console.WriteLine("Dobit prijevoznika " + a2.imePrijevoznika + " iznosi "
        + ostvarenaDobit2 + "!");

    Console.ReadKey();
}

class AvioKompanija
{
    public string imePrijevoznika;
    public int brojSjedala;
    private double cijenaKarte;

    public double dohvatiCijenuKarte()
    {
        return cijenaKarte;
    }

    public void postaviCijenuKarte(double CIJENAKARTE)
    {
        cijenaKarte = CIJENAKARTE;
    }
}
```



Kolokvij #1-23 (11. 04. 2023. – grupa A)

1. Napišite C# kod programa koji preko tipkovnice učitava broj redaka **n** i broj stupaca **m** matrice **M**. Potom program treba stvoriti matricu **M** brojeva s plivajućim zarezom i popuniti ju proizvoljnim vrijednostima s konzole.

Program treba u jednodimenzionalno polje brojeva **P** spremiti rezultat operacija zbrajanja po recima ili stupcima matrice **M**. Ako je matrica **M** kvadratna izvršava se operacija zbrajanja vrijednosti u stupcima, inače ako matrica **M** nije kvadratna izvršava se operacija zbrajanja vrijednosti po recima. Prilikom operacije zbrajanja u obzir se uzimaju samo vrijednosti matrice koje se nalaze u intervalu $[0, 200]$.

Na kraju je potrebno ispisati sadržaj polja **P** u proizvoljnom formatu.



Rješenje:

```
1 Console.WriteLine("Unesite broj redaka matrice n:");
2 int n = Convert.ToInt32(Console.ReadLine());
3 Console.WriteLine("Unesite broj stupaca matrice m:");
4 int m = Convert.ToInt32(Console.ReadLine());
5
6 double[,] M = new double[n, m];
7
8 for (int i = 0; i < M.GetLength(0); i++)
9 {
10     for (int j = 0; j < M.GetLength(1); j++)
11     {
12         Console.WriteLine("Unesite broj:");
13         double x = Convert.ToDouble(Console.ReadLine());
14
15         M[i, j] = x;
16     }
17 }
18 double[] P;
19 double suma = 0;
20 if (M.GetLength(0) == M.GetLength(1))
21 {
22     P = new double[M.GetLength(1)];
23
24     for (int i = 0; i < P.Length; i++)
25     {
26         suma = 0;
27         for (int j = 0; j < M.GetLength(0); j++)
28         {
29             if (M[j, i] >= 0 && M[j, i] < 200) suma += M[j, i];
30         }
31         P[i] = suma;
32     }
33 }
34 else
35 {
36     P = new double[M.GetLength(0)];
37
38     for (int i = 0; i < P.Length; i++)
39     {
40         suma = 0;
41         for (int j = 0; j < M.GetLength(1); j++)
42         {
43             if (M[i, j] >= 0 && M[i, j] < 200) suma += M[i, j];
44         }
45         P[i] = suma;
46     }
47 }
48
49 Console.WriteLine("Sadržaj polja P:");
50 for (int i = 0; i < P.Length; i++)
51 {
52     Console.WriteLine(P[i]);
53 }
```



2. Napišite C# kod klase **Nasreca** koja mora imati tri javna atributa: *dužina* (broj s plivajućim zarezom), *sirina* (broj s plivajućim zarezom) i *policijska uprava* (niz znakova).

Klasa dodatno treba imati 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 **2** proizvoljne instance klase **Nasreca** koristeći **petlju** po vlastitom izboru, dodajte ih u listu proizvoljnog naziva i ispišite sadržaj pojedine instance klase u konzolu koristeći napisanu metodu **ispis**.



Rješenje:

```
1 static void Main(string[] args)
2 {
3     List<Nesreca> listaNesreca = new List<Nesreca>();
4
5     for (int i = 1; i <= 2; i++)
6     {
7         Console.WriteLine("Unesite x koordinatu lokacije "
8             + i + ". prometne nesreće:");
9         double x = Convert.ToDouble(Console.ReadLine());
10
11         Console.WriteLine("Unesite y koordinatu lokacije "
12             + i + ". prometne nesreće:");
13         double y = Convert.ToDouble(Console.ReadLine());
14
15         Console.WriteLine("Unesite naziv policijske uprave koja
16             je evidentirala " + i + ". prometnu nesreću:");
17         string policijskaUprava = Console.ReadLine();
18
19         Nesreca n = new Nesreca();
20         n.duzina = x;
21         n.sirina = y;
22         n.policijskaUprava = policijskaUprava;
23
24         listaNesreca.Add(n);
25     }
26
27     Console.WriteLine("Podaci o nesrećama:");
28     for (int i = 0; i < listaNesreca.Count; i++)
29     {
30         listaNesreca[i].ispis();
31     }
32 }
33
34 class Nesreca
35 {
36     public double duzina;
37     public double sirina;
38     public string policijskaUprava;
39
40     public void ispis()
41     {
42         Console.WriteLine("Dužina: " + duzina + ", širina: "
43             + sirina + " i policijska uprava: " + policijskaUprava);
44     }
45 }
```



3. Napišite C# kod 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 1:

```
1 static void Main(string[] args)
2 {
3     bool rijecJeIspravna = true;
4
5     do
6     {
7         Console.WriteLine("Unesite neku riječ:");
8         string rijec = Console.ReadLine();
9
10        rijecJeIspravna = provjeraRijeci(rijec);
11
12        if (rijecJeIspravna)
13        {
14            Console.WriteLine("Unesena riječ JE ispravna!");
15        }
16        else
17        {
18            Console.WriteLine("Unesena riječ NIJE ispravna!");
19        }
20    } while (rijecJeIspravna);
21 }
22 static bool provjeraRijeci(string rijec)
23 {
24     rijec = rijec.ToLower();
25     bool rijecJeIspravna = true;
26
27     if (rijec.Length > 4)
28     {
29         for (int i = 0; i < rijec.Length; i++)
30         {
31             if (rijec[i] == 'a' || rijec[i] == 'e' || rijec[i] == 'i'
32                 || rijec[i] == 'o' || rijec[i] == 'u')
33             {
34                 rijecJeIspravna = false;
35                 break;
36             }
37         }
38     }
39     else
40     {
41         rijecJeIspravna = false;
42     }
43
44     return rijecJeIspravna;
45 }
```



Rješenje 2:

```
1 static void Main(string[] args)
2 {
3     bool rijecJeIspravna = true;
4
5     do
6     {
7         Console.WriteLine("Unesite neku riječ:");
8         string rijec = Console.ReadLine();
9
10        rijecJeIspravna = provjeraRijeci(rijec);
11
12        if (rijecJeIspravna)
13        {
14            Console.WriteLine("Unesena riječ JE ispravna!");
15        }
16        else
17        {
18            Console.WriteLine("Unesena riječ NIJE ispravna!");
19        }
20    } while (rijecJeIspravna);
21 }
22 static bool provjeraRijeci(string rijec)
23 {
24     rijec = rijec.ToLower();
25     bool rijecJeIspravna = true;
26     char[] zabranjenaSlova = { 'a', 'e', 'i', 'o', 'u' };
27
28     if (rijec.Length > 4)
29     {
30         for (int i = 0; i < rijec.Length; i++)
31         {
32             for (int j = 0; j < zabranjenaSlova.Length; j++)
33             {
34                 if (rijec[i] == zabranjenaSlova[j])
35                 {
36                     rijecJeIspravna = false;
37                     break;
38                 }
39             }
40         }
41     }
42     else
43     {
44         rijecJeIspravna = false;
45     }
46
47     return rijecJeIspravna;
48 }
```



Kolokvij #1-27 (23. 04. 2024. – grupa A)

1. Zadatak riješite unutar projekta naziva **Zadatak1**.

Napišite C# kôd programa koji preko tipkovnice popunjava matricu niza cijelih brojeva $M = 2 \times 2$ s proizvoljnim vrijednostima (cijelim brojevima).

Nakon unosa brojeva 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 znamenki sadrži u formatu: Broj *BROJ* sadrži *X* znamenki. (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):

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!



Rješenje:

