

PROGRAMIRANJE u C++-u

- zbirka zadataka -

dr.sc. Marija Brkić Bakarić
Odjel za informatiku, Sveučilište u Rijeci
mbrkic@inf.uniri.hr

lektura: Božena Bašić Mikulić, prof.

Sadržaj

1	Dijagram tijeka i pseudokôd	3
2	Ulaz, izlaz i varijable	4
3	Naredba <i>if-else</i> i logički izrazi	7
4	Višestruko grananje u programu	10
1.	kolokvij	15
5	Uporaba petlji: <i>while</i> , <i>do-while</i> , <i>for</i>	17
6	Ugnježdene petlje	22
7	Polja i nizovi	27
8	Dvodimenzionalna polja	32
9	Strukture	36
10	Funkcije	40
11	Polja u funkcijama	43
12	Nadjačavanje funkcija	47
13	Datoteke	49
14	Rekurzija	53
15	Najčešće pogreške	57

1 Dijagram tijeka i pseudokôd

1. • Razvojni ciklus C++ programa.
2. • Zbrajanje dviju vrijednosti koje unosi korisnik.
3. • Upis u vozačku školu ukoliko osoba ima 17 godina i 6 mjeseci ili više.
4. Ispis uspjeha studenta na temelju postignutih bodova:
 - <40 pad
 - <60 prolaz
 - <70 dobro
 - ≥ 70 vrlo dobro
5. • Suma prvih n vrijednosti prirodnih brojeva.
6. Pronalaženje najvećeg zajedničkog djelitelja dva prirodna broja (Euklidov algoritam). Rješenje se traži sve dok je a veći od 0. Ako je $a < b$, potrebno je zamijeniti te dvije vrijednosti. Potom se a umanjuje za b . Nakon što a postane manji ili jednak 0, ispisuje se b .
7. Pronalaženje prosjeka u nekom skupu prirodnih brojeva.
8. • Pronalaženje najvećeg broja u nekom skupu.
9. Zbrajanje svih parnih brojeva između u intervalu $[1, 20]$.
10. Evidencija cijene najskupljeg i najjeftinijeg prodanog računala u trgovini računalima te ukupni broj prodanih računala i njihova prosječna vrijednost.

2 Ulaz, izlaz i varijable

1. • Ispravite greške u danom izvornom kôdu:

```
#include<iostream
using namespace std

int mian(
int age;
cout < < "Hello world!:-)\nYour age: "
cin>age;
return 0
```

2. • Napišite program koji ispisuje vaše ime i prezime i adresu u novoj liniji.

3. Napišite program u kojem su deklarirane i inicijalizirane 2 varijable na dva različita načina, a zatim se ispisuju te dvije vrijednosti.

4. • Napišite program koji od korisnika traži unos sata i minuta, a zatim ispisuje te dvije vrijednosti u sljedećem formatu: hh:mm.

5. Napišite program koji od korisnika traži unos trenutnog vremena u obliku *hh:mm:ss* i ispisuje koliko je sekunda prošlo od ponoći.

6. • Napišite program pomoću kojeg je moguće obaviti sljedeću interakciju s računalom.

```
Agentu, unesi svoj kod: _____
Unio si 123456789.
Verifikacija koda...
Kod verificiran!
Operacija "Z3" je aktivirana!
```

7. Napišite program u koji korisnik unosi troznamenasti broj, a program ispisuje njegove znamenke u obrnutom poretku.

8. • Napišite program koji evidentira stanje na dva tekuća računa. Iznos koji Mate ima na svom tekućem računu nalazi se u varijabli tipa *short* i iznosi maksimalnu moguću vrijednost. Iznos koji Ana ima na svom tekućem računu nalazi se u varijabli tipa *int* i jednak je Matinom. Dospjela im je oboma kamata od 1\$. Program ispisuje koliki iznos ima svatko od njih na tekućem računu prije i nakon dospjeća kamate.

9. Napišite program čiji ulaz je temperatura u stupnjevima Cezijusima, a izlaz temperatura u Farenheitima. U programu treba izračunati temperaturu na 2 načina – s i bez uvođenja nove varijable. Formula za konverziju:

$$Fahrenheit = 1.8 \times Celsius + 32$$

10.♦ Napišite program koji kao izlaz daje udaljenost izraženu u astronomskim jedinicama, a na temelju ulaza danog u svjetlosnim godinama. Vrijednost 63240 potrebno je definirati kao konstantu.

$$1 \text{ svjetlosna godina} = 63240 \text{ astronomskih jedinica}$$

11.♦ Napišite program koji traži od korisnika udaljenost koju je prešao u kilometrima i benzin koji je potrošio u litrama, a kao izlaz daje potrošnju automobila (broj litara na 100 km). Podesite izlaz tako da se prikazuju samo 2 decimale.

12.♦ Napišite program koji traži od korisnika da unese broj sekunda kao cjelobrojnu vrijednost i zatim ispiše ekvivalentno vrijeme u danima, satima, minutama i sekundama. Koristite simboličke konstante za prikaz broja sati u danu, broja minuta u satu i broja sekunda u minuti.

13.♦ Napišite program u koji se unose 2 cjelobrojne vrijednosti i ispisuje rezultat njihova dijeljenja.

14. Napišite program u koji korisnik unosi dva broja, a ispisuje se je li prvi broj djeljiv s drugim.

15. Napišite program u koji se unosi pet brojeva, a ispisuje se koliko ih je parnih.

16. Odredite izlaz iz sljedećeg programa.

```
#include<iostream>
using namespace std;

int main()
{
    int myAge=19, yourAge=19;
    cout<<"Ja imam " <<myAge<<" godina.\n";
    cout<<"Ti imaš " <<yourAge<<" godina.\n";
    myAge++;
    ++yourAge;
    cout<<"Prošla je jedna godina ... \n";
    cout<<"Ja imam " <<myAge<<" godina.\n";
    cout<<"Ti imaš " <<yourAge<<" godina.\n";
    cout<<"Jos jedna godina je prošla ... \n";
```

```
    cout<<"Ja imam " <<myAge++<<" godina.\n";
    cout<<"Ti imaš " <<++yourAge<<" godina.\n";
    cout<<"Provjerimo jos jednom koliko imamo godina
        ...\n";
    cout<<"Ja imam " <<myAge<<" godina.\n";
    cout<<"Ti imaš " <<yourAge<<" godina.\n";
    return 0;
}
```

17. Napišite program u koji korisnik učitava koordinate 3 točke ravnine, koje se zatim ispisuju u obliku (x_i, y_i) . Za zadane točke program provjerava određuju li trokut (tj. ne leže na istom pravcu) te ispisuje odgovarajuću poruku na zaslon. Ako točke određuju trokut, prema zadanoj formuli izračunava se i ispisuje površina trokuta:

$$P = \frac{1}{2} |(x_1 - x_2)(y_1 + y_2) + (x_2 - x_3)(y_2 + y_3) + (x_3 - x_1)(y_3 + y_1)|$$

3 Naredba *if-else* i logički izrazi

1.♦ Napišite program u koji korisnik unosi dva cijela broja, a program izračunava i ispisuje vrijednost logičke varijable koja je istinita ako su oba unesena broja parna.

2.♦ Napišite program u koji korisnik unosi cjelobrojnu vrijednost, a program ispisuje je li riječ o parnom ili neparnom broju.

3.♦ Napišite program u koji korisnik unosi tri cjelobrojne vrijednosti. Ispisuje se vrijednost logičke varijable koja je istinita ako je a dvostruko veći od b ili ako c nije iz segmenta $[1, 15]$.

4.♦ Napišite program u koji se unosi datum, mjesec i godina rođenja (korisnika treba porukama upozoriti kada unosi koju od vrijednosti). Vrijednost logičke varijable je istinita ako je osoba rođena u periodu od 1990. do 2001. na neparan datum u mjesecu rujnu.

5. Napišite program u koji korisnik unosi tri cjelobrojne vrijednosti. Ispisuje se vrijednost logičke varijable koja je istinita ako je a neparan broj i b djeljiv s c , ili je a u intervalu $[0, 10]$, a c nije u intervalu $(5, 10)$.

6. Napišite program u koji korisnik unosi tri cjelobrojne vrijednosti. Ispisuje se vrijednost logičke varijable koja je istinita ako je x paran broj koji nije djeljiv sa z ili ako y nije iz segmenta $[0, 25]$.

7. Napišite program u koji korisnik unosi tri cjelobrojne vrijednosti. Ispisuje se vrijednost logičke varijable koja je istinita ako je a paran broj veći od b ili ako c nije iz segmenta $[0, 100]$.

8. Napišite program u koji korisnik unosi tri cjelobrojne vrijednosti. Ispisuje se vrijednost logičke varijable koja je istinita ako je a paran broj veći ili jednak b , a c se nalazi u intervalu $[2, 10]$.

9. Napišite program u koji korisnik unosi tri cjelobrojne vrijednosti. Ispisuje se vrijednost logičke varijable koja je istinita ako je b paran broj koji nije djeljiv s c , dok je a u intervalu $[-20, 20]$.

10. Napišite program u koji korisnik unosi tri cjelobrojne vrijednosti. Ispisuje se vrijednost logičke varijable koja je istinita ako je y paran broj koji je djeljiv s x ili ako z nije iz segmenta $[10, 20]$.

- 11.** Napišite program u koji korisnik unosi tri cjelobrojne vrijednosti. Ispisuje se vrijednost logičke varijable koja je istinita ako je x paran broj koji je jednak y , a apsolutna vrijednost z podijeljena s 10 nije iz segmenta $[1, 10)$.
- 12.** Napišite program u koji se unosi datum, mjesec i godina rođenja (korisnika treba porukama upozoriti kada unosi koju od vrijednosti). Vrijednost logičke varijable je istinita ako je osoba rođena 1. siječnja 1980. ili nakon Dana državnosti (25.06.).
- 13.** Napišite program u koji se unosi datum, mjesec i godina rođenja (korisnika treba porukama upozoriti kada unosi koju od vrijednosti). Vrijednost logičke varijable je istinita ako je osoba rođena prije 2000. godine za vrijeme kalendarskog trajanja ljeta.
- 14.** Napišite program u koji se unosi datum, mjesec i godina rođenja (korisnika treba porukama upozoriti kada unosi koju od vrijednosti). Vrijednost logičke varijable je istinita ako je osoba rođena poslije 1990. godine u horoskopskom znaku Ovna.
- 15.♦** Napišite program u koji se unosi kapacitet sobe i broj ljudi koji će doći na sastanak. Program ispisuje koliko ljudi još može doći ili koliko ih je previše ako je premašen kapacitet.
- 16.** Napišite program u koji korisnik unosi tri znaka, a program provjerava i ispisuje jesu li unesena slova permutacija riječi CIA (velika ili mala slova).
- 17.** Napišite program koji od korisnika traži da učitava godinu te provjerava je li godina prijestupna. Program ispisuje vrijednost 1 ako je godina prijestupna, a 0 ako nije. Prema našem Gregorijanskom kalendaru godina je prijestupna ako je djeljiva s 4, osim ako se radi o stoljetnoj godini jer tada mora biti djeljiva s 400.
- 18.** Napišite program u kojem se logičkoj varijabli p dodjeljuje vrijednost *true* ako se pravokutnik sa stranicama c i d može cijeli smjestiti u pravokutnik sa stranicama a i b (stranice su im paralelne). Treba uzeti u obzir i rotaciju pravokutnika za 90° .
- 19.** U program se unosi datum, mjesec i godina rođenja (korisnika treba porukama upozoriti kada unosi koju od vrijednosti) kako bi se utvrdilo može li dijete poći u školu. Logička varijabla ima vrijednost *true* ako dijete može upisati školu. Dijete ima pravo upisati školu ako je rođeno prije 3. mjeseca 2005. godine.
- 20.** Napišite program u koji korisnik unosi 3 prirodna broja manja od 100. Ako je bilo koja vrijednost veća od 100 ili manja od 1, završava se izvođenje programa. U suprotnom, program provjerava jesu li ta tri broja Pitagorini