```
1 int[, ] M = new int[2, 2];
2
3 for (int i = 0; i < M.GetLength(0); i++)
4 {
5     for (int j = 0; j < M.GetLength(1); j++)
6     {
7         Console.WriteLine("Unesite broj na poziciji (" + i + "," + j + "): ");
8         int x = Convert.ToInt32(Console.ReadLine());
9
10        M[i, j] = x;
11
12        int brojZnamenki = Convert.ToString(M[i, j]).Length;
13
14        if (brojZnamenki > 3)
15        {
16            Console.WriteLine("Broj " + M[i, j] + " sadrži " +
17                brojZnamenki + " znamenke!");
18        }
19        else
20        {
21            Console.WriteLine("Broj " + M[i, j] + " sadrži " +
22                brojZnamenki + " znamenke i broj je neispravan!");
23        }
24    }
25 }
```



2. Zadatak riješite unutar projekta naziva **Zadatak2**.

Napišite C# kôd programa koji s konzole kontinuirano učitava brzine (brojevi s plivajućim zarezom). Brzine se učitavaju sve dok se ne unese neispravna brzina. Prilikom unosa neispravne brzine, program izlazi iz petlje.

U suprotnom, ukoliko je brzina ispravna, brzina se dodaje na kraj liste naziva „**brzine**“. Brzina je ispravna ukoliko se nalazi u intervalu **[0, 200>**.

Nakon izlaska iz petlje, program ispisuje u proizvoljnom formatu najveću i najmanju brzinu spremljenu u listi „**brzine**“. Za dohvat traženih vrijednosti brzina trebate izraditi metode naziva „**MaxBrzina()**“ i „**MinBrzina()**“. Navedene dvije metode primaju kao ulazni parametar cijelu listu brzina, te nad tom listom izvršavaju potrebne operacije i na kraju vraćaju tražene vrijednosti. „**MaxBrzina()**“ kao rezultat vraća najveću brzinu prisutnu u listi, a „**MinBrzina()**“ kao rezultat vraća najmanju brzinu prisutnu u listi!

Primjer unosa i ispisa (napomena: format unosa/ispisa je proizvoljan):

1. Unesite 0. brzinu: 59
2. Unesite 1. brzinu: 85
3. Unesite 2. brzinu: 36
4. Unesite 3. brzinu: 47
5. Unesite 4. brzinu: 85
6. Unesite 5. brzinu: -1
7. Najveća unesena brzina u listi: 85
8. Najmanja unesena brzina u listi: 36



Rješenje:

```
1 static void Main(string[] args)
2 {
3     List<double> brzine = new List<double>();
4
5     do
6     {
7         Console.Write("Unesite " + brzine.Count + ". brzinu: ");
8         double brzina = Convert.ToDouble(Console.ReadLine());
9
10        if (brzina >= 0 && brzina < 200)
11        {
12            brzine.Add(brzina);
13        }
14        else
15        {
16            break;
17        }
18    } while (true);
19
20    Console.WriteLine("Najveća unesena brzina u listi: " + MaxBrzina(brzine));
21    Console.WriteLine("Najmanja unesena brzina u listi: " + MinBrzina(brzine));
22 }
23
24 static double MaxBrzina(List<double> brzine)
25 {
26     double najvecaBrzina = double.MinValue;
27     for (int i = 0; i < brzine.Count; i++)
28     {
29         if (brzine[i] > najvecaBrzina)
30         {
31             najvecaBrzina = brzine[i];
32         }
33     }
34
35     return najvecaBrzina;
36 }
37
38 static double MinBrzina(List<double> brzine)
39 {
40     double najmanjaBrzina = double.MaxValue;
41     for (int i = 0; i < brzine.Count; i++)
42     {
43         if (brzine[i] < najmanjaBrzina)
44         {
45             najmanjaBrzina = brzine[i];
46         }
47     }
48
49     return najmanjaBrzina;
50 }
```



3. Zadatak riješite unutar projekta naziva **Zadatak3**.

Napišite C# kôd konzolne aplikacije koja učitava cijeli broj **n** s tipkovnice, te potom u listu naziva **taksiji** dodaje **n** instanci klase *Taxi* pri čemu se vrijednosti atributa pojedine instance učitavaju s konzole.

Nakon učitavanja taksija u listu **taksiji**, potrebno je pronaći taksi koji je najbliži poziciji (10, 10) i ispisati njegovog vozača u konzolu u proizvoljnom formatu. Za izračun udaljenosti koristiti euklidsku udaljenost danu sljedećim izrazom:

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

Primjer unosa i ispisa (napomena: format unosa/ispisa je proizvoljan):

1. Unesite broj taksija n: 2
2. Unesite registraciju taksija na indeksu 0: 1337
3. Unesite vozača taksija na indeksu 0: Mufasa
4. Unesite x koordinatu taksija na indeksu 0: 1
5. Unesite y koordinatu taksija na indeksu 0: 2
6. Unesite registraciju taksija na indeksu 1: 185
7. Unesite vozača taksija na indeksu 1: Pumba
8. Unesite x koordinatu taksija na indeksu 1: 6
9. Unesite y koordinatu taksija na indeksu 1: 9
10. Vozač najbližeg taksija poziciji (10, 10) je: Mufasa

```
1 class Taxi
2 {
3     private string REGISTRACIJA;
4     public string VOZAC;
5     private double POZICIJA_X;
6     private double POZICIJA_Y;
7     public Taxi(string registracija, string vozac)
8     {
9         REGISTRACIJA = registracija;
10        VOZAC = vozac;
11    }
12    public void postaviX(double x)
13    {
14        POZICIJA_X = x;
15    }
16    public void postaviY(double y)
17    {
18        POZICIJA_Y = y;
19    }
20    public double dohvatiX()
21    {
22        return POZICIJA_X;
23    }
24    public double dohvatiY()
25    {
26        return POZICIJA_Y;
27    }
28 }
```



Rješenje:

```
1 Console.Write("Unesite broj taksija n: ");
2 int n = Convert.ToInt32(Console.ReadLine());
3
4 List<Taxi> taksiji = new List<Taxi>();
5 double najmanjaUdaljenost = double.MaxValue;
6 int indeksNajblizeg = 0;
7
8 for (int i = 0; i < n; i++)
9 {
10     Console.Write("Unesite registraciju taksija na indeksu " + i + ": ");
11     string registracija = Console.ReadLine();
12
13     Console.Write("Unesite vozača taksija na indeksu " + i + ": ");
14     string vozac = Console.ReadLine();
15
16     Console.Write("Unesite x koordinatu taksija na indeksu " + i + ": ");
17     double x = Convert.ToDouble(Console.ReadLine());
18
19     Console.Write("Unesite y koordinatu taksija na indeksu " + i + ": ");
20     double y = Convert.ToDouble(Console.ReadLine());
21
22     Taxi t = new Taxi(registracija, vozac);
23     t.postaviX(x);
24     t.postaviY(y);
25
26     taksiji.Add(t);
27
28     double trenutnaUdaljenost = Math.Sqrt(Math.Pow(10 - x, 2) + Math.Pow(10 - y, 2));
29     if (trenutnaUdaljenost < najmanjaUdaljenost)
30     {
31         najmanjaUdaljenost = trenutnaUdaljenost;
32         indeksNajblizeg = i;
33     }
34 }
35
36 Console.WriteLine("Vozač najbližeg taksija poziciji (10,10) je: "
37 + taksiji[indeksNajblizeg].VOZAC);
```



Rješenje – drugi način:

```
1 Console.Write("Unesite broj taksijski n: ");
2 int n = Convert.ToInt32(Console.ReadLine());
3
4 List<Taxi> taksiji = new List<Taxi>();
5
6 for (int i = 0; i < n; i++)
7 {
8     Console.Write("Unesite registraciju taksijski na indeksu " + i + ": ");
9     string registracija = Console.ReadLine();
10    Console.Write("Unesite vozača taksijski na indeksu " + i + ": ");
11    string vozac = Console.ReadLine();
12    Console.Write("Unesite x koordinatu taksijski na indeksu " + i + ": ");
13    double x = Convert.ToDouble(Console.ReadLine());
14    Console.Write("Unesite y koordinatu taksijski na indeksu " + i + ": ");
15    double y = Convert.ToDouble(Console.ReadLine());
16    Taxi t = new Taxi(registracija, vozac);
17    t.postaviX(x);
18    t.postaviY(y);
19    taksiji.Add(t);
20 }
21 double najmanjaUdaljenost = double.MaxValue;
22 int indeksTrazenog = 0;
23
24 for (int i = 0; i < taksiji.Count; i++)
25 {
26     double trenutnaUdaljenost = Math.Sqrt(
27         Math.Pow(10 - taksiji[i].dohvatiX(), 2)
28         + Math.Pow(10 - taksiji[i].dohvatiY(), 2));
29     if (trenutnaUdaljenost < najmanjaUdaljenost)
30     {
31         najmanjaUdaljenost = trenutnaUdaljenost;
32         indeksTrazenog = i;
33     }
34 }
35
36 Console.WriteLine("Vozač najbližeg taksijski poziciji (10, 10) je: "
37 + taksiji[indeksTrazenog].VOZAC);
```



4. Što će se ispisati nakon izvršavanja sljedećeg programskog odsječka?