brojevi: $a^2 + b^2 = c^2$.

4 Višestruko grananje u programu

1. Ispravite logičku pogrešku u danom kôdu.

```
#include<iostream>
using namespace std;

int main()
{
    int x;
    cout<<"Unesi broj manji od 10 ili veci od 100: ";
    cin>>x;
    if (x>10)
        if (x>100)
            cout<<"Broj je redujer je veci od 100!\n";
        else
            cout<<"Broj je redujer je manji od 10.\n";
    return 0;
}
```

2. • Napišite program u koji korisnik unosi broj, a ispisuje se je li riječ o jednoznamenastom, dvoznamenkastom, troznamenkastom, četveroznamenastom ili višeznamenastom broju.

3. • Napišite program u koji korisnik unosi znak, a ispisuje se je li uneseni znak brojka, veliko slovo, malo slovo ili nešto drugo.

4. • Napišite program u koji korisnik unosi broj, a ispisuje se naziv dana u tjednu, ovisno o unesenom broju. U slučaju krivog unosa, ispisuje se odgovarajuća poruka.

5. • Napišite program koji za dane cjelobrojne vrijednosti a i b izračunava i ispisuje c prema formuli:

- $c = a + b$ za neparni a i neparni b
- $c = a - b$ za neparni a i parni b
- $c = a * b$ za parni a i parni b
- $c = a/b$ za parni a i neparni b

6. Napišite program koji od korisnika traži unos cjelobrojne vrijednosti x . Program računa i ispisuje vrijednost y , koju dobiva na sljedeći način:

- $y = x * 2$ ako je x neparan i djeljiv s brojem 3
- $y = x/2$ ako je x paran broj i ne pripada segmentu $[10, 100]$
- $y = 0$ inače

7. Napišite program koji od korisnika traži unos cjelobrojnih vrijednosti a i b . Program računa i ispisuje vrijednost c , koju dobiva na sljedeći način:

- $c = a + b$ ako su a i b neparni brojevi koji pripadaju segmentu $[0, 50]$
- $c = a - b$ ako su a i b neparni brojevi koji ne pripadaju segmentu $[10, 100]$
- $y = 0$ inače

8.♦ Napišite program koji od korisnika traži unos realne vrijednosti x . Program računa i ispisuje vrijednost funkcije $f(x)$. Funkcija je definirana kako slijedi:

- $f(x) = x * 10$ ako je x element segmenta $[1, 20]$
- $f(x) = x + 2$ ako je x element segmenta $[21, 40]$
- $f(x) = x/3$ ako je x element segmenta $[41, 60]$
- $f(x) = 0$ za sve ostale vrijednosti x

9.♦ Napišite program koji od korisnika traži unos realne vrijednosti x . Program računa i ispisuje vrijednost funkcije $f(x)$. Funkcija je definirana kako slijedi:

- $f(x) = x * 5$ ako je x element segmenta $[1, 100]$
- $f(x) = x + 5$ ako je x element segmenta $[-100, -1]$
- $f(x) = x/5$ ako je x element segmenta $[150, 200]$
- $f(x) = 0$ za sve ostale vrijednosti x

10.♦ Napišite program koji od korisnika traži unos cjelobrojnih vrijednosti a i b . Program računa i ispisuje vrijednost c , koju dobiva na sljedeći način:

- $c = a + b$ ako je a djeljiv s 10 i neparni b
- $c = a$ ako je a djeljiv s 10 i parni b
- $c = a * b$ ako a nije djeljiv s 10 i neparni b
- $c = a/b$ za sve ostale vrijednosti a i b

11. Napišite program koji od korisnika traži unos cjelobrojnih vrijednosti a i b . Program računa i ispisuje vrijednost c koju dobiva na sljedeći način:

- $c = a + b$ ako je a djeljiv s 10 i b element intervala $[10, 20]$
- $c = a$ ako a nije djeljiv s 10 i b je element intervala $[10, 20]$
- $c = a * b$ ako je a djeljiv s 10 i b nije element intervala $[10, 20]$
- $c = a/b$ za sve ostale vrijednosti a i b

12. Napišite program za rješavanje linearne jednadžbe $a * x + b = 0$. Brojevi a i b su realni brojevi. Posebno razmatrati slučajeve:

- $a = 0, b = 0$ Svaki broj je rješenje linearne jednadžbe.
- $a = 0, b \neq 0$ Linearna jednadžba nema rješenja.
- $a \neq 0, b \neq 0$ $x = -b/a$

Test podaci (a, b, x): $(1, -3, 3)$, $(-2, 5, 2.5)$, $(0, 0, \text{svaki broj})$, $(0, 10, \text{nema rješenja})$.