```
1 int n = 8 % 5;  
2 int m = 2 * n;  
3 while (m > 1)  
4 {  
5     if (m % 2 == 0)  
6     {  
7         m++;  
8         n = m;  
9         m = m / 2;  
10    }  
11    else  
12    {  
13        n = m - 2;  
14        m = m - 2;  
15    }  
16    Console.WriteLine(n + ", " + m);  
17 }
```

Rješenje:

1	7, 3
2	1, 1



5. Što će se ispisati nakon izvršavanja sljedećeg programskog odsječka?

```
1 static void Main(string[] args)
2 {
3     int a = 7 % 4;
4     int b = a + 3;
5     M1(b, a);
6     Console.WriteLine(a + "," + b);
7 }
8 static void M1(int a, int b)
9 {
10    a = b;
11    Console.WriteLine(a + "," + b);
12    M3(a, b);
13 }
14 static void M2(int a, int b)
15 {
16    a = 3;
17    b = a;
18    Console.WriteLine(a + "," + b);
19 }
20 static void M3(int b, int a)
21 {
22    b = Convert.ToInt32(Math.Pow(a, 3));
23    Console.WriteLine(b + "," + a);
24 }
```

Rješenje:

1	3,3
2	27,3
3	3,6



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

```
1 int[, ] X = new int[4, 4] { { 1, 4, 5, 6 },  
2     { 3, 8, 4, 6 },  
3     { -5, 11, 8, 18 },  
4     { 7, -8, 10, 15 } };  
5 for (int j = 0; j < X.GetLength(0) - 1; j++)  
6 {  
7     int v = 0;  
8     for (int i = 1; i < 4; i++)  
9     {  
10        if (X[i, j] % 3 == 0 ^ X[i, j] > 4)  
11        {  
12            v++;  
13        }  
14    }  
15    Console.WriteLine(v);  
16 }
```

Rješenje:

1	2
2	2
3	2



Kolokvij #1-28 (23. 04. 2024. – grupa B)

1. Zadatak riješite unutar projekta naziva **Zadatak1**.

Napišite C# kôd programa koji preko tipkovnice učitava broj redaka **n** i broj stupaca **m** matrice **M**. Potom program treba stvoriti matricu **M** brojeva s plivajućim zarezom i popuniti ju s proizvoljnim vrijednostima s konzole.

Program treba u listu brojeva **ZbrojPoRetcima** spremiti rezultat operacija zbrajanja svih elemenata matrice **M** tako da svaki element liste predstavlja ukupan zbroj vrijednosti po promatranom retku matrice. Prilikom operacije zbrajanja u obzir se uzimaju samo vrijednosti matrice koje se nalaze u intervalu **[0, 200]**.

Na kraju je potrebno ispisati sadržaj liste **ZbrojPoRetcima** u proizvoljnom formatu.

Primjer unosa i ispisa (napomena: format unosa/ispisa je proizvoljan):

Primjer zbrajanja:

[2 5 6] -> 2 + 5 + 6 = 13

[3 6 7] -> 3 + 6 + 7 = 16

Primjer izvođenja programa:

1. Unesite broj redaka n: 2
2. Unesite broj stupaca m: 3
3. Unesite vrijednost 0,0: 2
4. Unesite vrijednost 0,1: 5
5. Unesite vrijednost 0,2: 6
6. Unesite Vrijednost 1,0: 3
7. Unesite vrijednost 1,1: 6
8. Unesite vrijednost 1,2: 7
9. Zbroj 0. retka iznosi 13.
10. Zbroj 1. retka iznosi 16.



Rješenje:

```
1 Console.WriteLine("Unesite broj redaka n: ");
2 int n = Convert.ToInt32(Console.ReadLine());
3
4 Console.WriteLine("Unesite broj stupaca m: ");
5 int m = Convert.ToInt32(Console.ReadLine());
6
7 double[,] M = new double[n, m];
8
9 List<double> ZbrojPoRetcima = new List<double>();
10
11 for (int i = 0; i < M.GetLength(0); i++)
12 {
13     double zbrojTrenutnogReda = 0;
14     for (int j = 0; j < M.GetLength(1); j++)
15     {
16         Console.WriteLine("Unesite vrijednost " + i + "," + j + ": ");
17         double x = Convert.ToDouble(Console.ReadLine());
18
19         M[i, j] = x;
20
21         if (M[i, j] >= 0 && M[i, j] < 200)
22         {
23             zbrojTrenutnogReda = zbrojTrenutnogReda + M[i, j];
24         }
25     }
26     ZbrojPoRetcima.Add(zbrojTrenutnogReda);
27 }
28
29 for (int i = 0; i < ZbrojPoRetcima.Count; i++)
30 {
31     Console.WriteLine("Zbroj " + i + ". retka iznosi "
32         + ZbrojPoRetcima[i]);
33 }
```



2. Zadatak riješite unutar projekta naziva **Zadatak2**.

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



Rješenje:

```
1 static void Main(string[] args)
2 {
3     List<Ured> uredi = new List<Ured>();
4
5     for (int i = 0; i < 2; i++)
6     {
7         Console.Write("Unesite površinu ureda na indeksu " + i + ": ");
8         double površina = Convert.ToDouble(Console.ReadLine());
9         Console.Write("Unesite broj stolova na indeksu " + i + ": ");
10        int brojStolova = Convert.ToInt32(Console.ReadLine());
11        Console.Write("Unesite identifikacijski broj ureda na indeksu "
12            + i + ": ");
13        int id = Convert.ToInt32(Console.ReadLine());
14
15        Ured u = new Ured(površina, brojStolova, id);
16
17        uredi.Add(u);
18        uredi[i].ispis();
19    }
20 }
21 class Ured
22 {
23     public double površina;
24     public int brojStolova;
25     public int ID;
26     public Ured(double POVRŠINA, int BROJSTOLOVA, int id)
27     {
28         površina = POVRŠINA;
29         brojStolova = BROJSTOLOVA;
30         ID = id;
31     }
32     public void ispis()
33     {
34         Console.WriteLine("Površina:" + površina + ", broj stolova:"
35             + brojStolova + " i ID:" + ID);
36     }
37 }
```



3. Zadatak riješite unutar projekta naziva **Zadatak3**.

Napišite C# kôd metode **ProvjeraBroja** koja vraća **istinu ili laž** ovisno o tome da li je broj (cijeli broj) predan kao argument ispravan ili ne. Broj je ispravan ako je broj znakova u broju strogo veći od **3** (pomoć: cijeli broj se može pretvoriti u tip podataka za niz znakova).

U glavnom dijelu programa potrebno je kontinuirano učitavati proizvoljne brojeve s tipkovnice sve dok se ne učitava neispravan broj. Nakon unosa neispravnog broja, program izlazi iz petlje. Za provjeru ispravnosti koristiti metodu *ProvjeraBroja*.

Nakon učitavanja pojedinog broja potrebno je ispisati u konzolu da li je ispravan ili ne u proizvoljnom formatu.

Primjer unosa i ispisa (napomena: format unosa/ispisa je proizvoljan):

Primjer ispravnog broja:

1. Unesite 1. broj: 1337
2. Učitani broj 1337 je ispravan!

Primjer neispravnog broja:

3. Unesite 2. broj: 133
4. Učitani broj 133 je neispravan!



Rješenje:

```
1 static void Main(string[] args)
2 {
3     int i = 1;
4
5     do
6     {
7         Console.Write("Unesite " + i + ". broj: ");
8         int x = Convert.ToInt32(Console.ReadLine());
9
10        if (ProvjeraBroja(x))
11        {
12            Console.WriteLine("Učitani broj " + x + " je ispravan!");
13            i++;
14        }
15        else
16        {
17            Console.WriteLine("Učitani broj " + x + " je neispravan!");
18            break;
19        }
20    } while (true);
21 }
22 static bool ProvjeraBroja(int x)
23 {
24     int brojZnakova = Convert.ToString(x).Length;
25
26     if (brojZnakova <= 3)
27     {
28         return false;
29     }
30
31     return true;
32 }
```



4. Što će se ispisati nakon izvršavanja sljedećeg programskog odsječka?

```
1 string s1 = " Marko ";
2 string s2 = "Mirko";
3 string s = s1 + s2;
4 int x1 = 0;
5 int x2 = 0;
6 int x3 = 0;
7 int x4 = 0;
8 for (int i = 0; i < s.Length; i++)
9 {
10     if (s[i] == ' ')
11     {
12         x1++;
13     }
14     else if (s[i] == 'o')
15     {
16         x2++;
17     }
18
19     if (s[i] == 'k')
20     {
21         x3++;
22     }
23     else
24     {
25         x4++;
26     }
27 }
28 Console.WriteLine(x1);
29 Console.WriteLine(x2);
30 Console.WriteLine(x3);
31 Console.WriteLine(x4);
```

Rješenje:

1	2
2	2
3	2
4	10



5. Što će se ispisati nakon izvršavanja sljedećeg programskog odsječka?