13. Napišite program za izračunavanje kvadratne jednadžbe $ax^2 + bx + c = 0$. Korisnik unosi realne vrijednosti a , b i c . Ukoliko su rješenja realna, ona se ispisuju, a ako nisu, ispisuje se odgovarajuća poruka.

14. Top u šahu «napada» sve što se na ploči nalazi u istom retku i istom stupcu s njim. Pozicija figure na šahovskoj ploči određena je slovom A-H (x-os) i brojem 1-8 (y-os), a šahovska ploča je prvi kvadrant koordinatnog sustava. Napišite program u koji korisnik učitava poziciju topa i poziciju neke druge figure, a ispisuje se da napada li je top ili ne. Ako je ne napada, ispisuje se i relativna pozicija figure u odnosu na top, npr. «gore desno» ili «dolje lijevo».

15. Napišite program u koji korisnik učitava tri prirodna broja manja od 30. Ako neki od brojeva nije prirodni broj manji od 30, program ispisuje poruku «Neispravan unos brojeva» i završava izvođenje. Ako su brojevi ispravno zadani, program ispituje mogu li oni biti duljine stranica trokuta i ispisuje odgovarajuću poruku.

UPUTE: Za stranice trokuta vrijedi da je suma duljina bilo koje dvije stranice veća od duljine treće stranice.

16. Napišite program u koji korisnik unosi vrijednost znakovnog tipa, a prikazuje se kodiran znak. Ako je znak slovo ili znamenka, treba ga zamijeniti sljedećim znakom. Veliko slovo 'Z' treba zamijeniti s velikim slovom 'A', malo slovo 'z' s malim slovom 'a', a brojku '9' s brojkom '0' ('1' -> '2', 'C' -> 'D', 'p' -> 'q', ...). Znakovi različiti od slova i znamenki zamjenjuju se znakom '*'.

17.♦ Napišite program u koji korisnik učitava tri prirodna broja iz intervala od 1 do 100. Ako brojevi nisu iz zadanog intervala (barem jedan broj), program ispisuje poruku «Broj nije iz zadanog intervala» i završiti izvođenje. Ako su brojevi unutar intervala, program ispisuje učitane brojeve po veličini od najvećeg prema najmanjem.

18.♦ Program traži od korisnika unos 2 cijela broja. Ispisuje se sljedeći izbornik:

1. Zbrajanje (+)
2. Oduzimanje (-)
3. Množenje (*)
4. Dijeljenje (/)

Korisnik unosi broj pored željene opcije, a program ispisuje rezultat izvođenja odabrane operacije nad unesenim operandima.

Napomena: Predvidjeti dijeljenje s nulom!

19.♦ Modificirajte prethodni zadatak tako da korisnik može unijeti ili redni broj opcije (1, 2, 3, 4) ili operator (+, -, *, /).

20.♦ Napišite program koji od korisnika traži unos starosti (broj godina), a za unesenu starost ispisuje jednu od sl. • • • • • sljedećih poruka:

- | | |
|------------|---------------|
| • 0 - 6 | malo dijete |
| • 7 - 14 | osnovnoškolac |
| • 15 - 18 | srednjoškolac |
| • 19 - ... | odrasla osoba |

21.♦ Napišite program u koji korisnik unosi starosti neke osobe. Program provjerava i ispisuje u koju od tri kategorije ulazi osoba s obzirom na starost:

- | | |
|------------|---------------------------------------|
| • 0 - 17 | Osoba je premlada za vozačku dozvolu. |
| • 18 - 80 | Osoba može imati vozačku dozvolu. |
| • 81 - 110 | Osoba je prestara za vozačku dozvolu. |

Prilikom unosa starosti osobe potrebno je provjeriti je li starost u rasponu od 1 do 110 godina. Ukoliko korisnik nije upisao točan podatak, program ispisuje poruku «KRIVI UNOS GODINA!» i izlazi van bez provjere.

22.♦ Napišite program u koji korisnik unosi mjesec i godinu rođenja za osobu A i za osobu B. Godina mora biti element segmenta [1950, 2004] (upozoriti korisnika na ispravan unos). Ukoliko godina nije iz zadanog intervala ili mjesec nije dobro unesen (1, 2, ..., 12), korisnika se upozorava da nije dobro upisao brojeve i prekida se izvođenje programa. Ukoliko je korisnik upisao dobre brojeve, na ekranu se ispisuje poruka koja osoba je starija.

23. Napišite program u koji korisnik unosi dvoznamenkasti prirodni broj. U slučaju krivog unosa, program prekida izvođenje, a inače se računa i ispisuje suma i umnožak znamenki unesenog broja. Ukoliko je suma znamenki paran broj, program računa i ispisuje ostatak pri dijeljenju te dvije znamenke, a ukoliko je neparan broj, računa se i ispisuje rezultat cjelobrojnog dijeljenja.

24. Napišite program u koji korisnik unosi koordinate x i y neke točke ravnine za koje mora vrijediti $x \neq 0$, $y \neq 0$, $|x| < 100$ i $|y| < 100$. U slučaju krivog unosa, program prekida izvođenje, a inače se ispisuje u kojem se kvadrantu koordinatnog sustava unesena točka nalazi.

UPUTE: Točka se nalazi u 1. kvadrantu ukoliko je koordinata $x > 0$ i koordinata $y > 0$, točka se nalazi u 2. kvadrantu ukoliko je koordinata $x < 0$ i koordinata $y > 0$, točka se nalazi u 3. kvadrantu ukoliko je koordinata $x < 0$ i koordinata $y < 0$, točka se nalazi u 4. kvadrantu ukoliko je koordinata $x > 0$ i koordinata $y < 0$.

25. Napišite program u koji korisnik unosi vrijednost radijusa kruga i koordinate središta kruga. Ako je učitana vrijednost radijusa manja ili jednaka nuli, program ispisuje poruku «Radijus mora biti veći od 0» i završava se njegovo izvođenje. Ako je učitana vrijednost radijusa ispravna, program ispisuje vrijednost površine kruga. Nakon toga program provjerava nalazi li se ishodište unutar kruga (provjerom je li udaljenost središta od ishodišta veća ili jednaka radijusu). Ako se ishodište nalazi unutar kruga, ispisuje se poruka «Krug sadrži točku ishodišta», odnosno «Krug ne sadrži točku ishodišta».

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

26. Napišite program u koji korisnik unosi 3 cijela broja. Brojevi moraju biti takvi da njihova suma ne bude manja od 400 i da je barem jedan troznamenkasti broj. Program provjerava ulazne podatke i, u slučaju da brojevi ne zadovoljavaju zadane kriterije, ispisuje odgovarajuću poruku. Potom se pronalazi i ispisuje koji je od učitanih brojeva najmanji.

1. kolokvij

1. Napišite program koji korisniku ispisuje sljedeći izbornik:

1. Pokeni interakciju
2. Ispiši pozdravnu poruku
3. Prekini izvođenje programa

Ukoliko korisnik odabere opciju 1, od njega se traži unos peteroznamenastog broja. Nije potrebno provjeravati je li korisnik upisao ispravan broj. Vrijednost logičke varijable je istinita ako su prva i posljednja znamenka parni brojevi. Ukoliko je vrijednost logičke varijable istinita, ispisuje se poruka «Prva i posljednja znamenka su parni brojevi.». Ukoliko korisnik odabere opciju 2, ispisuje se poruka «Dobrodošli u program!». U programu je potrebno demonstrirati naredbe za grananje *if* i *switch*.

2. Napišite program u koji korisnik unosi cijeli broj. Potom se ispisuje sljedeći izbornik:

- (a) Ispiši broj
- (b) Ispiši apsolutnu vrijednost broja
- (c) Prekini izvođenje programa

Ukoliko korisnik odabere opciju a, ispisuje se uneseni broj. Ukoliko korisnik odabere opciju b, najprije se izračunava vrijednost logičke varijable koja je istinita ako je apsolutna vrijednost broja jednaka unesenom broju. Ako je logička varijabla istinita, ispisuje se poruka «Uneseni broj je pozitivan.», a ako nije, ispisuje se apsolutna vrijednost broja. U programu je potrebno demonstrirati naredbe za grananje *if* i *switch*.

3. Napišite program koji korisniku ispisuje sljedeći izbornik (koristiti *switch*):

1. Pokeni igru
2. Unesi broj
3. Prekini igru

Ukoliko korisnik odabere opciju 1, ispisuje se poruka «Da vidimo kako te idu igre na sreću!». Ukoliko korisnik odabere opciju 2, izračunava se vrijednost logičke varijable koja je istinita ako uneseni broj nije u rasponu [20, 80]. Brojevi 20 i 80 su definirani kao konstante. U slučaju da je logička varijabla neistinita, ispisuje se poruka «Ni blizu!» (koristiti *if*).

4. Napišite program koji korisniku ispisuje sljedeći izbornik (koristiti *switch*):

1. Unesi brojeve
2. Unesi slova

3. Prekini izvođenje programa

Ukoliko korisnik odabere opciju 1, program traži unos 3 realna broja. Vrijednost logičke varijable je istinita ako su uneseni brojevi u rastućem poretku te unutar raspona $[1, 100]$. Ukoliko je logička varijabla istinita, ispisuje se poruka «Brojevi zadovoljavaju uvjet!» (koristiti *if*). Ukoliko korisnik odabere opciju 2, program traži unos 3 slova. Izračunava se i ispisuje vrijednost logičke varijable koja je istinita ako su unesena slova u rastućem poretku.

5. Napišite program u koji se unosi datum, mjesec i godina rođenja (korisnika treba porukama upozoriti kada unosi koju od vrijednosti). Logička varijabla ima vrijednost *true* ako osoba može u mirovinu. Osoba ima pravo na mirovinu ako je rođena prije 17.11.1946. godine. Potom se ispisuje sljedeći izbornik:

- 1a. Uvjet za mirovinu
- 2b. Godine do mirovine
- 3c. Prekini izvođenje programa

Ukoliko korisnik odabere opciju 1 ili a (treba omogućiti unos broja i slova), ispisuje se poruka «Osoba ima uvjet za mirovinu!» ili «Osoba nema uvjet za mirovinu!». Ukoliko korisnik odabere opciju 2 ili b, ispisuje se broj godina do stjecanja uvjeta za mirovinu. Ako će osoba steći uvjet ove godine, ispisuje se poruka «U mirovinu možete ove godine!». Ako osoba već ima uvjet za mirovinu, ispisuje se poruka «Zašto vi još uvijek radite?!». U programu je potrebno demonstrirati naredbe za grananje *if* i *switch*.