```
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 }
```

Rješenje:

1	10
2	10
3	5
4	8,5



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

```
1 List<int> lista = new List<int>();
2 for (int i = 32; i > 0; i = i / 2)
3 {
4     lista.Add(i);
5 }
6 lista.RemoveAt(lista.Count - 2);
7 lista.Insert(0, lista.Count);
8 lista.AddRange(new int[] { -2, 0, 3 });
9 for (int i = 0; i < lista.Count; i += 3)
10 {
11     Console.WriteLine(lista[i]);
12 }
```

Rješenje:

1	5
2	8
3	-2



Kolokvij #1-29 (23. 04. 2024. – grupa C)

1. Zadatak riješite unutar projekta naziva **Zadatak1**.

Napišite C# kôd programa koji kontinuirano učitava podatke vozila: duljinu (broj s plivajućim zarezom) i brzinu (cijeli broj) izmjerene radarom koji je postavljen na prometnici. Učitavanje podataka o vozilu treba ponavljati dok god je duljina vozila **veća od 0** i dok je brzina **veća od 0** (petlja po vlastitom izboru, s tim da se preporučuje korištenje petlje koja barem jednom izvrši naredbe prije provjere uvjeta).

U svakom koraku učitavanja, potrebno je podatke vozila spremati u dvije liste, i to: duljine vozila u listu **duljine**, a brzine vozila u listu **brzine**.

Nakon završetka učitavanja svih vozila potrebno je s konzole učitati proizvoljan indeks **i** te je potrebno ispisati duljinu i brzinu vozila određenog indeksom **i** samo ako je vrijednost indeksa **i** ispravna (indeks postoji u promatranoj listi).

Primjer unosa i ispisa (napomena: format unosa/ispisa je proizvoljan):

Primjer izvođenja programa:

7. Unesite duljinu 0. vozila: 5
8. Unesite brzinu 0. vozila: 70
9. Unesite duljinu 1. vozila: 7
10. Unesite brzinu 1. vozila: 0
11. Unesite indeks: 0
12. Na indeksu 0 duljina vozila je 5, a brzina vozila je 70!



Rješenje:

```
1 List<double> listaDuljina = new List<double>();
2 List<int> listaBrzina = new List<int>();
3
4 do
5 {
6     Console.WriteLine("Unesite duljinu " + listaBrzina.Count + ". vozila: ");
7     double duljina = Convert.ToDouble(Console.ReadLine());
8
9     Console.WriteLine("Unesite brzinu " + listaBrzina.Count + ". vozila: ");
10    int brzina = Convert.ToInt32(Console.ReadLine());
11
12    if (duljina > 0 && brzina > 0)
13    {
14        listaDuljina.Add(duljina);
15        listaBrzina.Add(brzina);
16    }
17    else
18    {
19        break;
20    }
21 } while (true);
22
23 Console.WriteLine("Unesite indeks: ");
24 int indeks = Convert.ToInt32(Console.ReadLine());
25
26 if (indeks >= 0 && indeks < listaDuljina.Count)
27 {
28     Console.WriteLine("Na indeksu " + indeks + " duljina vozila je "
29         + listaDuljina[indeks] + ", a brzina vozila je "
30         + listaBrzina[indeks] + "!");
31 }
```



2. Zadatak riješite unutar projekta naziva **Zadatak2**.

Napišite C# kôd metode **Prebroji** koja kao argument prima polje cijelih brojeva **brojevi**.

Program treba prebrojati koliko ima elemenata u polju **brojevi** koji su dozvoljeni – element je validan ako ima strogo više od **4** znamenke.

Metoda treba **vratiti** koliko takvih elemenata ima (pomoć: za provjeru broja znamenki cijelog broja moguće je cijeli broj pretvoriti u tip podataka za niz znakova).

Unutar glavnog dijela programa potrebno je učitati vrijednosti veličine polja **n** te stvoriti polje **brojevi** veličine **n**. Potom je potrebno popuniti polje proizvoljnim vrijednostima, iskoristiti metodu **Prebroji** i, na kraju, u proizvoljnom formatu ispisati rezultat.

Primjer unosa i ispisa (napomena: format unosa/ispisa je proizvoljan):

Primjer izvođenja programa:

1. Unese veličinu polja brojevi n: 4
2. Unesite broj na 0. indeks u polje brojevi: 10
3. Unesite broj na 1. indeks u polje brojevi: 3242
4. Unesite broj na 2. indeks u polje brojevi: 13
5. Unesite broj na 3. indeks u polje brojevi: 34545
6. Broj dozvoljenih vrijednosti u polju brojevi iznosi: 1!



Rješenje:

```
1 static void Main(string[] args)
2 {
3     Console.Write("Unese veličinu polja brojevi n: ");
4     int n = Convert.ToInt32(Console.ReadLine());
5
6     int[] brojevi = new int[n];
7
8     for (int i = 0; i < brojevi.Length; i++)
9     {
10         Console.Write("Unesite broj na " + i + ". indeks u polje brojevi: ");
11         brojevi[i] = Convert.ToInt32(Console.ReadLine());
12     }
13     Console.WriteLine("Broj dozvoljenih vrijednosti u polju
14     brojevi iznosi: " + Prebroji(brojevi) + "!");
15 }
16 static int Prebroji(int[] brojevi)
17 {
18     int brojTrazenih = 0;
19
20     for (int i = 0; i < brojevi.Length; i++)
21     {
22         if (Convert.ToString(brojevi[i]).Length > 4)
23         {
24             brojTrazenih++;
25         }
26     }
27
28     return brojTrazenih;
29 }
```



3. Zadatak riješite unutar projekta naziva **Zadatak3**.

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 argumente**, ali izračunava i **vraća** euklidski 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 metodi *izracunajDuzinu*. Format ispisa mora biti: *Duljina prometnice kategorije x iznosi Y!* (znakove X i Y potrebno je zamijeniti odgovarajućim vrijednostima).



Rješenje:

```
1 static void Main(string[] args)
2 {
3     Console.WriteLine("Unesite kategoriju");
4     string kategorija = Console.ReadLine();
5     Console.WriteLine("unesite xPocetak:");
6     double xPocetak = Convert.ToDouble(Console.ReadLine());
7     Console.WriteLine("unesite yPocetak:");
8     double yPocetak = Convert.ToDouble(Console.ReadLine());
9     Console.WriteLine("unesite xKraj:");
10    double xKraj = Convert.ToDouble(Console.ReadLine());
11    Console.WriteLine("unesite yKraj:");
12    double yKraj = Convert.ToDouble(Console.ReadLine());
13    Prometnica p = new Prometnica(kategorija, xPocetak, yPocetak, xKraj, yKraj);
14    Console.WriteLine("Duljina prometnice kategorije "
15        + kategorija + " iznosi " + p.izracunajDuzinu() + "!");
16 }
17 class Prometnica
18 {
19     private string kategorija;
20     private double xPocetak;
21     private double yPocetak;
22     private double xKraj;
23     private double yKraj;
24     public Prometnica(string KATEGORIJA, double XPOCETAK, double YPOCETAK,
25         double XKRAJ, double YKRAJ)
26     {
27         kategorija = KATEGORIJA;
28         xPocetak = XPOCETAK;
29         yPocetak = YPOCETAK;
30         xKraj = XKRAJ;
31         yKraj = YKRAJ;
32     }
33     public double izracunajDuzinu()
34     {
35         return Math.Sqrt(Math.Pow(xKraj - xPocetak, 2)
36             + Math.Pow(yKraj - yPocetak, 2));
37     }
38 }
```



4. Što će se ispisati nakon izvršavanja sljedećeg programskog odsječka?

```
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    Console.WriteLine(z + "," + y);  
18 }  
19 while (x > 0);
```

Rješenje:

1	11,6
2	22,0



5. Što će se ispisati nakon izvršavanja sljedećeg programskog odsječka?