5 Uporaba petlji: *while*, *do-while*, *for*

1. Napišite program koji ispisuje brojeve od 1 do 20.
2. Napišite program koji od korisnika traži unos broja u intervalu [1,10]. Unos se ponavlja dok korisnik ne unese odgovarajuću vrijednost. Nakon toga ispisuju se vrijednosti faktoriijela svih brojeva od 1 do broja kojeg je unio korisnik.
3. Modificirajte kôd prethodnog zadatka tako da koristite *while* petlju.
4. Ispravite logičku pogrešku u danom kôdu. Program od korisnika traži unos niza cijelih brojeva. Korisnik unosi brojeve sve dok zbroj brojeva ne prijeđe 20. Izračunava se i ispisuje koliko brojeva je upisano.

```
#include <iostream>
using namespace std;

int main() {
    int x, zbroj, brojac;
    cout << "Ucitaj niz brojeva: \n";
    while (zbroj != 20) {
        cin >> x;
        zbroj += x;
        brojac++;
    }
    cout << "Broj ucitanih brojeva je " << brojac << endl;
    cin.ignore();
    cin.get();
    return 0;
}
```

5. Napišite program u koji korisnik unosi broj x , a ispisuje se prvih x potencija broja 2, počevši od nulte potencije.
6. Napišite program koji će za učitani prirodni broj n (ukoliko korisnik unese $n \leq 0$ ili $n > 100$, traži se ponovni unos) ispisati prvo kvadrato svih brojeva od 1 do n , a potom drugi korijen svih brojeva od 1 do n .
7. Napišite program koji za učitani cijeli broj n (n mora biti iz segmenta [1,20], ako nije, prekida se izvođenje programa) ispisuje n parnih brojeva počevši od broja 2.
8. Napišite program u koji korisnik unosi broj x , a ispisuju se sve potencije broja 2 manje od broja x .

9. Napišite program koji ispisuje englesku abecedu.
10. Modificirajte kôd prethodnog zadatka tako da se ispisuju dva stupca: u prvom stupcu neka je ASCII kod (od 65 do 90 – velika slova, odnosno 97 do 122 mala slova), a u drugom slova abecede.
11. Napišite program u koji korisnik unosi broj x , a ispisuje se svaki treći broj od x do 1.
12. Napišite program u koji korisnik unosi najprije broj x , a zatim i x brojeva. Ispisuje se koliko brojeva je parno.
13. Napišite program u koji korisnik unosi najprije broj x , a zatim i x ocjena. Ispisuje se prosjek ocjena.
14. Modificirajte kôd prethodnog zadatka tako da se prikazuje i najmanja od x unesenih ocjena.
15. Napišite program u koji korisnik unosi najprije broj x , a zatim i x brojeva. Ispisuje se najveći broj i najmanji broj.
16. Napišite program u koji korisnik unosi brojeve a , b , x i y , a ispisuju se svi brojevi iz intervala $[a, b]$ koji nisu djeljivi niti s x niti s y .
17. Napišite program u koji korisnik unosi vrijednost n , a ispisuju se sve vrijednosti u intervalu $[1, n]$ i njihov zbroj, s time da je svaka vrijednost duplo veća od prethodne.
- Test podaci:** (65) -> (1, 2, 4, 8, 16, 32, 64) -> suma=127
18. Napišite program u koji korisnik unosi broj, a ispisuje se je li uneseni prirodni broj prost (prim). Broj 1 nije prost.
19. Napišite program za izračunavanje kamate na štednju tijekom 10 godina. Broj godina definirati kao konstantu. Početni iznos je 1000\$, a kamata na štednju iznosi 5%. Kamatni postotak definirati također kao konstantu. Program izračunava i ispisuje iznos na računu na kraju svake godine u narednih 10 godina. Kamatni iznos poslije svake godine ostaje na računu.
20. Napišite program koji izračunava koliko godina je potrebno kako bi iznos na Matinom štednom računu bio veći od iznosa na Aninom računu i ispisuje oba iznosa nakon n godina. Ana je uložila \$100 na štedni račun s jednostavnom kamatnom stopom od 10%. Svake godine kamata iznosi 10% na inicijalni iznos, odnosno \$10. Mate je uložio \$100 na štedni račun sa složenom kamatnom stopom od 5%. Svake godine na račun se dodaje kamata od 5% na trenutni iznos.

Kamatne stope definirati kao konstante.

21. Napišite program u koji korisnik unosi znakove (@ je oznaka za kraj), a program ispisuje je li riječ o znamenki, slovu, razmaku, interpunkcijskoj oznaci ili nekom drugom znaku.

22. Napišite program u koji korisnik unosi rečenicu (točka je oznaka za kraj), a ispisuje se broj praznina te ukupan broj znakova u rečenici (uključujući i oznaku za kraj).

23. Napišite program koji za niz znakova (točka je oznaka za kraj) koji unosi korisnik provjerava pojavljuje li se u nizu bar pet samoglasnika. Program ispisuje «U nizu se pojavljuje 5 ili više samoglasnika.» ili «U nizu se ne pojavljuje 5 ili više samoglasnika.».

24. Napišite program u koji korisnik unosi znakove sve dok se ne učita veliko slovo. Program ispisuje koliko je malih slova upisano.

25.♦ Napišite program u koji korisnik unosi niz cijelih brojeva s oznakom kraja 0, a program provjerava je li svaki treći član niza jednak prvom članu, te ispisuje poruku «Niz ima svojstvo.» ili «Niz nema svojstvo.».

26. Napišite program koji će tražiti od korisnika unos cijelih brojeva sve dok se ne učita 0. Program provjerava ima li niz svojstvo da mu je svaki idući broj 2 puta manji od prethodnog. Ispisuje se ima li niz svojstvo ili ne.

27. Napišite program koji će tražiti od korisnika unos cijelih brojeva sve dok se ne učita 0. Program provjerava ima li niz svojstvo da mu je (barem) svaki drugi element paran broj (počevši od prvog elementa). Ispisuje se ima li niz svojstvo ili ne.

28. Napišite program koji će tražiti od korisnika unos realnih brojeva sve dok se ne učita negativan broj. Program provjerava ima li niz svojstvo da su mu svi elementi iz segmenta [1,100]. Ispisuje se ima li niz svojstvo ili ne.

29. Napišite program koji će tražiti od korisnika unos prirodnih brojeva sve dok se ne učita nula. Program provjerava pojavljuju li se u nizu barem tri broja sa svojstvom da su djeljivi s 2, a nisu djeljivi s 5. U slučaju da niz ima svojstvo, ispisuje se poruka «Niz ima traženo svojstvo.».

30. Napišite program koji će tražiti od korisnika unos niza prirodnih brojeva s oznakom kraja niza 0. Program provjerava pojavljuju li se u nizu bar tri broja sa svojstvom da je broj djeljiv s 2, a nije djeljiv sa 4. Potvrda traženog svojstva niza uzrokuje izlaz iz petlje prije učitavanja oznake za kraj niza. U tom slučaju, ispisuje se poruka «Niz ima traženo svojstvo.».

31. Napišite program koji će tražiti od korisnika unos cijelih brojeva sve dok se ne učitava broj djeljiv s 5. Određuje se i ispisuje najmanji broj u nizu te broj svih negativnih brojeva (zadnji učitani broj se kao oznaka za kraj niza ne računa).

32. Napišite program koji od korisnika traži unos realnih brojeva sve dok se ne učitava 0. Ispisuje se koliko je ukupno brojeva korisnik upisao, te koliko je parnih, a koliko neparnih brojeva.

33. Napišite program koji za prirodne brojeve koje unosi korisnik (oznaka kraja niza je negativna vrijednost ili 0) izračunava sumu onih koji zadovoljavaju svojstvo da su djeljivi sa 7 ili su istovremeno djeljivi s 5, ali ne i s 3.

Test podaci: (3, 5, 7, 15, 20, 21, 25, 30, 42, 49 -8) \rightarrow suma=151

34. Napišite program koji će od korisnika tražiti unos niza cijelih brojeva. Korisnik unosi brojeve sve dok suma brojeva ne postane veća od 50. Ispisuje se koliko je ukupno brojeva korisnik upisao.

35. Napišite program koji traži od korisnika unos niza prirodnih brojeva. Korisnik unosi brojeve sve dok se ne učitaju tri jedinice za redom. Unos prestaje i ukoliko korisnik unese više od 20 brojeva ili ako uneseni broj nije prirodan. Ispisuje se suma upisanih brojeva.

36. Napišite program u koji korisnik učitava prirodne brojeve sve dok ne bude narušeno svojstvo da je svaki idući element niza strogo veći od prethodnog elementa. Na kraju se ispisuje umnožak svih učitanih dvoznamenkastih brojeva.

37. Napišite program koji za učitani cijeli broj n (n mora biti iz segmenta $[1,1000]$, a ako nije, prekida se izvođenje programa) računa i ispisuje aritmetičku sredinu svih neparnih brojeva iz segmenta $[1,n]$.

38. Napišite program koji za učitani cijeli broj n (n mora biti iz segmenta $[1,1000]$, a ako nije, prekida se izvođenje programa) računa te ispisuje sumu i aritmetičku sredinu svih parnih brojeva iz segmenta $[1,n]$.

39. Napišite program koji od korisnika traži unos cijelog broja x (x mora biti iz segmenta $[1,30]$, a ako nije, prekida se izvođenje programa uz poruku korisniku da nije upisao ispravan broj). Nakon toga od korisnika se traži unos x realnih brojeva. Program ispisuje sumu i aritmetičku sredinu učitanih brojeva.

40. Napišite program koji će za učitani prirodni broj n (ukoliko korisnik unese $n \leq 0$ ili $n > 50$, prekida se izvođenje programa) ispisati sve prirodne brojeve manje od n koji su djeljivi brojem 7.

41. Napišite program koji od korisnika traži unos cijelog broja x (x mora biti iz segmenta $[1,30]$, a ako nije, prekida se izvođenje programa uz poruku korisniku da nije upisao ispravan broj). Nakon toga od korisnika se traži unos x realnih brojeva. Program ispisuje koliko učitanih brojeva ne pripada segmentu $[20,80]$.