```
1 double[] p = new double[7] { 1.2346, 3.257, 8.3678, 4.23345,  
2                               9.7865, 5.68974, 11.4678 };  
3 int i = 1;  
4 for (i = 0; i < p.Length - 1; i += 2)  
5 {  
6     Console.WriteLine(Math.Round(p[(i + 1) / 2], 2));  
7 }  
8 Console.WriteLine(p[p.Length - 2]);
```

Rješenje:

1	1.23
2	3.26
3	8.37
4	5.68974



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

```
1 static void Main(string[] args)
2 {
3     PravokutanTrokut p = new PravokutanTrokut(4, 8);
4     p.izracunajPovrsinu();
5     Console.WriteLine(p.izracunajPovrsinu(1, 3));
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)
14     {
15         A = a;
16         B = b;
17         povrsina = 0;
18         ispis();
19     }
20     public double izracunajPovrsinu(int a, int b)
21     {
22         double povrsina = (b * a) / 2;
23         ispis();
24         return povrsina;
25     }
26     public void izracunajPovrsinu()
27     {
28         povrsina = (A * B) / 2;
29         ispis();
30     }
31     public void ispis()
32     {
33         Console.WriteLine(povrsina);
34     }
35 }
```

Rješenje:

1	0
2	16
3	16
4	1
5	16



Kolokvij #1-30 (23. 04. 2024. – grupa D)

1. Zadatak riješite unutar projekta naziva **Zadatak1**.

Napišite C# kôd klase **Lokacija**.

Klasa *Lokacija* sadrži javne attribute **x** i **y** (brojevi s plivajućim zarezom) koji predstavljaju geografsku dužinu, odnosno širinu.

Konstruktor klase *Lokacija* prima argumente za **sve** attribute klase. Klasa *Lokacija* treba imati i javnu metodu **Ispis** koja ispisuje sve attribute objekta u formatu: *Lokacija (X,Y)* (znakove X i Y potrebno je dohvatiti iz odgovarajućih atributa klase).

Unutar glavnog dijela programa potrebno je s tipkovnice učitati podatke **dvije** proizvoljne lokacije (instance klase *Lokacija*) i za svaku u konzolu ispisati njen sadržaj koristeći metodu *Ispis*.

Primjer unosa i ispisa (napomena: format unosa/ispisa je proizvoljan):

Primjer izvođenja programa:

1. Unesite x 1. lokacije: 4
2. Unesite y 1. lokacije: 6
3. (ispis) Lokacija (4,6)
4. Unesite x 2. lokacije: 3
5. Unesite y 2. lokacije: 2
6. (ispis) Lokacija (3,2)



Rješenje:

```
1 static void Main(string[] args)
2 {
3     for (int i = 1; i <= 2; i++)
4     {
5         Console.Write("Unesite x " + i + ". lokacije: ");
6         double x = Convert.ToDouble(Console.ReadLine());
7         Console.Write("Unesite y " + i + ". lokacije: ");
8         double y = Convert.ToDouble(Console.ReadLine());
9         Lokacija l = new Lokacija(x, y);
10        l.Ispis();
11    }
12 }
13 class Lokacija
14 {
15     public double x;
16     public double y;
17     public Lokacija(double X, double Y)
18     {
19         x = X;
20         y = Y;
21     }
22     public void Ispis()
23     {
24         Console.WriteLine("Lokacija (" + x + ", " + y + ")");
25     }
26 }
```



2. Zadatak riješite unutar projekta naziva **Zadatak2**.

Napišite C# kôd programa koji preko tipkovnice proizvoljnim vrijednostima popunjava matricu cijelih brojeva dimenzije **M = 2 x 2**.

Nakon popunjavanja cijele matrice, potrebno je ispisati u konzolu najveću vrijednost spremljenu u svakom retku matrice (svaki redak zasebno). Ispis se vrši u proizvoljnom formatu.

Primjer unosa i ispisa (napomena: format unosa/ispisa je proizvoljan):

Primjer izvođenja programa:

1. Unesite (0,0) element: 2
2. Unesite (0,1) element: 4
3. Unesite (1,0) element: 5
4. Unesite (1,1) element: 11
5. (ispis) Najveća vrijednost 0. retka je 4!
6. (ispis) Najveća vrijednost 1. retka je 11!

Rješenje:

```
1 int[,] M = new int[2, 2];
2
3 for (int i = 0; i < M.GetLength(0); i++)
4 {
5     for (int j = 0; j < M.GetLength(1); j++)
6     {
7         Console.Write("Unesite (" + i + "," + j + ") element: ");
8         int x = Convert.ToInt32(Console.ReadLine());
9         M[i, j] = x;
10    }
11 }
12
13 for (int i = 0; i < M.GetLength(0); i++)
14 {
15     int max = int.MinValue;
16     for (int j = 0; j < M.GetLength(1); j++)
17     {
18         if (M[i, j] > max)
19         {
20             max = M[i, j];
21         }
22     }
23     Console.WriteLine("Najveća vrijednost " + i + ". retka je " + max);
24 }
```



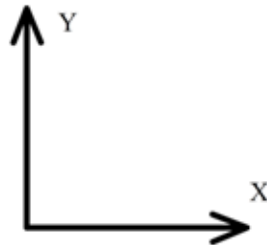
3. Zadatak riješite unutar projekta naziva **Zadatak3**.

Napišite C# kôd konzolne aplikacije koja simulira kretanje točke (objekta) na ekranu. Početna pozicija točke je (x_0, y_0) (brojevi s plivajućim zarezom) učitava se preko tipkovnice.

U svakoj iteraciji točka se pomiče za određenu vrijednost prema sljedećim pravilima:

- *W* predstavlja pomicanje gore za 3 (y-os)
- *S* pomicanje dolje za 4.5 (y-os)
- *A* pomicanje u lijevo za 1.5 (x-os)
- *D* pomicanje u desno za 1.5 (x-os)
- ukoliko se učitava neki od preostalih znakova ne događa se ništa.

Učitavanje znakova traje dok nije učitani znakovni niz **kraj**. Na kraju je potrebno u konzolu u proizvoljnom formatu ispisati konačnu poziciju točke. Razmatra se klasično postavljan koordinatni sustav prikazan na sljedećoj slici. Početna pozicija iznosi $x_0 = 0, y_0 = 0$.



Primjer unosa i ispisa (napomena: format unosa/ispisa je proizvoljan):

Primjer izvođenja programa:

1. Unesite znak: W
(ispis) Nova pozicija: (0,3)
2. Unesite znak: W
(ispis) Nova pozicija: (0,6)
3. Unesite znak: D
(ispis) Nova pozicija: (1.5,6)
4. Unesite znak: A
(ispis) Nova pozicija (0,6)
5. Unesite znak: S
(ispis) Nova pozicija: (0,1.5)
6. Unesite znak: A
(ispis) Nova pozicija: (-1.5,1.5)
7. Unesite znak: kraj
8. (ispis) Konačna pozicija točke je (-1.5,1.5)!



Rješenje:

```
1 Console.Write("Unesite x0: ");
2 double x = Convert.ToDouble(Console.ReadLine());
3
4 Console.Write("Unesite y0: ");
5 double y = Convert.ToDouble(Console.ReadLine());
6
7 do
8 {
9     Console.Write("Unesite znak: ");
10    string znak = Console.ReadLine();
11
12    if (znak == "kraj")
13    {
14        Console.WriteLine("Konačna pozicija točke: (" + x + "," + y + ")");
15        break;
16    }
17    else if (znak == "W")
18    {
19        y += 3;
20    }
21    else if (znak == "S")
22    {
23        y -= 4.5;
24    }
25    else if (znak == "A")
26    {
27        x -= 1.5;
28    }
29    else if (znak == "D")
30    {
31        x += 1.5;
32    }
33
34    Console.WriteLine("Nova pozicija: (" + x + "," + y + ")");
35 } while (true);
```



4. Što će se ispisati nakon izvršavanja sljedećeg programskog odsječka?

```
1 int n = 8 % 5;
2 int m = n / 2;
3 int x = 7 + 3 * 2;
4 int y = 2 - 9 / 2;
5 if (m == 1 && (n != 3 ^ true))
6 {
7     m = Convert.ToInt32(Math.Pow(n, 3));
8     n = m;
9     m--;
10    if (x / 2 >= 6 && !(y < 0))
11    {
12        n++;
13        x = n;
14        y = Convert.ToInt32(Math.Sqrt(m + 10));
15    }
16    else
17    {
18        y = Convert.ToInt32(Math.Sqrt(n + 9));
19        x = y;
20        y++;
21    }
22 }
23 else
24 {
25     n = n * 2;
26     m = Convert.ToInt32(Math.Pow(n, 2));
27     if (x / 2 < 6 && !(y >= 0))
28     {
29         x = n;
30         n++;
31         y = Convert.ToInt32(Math.Sqrt(m));
32     }
33     else
34     {
35         y = Convert.ToInt32(Math.Sqrt(n + 10));
36         x = y;
37         y++;
38     }
39 }
40 Console.WriteLine(x + "," + y + "," + n + "," + m);
```

Rješenje:

1	6, 7, 27, 26
---	--------------



5. Što će se ispisati nakon izvršavanja sljedećeg programskog odsječka?

```
1 static void Main(string[] args)
2 {
3     int[] p = new int[6] { 5, 4, 3, 8, -1, 2 };
4     p[p.Length - 1] = 1;
5     M(p);
6     for (int i = 0; i < p.Length; i += 2)
7     {
8         Console.WriteLine(p[i]);
9     }
10 }
11 private static void M(int[] q)
12 {
13     for (int i = 0; i < q.Length; i++)
14     {
15         if (q[i] % 2 == 1)
16         {
17             q[i] = q[i] - 1;
18         }
19     }
20 }
```

Rješenje:

1	4
2	2
3	-1

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

```
1 List<string> list = new List<string>();
2 list.AddRange(new string[] { "Ana", "Marko", "Miro" });
3 list.InsertRange(list.Count - 2, new string[] { "Tomo", "Orogoro" });
4 list.Remove("Ana");
5 list.RemoveAt(list.Count - 1);
6 for (int i = 0; i < list.Count; i++)
7 {
8     Console.WriteLine(list[i].Length);
9 }
```

Rješenje:

1	4
2	7
3	5



Priprema za drugi kolokvij – Rješenja

Kolokvij #2-24 (02. 06. 2020. – 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

```
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;
}
```



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 Šifra (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;
    }
}

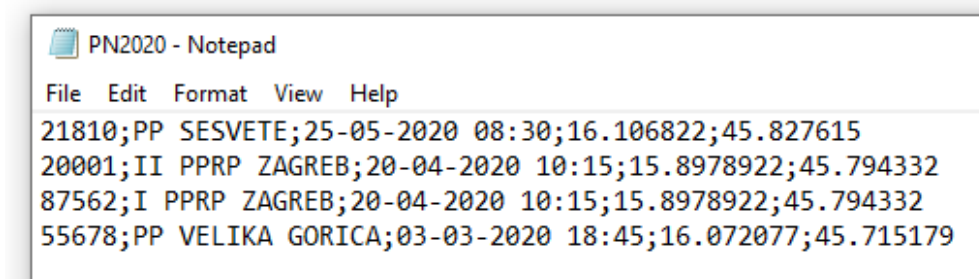
class PrometnaNesreca
{
    public int Sifra;
    public string Postaja;
    public string DatumVrijeme;
    public Lokacija LokacijaNesrece;

    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 + "!");
    }
}
```



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();
}

```



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";
    }

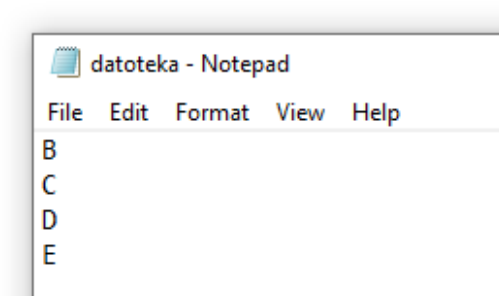
    Console.WriteLine("Učitana lokacija (" + x2 + ", " + y2 + ") je " + tekst
        + " zadanoj lokaciji prometne nesreće PN1!");

    Console.ReadKey();
}
```



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 (!citiranje.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]	



Kolokvij #2-25 (01. 06. 2021. – grupa A)

1. Što će biti zapisano u datoteci data.txt nakon što se izvrši naredba Flush na kraju sljedećeg programskog odsječka.

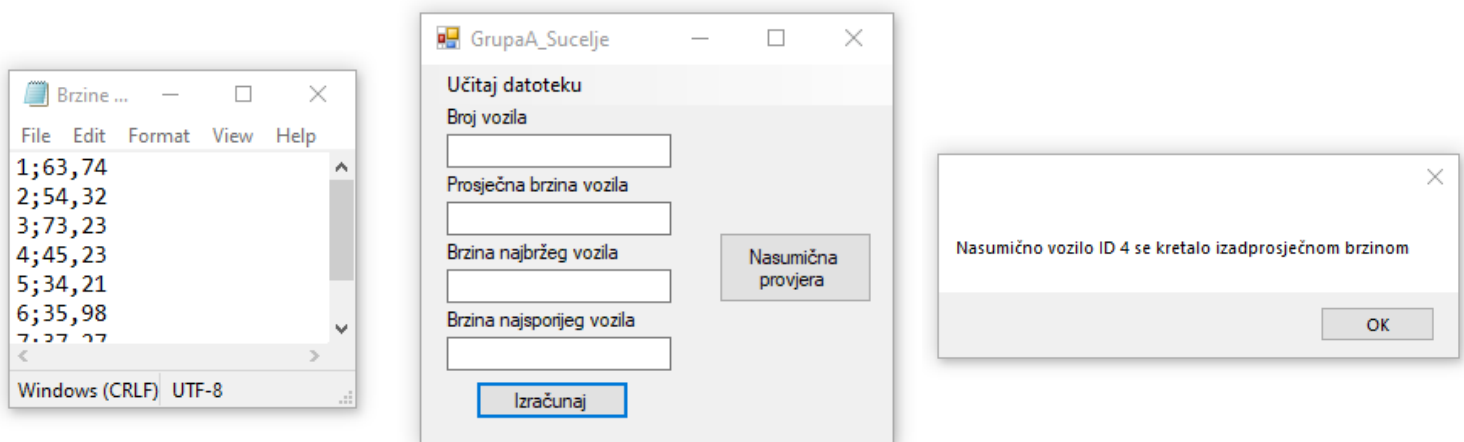
Kao točno rješenje priznaje se samo ono u kojem je vidljiv postupak i sadržaj varijabli u svakoj iteraciji, te u kojima je naznačen svaki trenutak poziva naredbe Flush i sadržaj upisan u datoteku nakon izvršavanja naredbe.

```
static void Main(string[] args)
{
    StreamWriter izlaz = new StreamWriter("data.txt");
    bool[] A = new bool[10];
    for (int i = 2; i <= 9; i++) A[i] = true;
    for (int i = 2; i <= 9; i++)
    {
        if (A[i])
        {
            izlaz.WriteLine(i);
            for (int j = i * 2; j <= 9; j = j + i)
            {
                A[j] = false;
            }
        }
    }
    izlaz.Flush();
    izlaz.Close();
}
```



2. Nadopunite C# kod sučelja GrupaA_Sucelje u nastavku, koje je detaljno prikazano na slici. U rješenju je potrebno naznačiti koji dio koda se odnosi na koju nadopunu (1, 2, 3 ili 4).

- Pritiskom na *Učitaj datoteku* potrebno je učitati datoteku *Brzine.txt*, koja se nalazi u mapi izvršne datoteke programa, u dvodimenzionalno polje, gdje će prvi stupac označavati identifikator vozila ID, a drugi stupac brzinu kretanja. Datoteka sadrži podatke o izmjerenim brzinama vozila u zasebnom retku u sljedećem formatu: ID;BRZINA, pri čemu ID predstavlja identifikator vozila (cijeli broj), BRZINA (broj s plivajućim zarezom) predstavlja izmjerenu brzinu vozila (km/h). Potrebno je koristiti *try-catch* blok te nije potrebno dodati imenski prostor *System.IO*. Unaprijed nije poznato koliko ima redaka u datoteci.
- Pritiskom na gumb *Izračunaj* potrebno je izračunati ukupan broj vozila, prosječnu brzinu svih vozila, najmanju i najveću izmjerenu brzinu. Navedene podatke potrebno je prikazati unutar odgovarajućih TextBoxova, *txtBrojVozila*, *txtProsjecnaBrzina*, *txtNajvecaBrzina* i *txtNajmanjaBrzina*. Nije dozvoljeno ponovno učitavanje podataka iz datoteke već je potrebno koristiti prethodno stvorene liste.
- Pritiskom na gumb *Nasumična provjera* iz polja se odabire nasumično vozilo te se uspoređuje njegova brzina s izračunatim prosjekom te se ispisuje pripadna poruka ukoliko je vozilo brže ili sporije od prosječne brzine. Za nasumičan odabir koristiti klasu *Random*.



```
public partial class Form1 : Form
{
    //Nadopuni - oznaka 1. dio
    public Form1()
    {
        InitializeComponent();
    }
    private void ucitajDatotekuToolStripMenuItem_Click(object sender, EventArgs e)
    {
        //Nadopuni - oznaka 2. dio
    }
    private void btnIzracunaj_Click(object sender, EventArgs e)
    {
        //Nadopuni - oznaka 3. dio
    }
    private void btnNasumicnaProvjera_Click(object sender, EventArgs e)
    {
        //Nadopuni - oznaka 4. dio
    }
}
```



Rješenje:

```
//Nadopuni - oznaka 1. dio
double[,] brzine;
double prosjecnaBrzina = 0.0;
public Form1()
{
    InitializeComponent();
}

private void ucitajDatotekuToolStripMenuItem_Click(object sender, EventArgs e)
{
    //Nadopuni - oznaka 2. dio

    try
    {
        //broj redaka je moguće izračunati na ovaj (lakši i brži) način:
        //int brojRedaka = File.ReadLines("Brzine.txt").Count();
        //ovaj način se ne uči na kolegiju pa je korišteno samo gradivo koje
        //ste učili

        int brojLinija = 0;
        StreamReader reader = new StreamReader("Brzine.txt");

        while (!reader.EndOfStream)
        {
            string linija = reader.ReadLine();
            brojLinija++;
        }

        brzine = new double[brojLinija, 2];
        int trenutniRedMatrice = 0;
        reader = new StreamReader("Brzine.txt");

        while (!reader.EndOfStream)
        {
            string[] dijeloviLinije = reader.ReadLine().Split(';');

            brzine[trenutniRedMatrice, 0] =
                Convert.ToDouble(dijeloviLinije[0]);
            brzine[trenutniRedMatrice, 1] =
                Convert.ToDouble(dijeloviLinije[1]);

            trenutniRedMatrice++;
        }

        reader.Close();
    }
    catch (Exception ex)
    {
        MessageBox.Show("Greška: " + ex.Message);
    }
}
```



```

private void btnIzracunaj_Click(object sender, EventArgs e)
{
    //Nadopuni - oznaka 3. dio

    int ukupanBrojVozila = brzine.GetLength(0);
    double najvecnaBrzina = double.MinValue;
    double najmanjaBrzina = double.MaxValue;
    double sumaBrzina = 0.0;

    for (int i = 0; i < brzine.GetLength(0); i++)
    {
        sumaBrzina += brzine[i, 1];

        if (brzine[i, 1] < najmanjaBrzina)
        {
            najmanjaBrzina = brzine[i, 1];
        }
        if (brzine[i, 1] > najvecnaBrzina)
        {
            najvecnaBrzina = brzine[i, 1];
        }
    }

    prosjecnaBrzina = sumaBrzina / ukupanBrojVozila;

    txtBrojVozila.Text = ukupanBrojVozila.ToString();
    txtProsjecnaBrzina.Text = prosjecnaBrzina.ToString();
    txtNajvecnaBrzina.Text = najvecnaBrzina.ToString();
    txtNajmanjaBrzina.Text = najmanjaBrzina.ToString();
}

private void btnNasumicnaProvjera_Click(object sender, EventArgs e)
{
    //Nadopuni - oznaka 4. dio
    Random rnd = new Random();

    int indeksNasumicnogVozila = rnd.Next(0, brzine.GetLength(0));

    string poruka = "Nasumično vozilo ID " + brzine[indeksNasumicnogVozila, 0]
        + " se kretalo ";

    if (brzine[indeksNasumicnogVozila, 1] < prosjecnaBrzina)
    {
        poruka += "ispodprosječnom";
    }
    else if (brzine[indeksNasumicnogVozila, 1] > prosjecnaBrzina)
    {
        poruka += "iznadprosječnom";
    }
    else
    {
        poruka += "prosječnom";
    }

    poruka += " brzinom";

    MessageBox.Show(poruka);
}
}

```



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

$$S(n) = \sum_{i=0}^n (i + 2), \forall n \geq 0$$

Unutar *Main* metode potrebno je pozvati rekurzivnu metodu s argumentom $n = 5$ te ispisati rezultat u formatu: $S(n) = x!$, gdje umjesto x treba ispisati odgovarajuću vrijednost.

Rješenje:

```
static void Main(string[] args)
{
    Console.WriteLine("S(5) = " + S(5) + "!");

    Console.ReadKey();
}

static int S(int n)
{
    if (n >= 0)
    {
        return n + 2 + S(n - 1);
    }

    return 0;
}
```

4. Kako će izgledati polje 90, 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 iteracija.

Rješenje:

zadano polje	90	8	-44	-6	99	-7
1. korak	-44	8	90	-6	99	-7
2. korak	-44	-7	90	-6	99	8
3. korak	-44	-7	-6	90	99	8
4. korak	-44	-7	-6	8	99	90
5. korak	-44	-7	-6	8	90	99
6. korak	-44	-7	-6	8	90	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.



Kolokvij #2-26 (01. 06. 2021. – grupa B)

1. Što će biti zapisano u datoteci *data.txt* neposredno prije nego što se pojedini put izvrši naredba *Flush* unutar *if* grananja u sljedećem programskom odsječku.

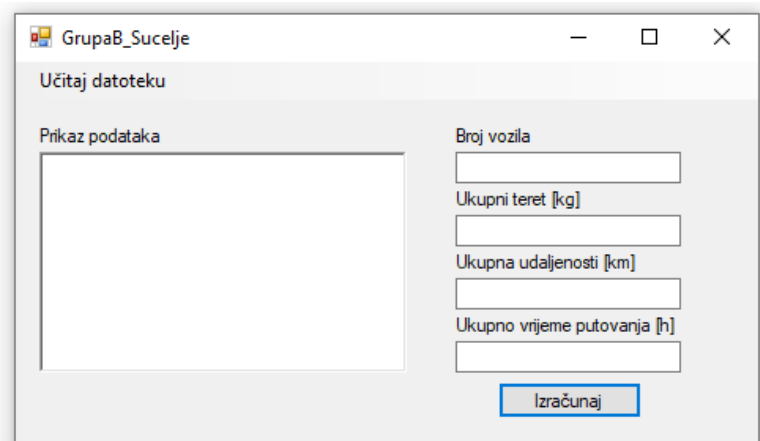
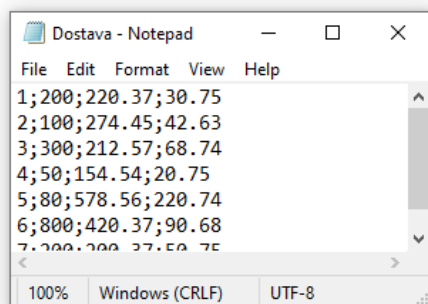
Kao točno rješenje priznaje se samo ono u kojem je vidljiv postupak i sadržaj varijabli u svakoj iteraciji, te u kojima je naznačen svaki trenutak poziva naredbe *Flush* i sadržaj upisan u datoteku neposredno prije izvršavanja naredbe.

```
static void Main(string[] args)
{
    StreamWriter izlaz = new StreamWriter("data.txt");
    for (int i = 6; i < 10; i++)
    {
        for (int j = i / 2; j < i; j += 2)
        {
            if (i - j == 2)
            {
                izlaz.Flush();
            }
            else {
                izlaz.WriteLine(i + j);
            }
        }
    }
    izlaz.Close();
}
```



2. Nadopunite C# kod sučelja *GrupaB_Sucelje* u nastavku, koje je detaljno prikazano na slici. U rješenju je potrebno naznačiti koji dio koda se odnosi na koju nadopunu (1, 2 ili 3).

- Pritiskom na *Učitaj datoteku* potrebno je učitati podatke iz datoteke *Dostava.txt* koja se nalazi na D disku u liste *udaljenosti*, *brzine* i *tereti* deklarirane na razini klase. Datoteka sadrži podatke o dostavnim vozilima u zasebnom retku u sljedećem formatu: ID;TERET;UDALJENOST;BRZINA, pri čemu ID predstavlja identifikator vozila (cijeli broj), TERET (cijeli broj) predstavlja ukupnu prijeđenu udaljenost u kilometrima koje je vozilo dostavilo taj dan, UDALJENOST (realan broj) predstavlja ukupnu prijeđenu udaljenost u kilometrima, dok BRZINA (realan broj) predstavlja prosječnu brzinu vozila (km/h). Učitano datoteku potrebno je liniju po liniju prikazivati unutar kontrole RichTextBox pod nazivom *richTxtPrikaz* (naredba za prelazak u novi red je niz znakova `\n`). Potrebno je koristiti *try-catch* blok te nije potrebno dodati imenski prostor *System.IO*. Unaprijed nije poznato koliko ima redaka u datoteci.
- Pritiskom na gumb *Izračunaj* potrebno je izračunati korišteni broj vozila za dostavu, ukupnu sumu prevezenog tereta (kg), ukupnu sumu prijeđenih udaljenosti svih vozila (km), te ukupno vrijeme trajanja dostave svih vozila (h) (za izračun vremena koristiti jednoliko gibanje). Navedene podatke potrebno je prikazati unutar odgovarajućih TextBox-ova, *txtBrojVozila*, *txtTeret*, *txtUdaljenost* i *txtVrijeme*. Nije dozvoljeno ponovno učitavanje podataka iz datoteke već je potrebno koristiti prethodno stvorene liste.



```
public partial class Form1 : Form
{
    //Nadopuni - oznaka 1. dio
    public Form1()
    {
        InitializeComponent();
    }

    private void ucitajDatotekuToolStripMenuItem_Click(object sender, EventArgs e)
    {
        //Nadopuni - oznaka 2. dio
    }

    private void btnIzracunaj_Click(object sender, EventArgs e)
    {
        //Nadopuni - oznaka 3. dio
    }
}
```



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

$$R(n) = \prod_{i=1}^n i, \forall n \geq 1$$

Unutar *Main* metode potrebno je pozvati rekurzivnu metodu s argumentom $n = 4$ te ispisati rezultat u formatu: „Umnožak brojeva od 1 do 4 iznosi: x!“, gdje umjesto x treba ispisati odgovarajuću vrijednost.

Rješenje:

```
static void Main(string[] args)
{
    Console.WriteLine("Umnožak brojeva od 1 do 4 iznosi: " + R(4) + "!");

    Console.ReadKey();
}

static int R(int n)
{
    if (n >= 1)
    {
        return n * R(n - 1);
    }

    return 1;
}
```



4. Kako će izgledati polje 11, 8, 6, -2, 4, -7 nakon cijele prve iteracije bubble sort algoritma, ukoliko se provodi sortiranje od najmanjeg do najvećeg člana. Potrebno je naznačiti svaku zamjenu elemenata polja zasebno u svim koracima prve iteracije te objasniti postupak.

Rješenje:

	1. iteracija sortiranja					
zadano polje	11	8	6	-2	4	-7
1. korak	8	11	6	-2	4	-7
2. korak	8	6	11	-2	4	-7
3. korak	8	6	-2	11	4	-7
4. korak	8	6	-2	4	11	-7
5. korak	8	6	-2	4	-7	11
6. korak	8	6	-2	4	-7	11

Objašnjenje: bubble sort je sortiranje zamjenom susjednih elemenata. Mijenjaju se vrijednosti dva susjedna elementa (ako je prvi element veći od drugog elementa). Sortiranje se završava kada prolaskom kroz cijeli niz ne bude zamjena.



Kolokvij #2-27 (01. 06. 2021. – grupa C)

1. Što će biti zapisano u datoteci *data.txt* neposredno prije nego što se pojedini put izvrši naredba *Flush* unutar *if* grananja u sljedećem programskom odsječku.

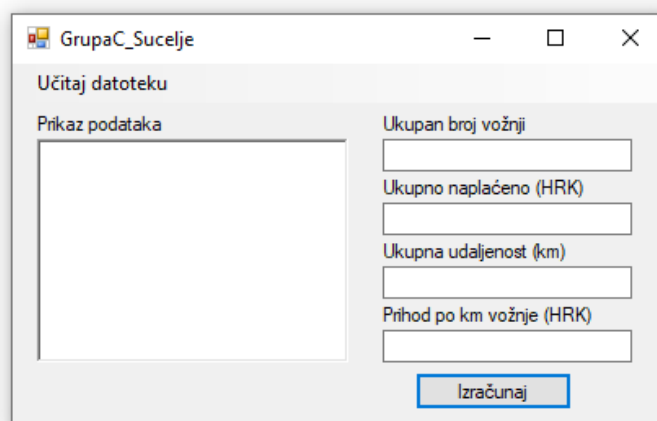
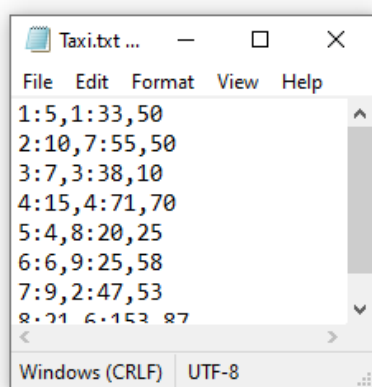
Kao točno rješenje priznaje se samo ono u kojem je vidljiv postupak i sadržaj varijabli u svakoj iteraciji, te u kojima je naznačen svaki trenutak poziva naredbe *Flush* i sadržaj upisan u datoteku neposredno prije izvršavanja naredbe.

```
static void Main(string[] args)
{
    StreamWriter izlaz = new StreamWriter("data.txt");
    for (int i = 6; i < 10; i++)
    {
        for (int j = i / 3; j < i; j += 2)
        {
            if (i - j == 2)
            {
                izlaz.Flush();
            }
            else
            {
                izlaz.WriteLine(i + j);
            }
        }
    }
    izlaz.Close();
}
```



2. Nadopunite C# kod sučelja *GrupaC_Sucelje* u nastavku, koje je detaljno prikazano na slici. U rješenju je potrebno naznačiti koji dio koda se odnosi na koju nadopunu (1, 2 ili 3).

- Pritiskom na *Dohvati podatke* potrebno je dohvatiti podatke iz datoteke *Taxi.txt* koja se nalazi na D disku u liste *voznja*, *udaljenosti* i *naplata* deklarirane na razini klase. Datoteka sadrži podatke o određenim taxi vožnjama u zasebnom retku u sljedećem formatu: ID:UDALJENOST:IZNOS, pri čemu ID predstavlja broj vožnje (cijeli broj), UDALJENOST (realan broj) predstavlja prijeđenu udaljenost u km za pripadnu vožnju, dok IZNOS (realan broj) predstavlja naplaćeni iznos u HRK za odrađenu vožnju. Učitanoj datoteku potrebno je liniju po liniju prikazivati unutar kontrole RichTextBox pod nazivom *richTxtPrikaz* (naredba za prelazak u novi red je niz znakova `\n`). Potrebno je koristiti try-catch blok te nije potrebno dodati imenski prostor *System.IO*. Unaprijed nije poznato koliko ima redaka u datoteci.
- Pritiskom na gumb *Izračunaj* potrebno je izračunati ukupan broj odrađenih vožnji, ukupan naplaćen iznos (HRK) za sve vožnje, ukupnu sumu prijeđenih udaljenosti svih vožnji (km), te ostvaren prihod u HRK po km vožnje. Navedene podatke potrebno je prikazati unutar odgovarajućih TextBox-ova *txtUkBrVoznji*, *txtNaplata*, *txtUdaljenost* i *txtPrihod*. Nije dozvoljeno ponovno učitavanje podataka iz datoteke već je potrebno koristiti prethodno stvorene liste.



```
public partial class Form1 : Form
{
    //Nadopuni - oznaka 1. dio

    public Form1()
    {
        InitializeComponent();
    }

    private void ucitajDatotekuToolStripMenuItem_Click(object sender, EventArgs e)
    {
        //Nadopuni - oznaka 2. dio
    }

    //3. dio
    private void btnIzracunaj_Click(object sender, EventArgs e)
    {
        //Nadopuni - oznaka 3. dio
    }
}
```



3. Potrebno je napisati kod C# programa koji će za dva zadana cijela broja pronaći njihovu najveću zajedničku mjeru korištenjem Euklidovog rekurzivnog algoritma.

Rješenje:

```
static void Main(string[] args)
{
    Console.Write("Unesite prvi broj: ");
    int a = Convert.ToInt32(Console.ReadLine());
    Console.Write("Unesite drugi broj: ");
    int b = Convert.ToInt32(Console.ReadLine());

    Console.WriteLine("Najveća zajednička mjera za a = " + a + " i b = "
        + b + " je " + NajvecaZajednickaMjera(a, b) + ".");

    Console.ReadKey();
}

static int NajvecaZajednickaMjera(int a, int b)
{
    if (b == 0)
    {
        return a;
    }
    else
    {
        return NajvecaZajednickaMjera(b, a % b);
    }
}
```

4. Polje 92, 34, 55, 47, 89, 67, 7 potrebno je sortirati od najmanjeg do najvećeg člana *insertion sort* algoritmom. Objasnite postupak sortiranja iteraciju po iteraciju do konačno sortiranog polja.

Rješenje:

zadano polje	92	34	55	47	89	67	7
1. korak	34	92	55	47	89	67	7
2. korak	34	55	92	47	89	67	7
3. korak	34	47	55	92	89	67	7
4. korak	34	47	55	89	92	67	7
5. korak	34	47	55	67	89	92	7
6. korak	7	34	47	55	67	89	92

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.