42. Napišite program koji od korisnika traži unos cijelog broja x (x mora biti iz segmenta $[1,50]$, a ako nije, prekida se izvođenje programa uz poruku korisniku da nije upisao ispravan broj). Nakon toga od korisnika se traži unos x cijelih brojeva. Program ispisuje sumu učitanih brojeva koji su djeljivi brojem 2.

43.♦ Napišite program u koji korisnik unosi prirodni broj, a ispisuje se suma znamenki unesenog prirodnog broja.

44. Napišite program u koji se unosi peteroznamenasti broj, a ispisuje se je li riječ o palindromu ili nije. Palindrom je broj ili string koji se čita jednako s obje strane.

45. Napišite program u koji korisnik unosi prirodni broj, a ispisuje se suma tog broja i broja koji dobijemo kada obrnemo redoslijed znamenki unesenog broja.

46. Napišite program u koji korisnik unosi dva prirodna broja, a ispisuje se njihova najveća zajednička mjera.

UPUTA: Koristiti sljedeću definiciju $NZM(x,y)$:

- za $x=y$ $NZM(x,y) = x$
- za $x<y$ $NZM(x,y) = NZM(y-x,x)$
- za $x>y$ $NZM(x,y) = NZM(x-y,y)$

npr: $NZM(10,6)=NZM(4,6)=NZM(2,4)=NZM(2,2)=2$

47.♦ Napišite program u kojem se generira slučajni broj u rasponu $[1,100]$. Potom korisnik unosi broj iz navedenog raspona, a program ispisuje je li uneseni broj veći, manji ili jednak generiranom pseudo-slučajnom broju. Izvođenje programa se završava kada korisnik pogodi slučajni broj.

48. Napišite program za simulaciju bacanja dviju kockica. Brojevi na koje se kockice okrenu su pseudoslučajni brojevi. Ako je suma 7 ili 11, ispisuje se pobjeda, ako je 2, 3 ili 12, ispisuje se poraz, a u ostalim slučajevima se ponovno baca. Ponovno se baca dok se ne dobije suma 7, kada se ispisuje poraz, ili kombinacija s prvog bacanja, kada se ispisuje pobjeda.

6 Ugnježdene petlje

1. • Napišite program koji će ispisati tablicu množenja od 1 do 10.

2. • Napišite program koji od korisnika traži učitavanje prirodnog broja n za koji mora vrijediti $2 < n < 10$. Ako korisnik unese broj koji nije iz intervala, prekida se izvođenje programa, a ako je upisao dobar broj na ekranu se treba ispisati kvadratna matrica s n redaka i n stupaca tako da su na glavnoj dijagonali ispisane jedinice (odnosno niz 1, -1, 1, -1, ...), iznad glavne dijagonale ispisuje se slovo x, a na ostalim pozicijama ispisuju se nule.

Test podaci: $n = 5$

1	x	x	x	x
0	-1	x	x	x
0	0	1	x	x
0	0	0	-1	x
0	0	0	0	1

3. Napišite program koji od korisnika traži učitavanje prirodnog broja n za koji mora vrijediti $2 < n < 10$. Ako korisnik unese broj koji nije iz intervala, prekida se izvođenje programa, a ako je upisao dobar broj na ekranu se treba ispisati kvadratna matrica s n redaka i n stupaca tako da se na glavnoj dijagonali ispisuju brojevi od n do 1, a na ostalim pozicijama zvjezdice.

Test podaci: $n = 5$

1	*	*	*	*
*	2	*	*	*
*	*	3	*	*
*	*	*	4	*
*	*	*	*	5

4. Napišite program koji od korisnika traži učitavanje tri prirodna broja, m , n i x , za koja mora vrijediti $2 < m < 10$ i $2 < n < 10$. Ako korisnik unese broj koji nije iz intervala, prekida se izvođenje programa, a ako je upisao dobre brojeve, na ekranu se treba ispisati matrica s m redaka i n stupaca čiji su elementi brojevi koji počinju od broja x i povećavaju se za 1.

Test podaci: $m = 5$, $n = 3$, $x = 10$

10	11	12
13	14	15
16	17	18
19	20	21
22	23	24

5. Napišite program koji od korisnika traži učitavanje prirodnog broja n za koji mora vrijediti $2 < n < 10$. Ako korisnik unese broj koji nije iz intervala, prekida se izvođenje programa, a ako je upisao dobar broj na ekranu se treba ispisati dio kvadratne matrice (iznad sporedne dijagonale) s n redaka i n stupaca čiji su elementi *.

Test podaci: $n = 5$

```
*   *   *   *
*   *   *
*   *
*
```

6. Napišite program koji od korisnika traži učitavanje prirodnog broja n za koji mora vrijediti $2 < n < 10$. Ako korisnik unese broj koji nije iz intervala, prekida se izvođenje programa, a ako je upisao dobar broj na ekranu se treba ispisati dio kvadratne matrice (iznad glavne dijagonale) s n redaka i n stupaca čiji su elementi brojevi kao što je prikazano u primjeru.

Test podaci: $n = 5$

```
1  2  3  4  5
   2  3  4  5
      3  4  5
         4  5
            5
```

7. Napišite program koji od korisnika traži učitavanje prirodnog broja n za koji mora vrijediti $2 < n < 10$. Ako korisnik unese broj koji nije iz intervala, prekida se izvođenje programa, a ako je upisao dobar broj, na ekranu se treba ispisati dio kvadratne matrice (ispod glavne dijagonale) s n redaka i n stupaca čiji su elementi brojevi kao što je prikazano u primjeru.

Test podaci: $n = 5$

```
1
1  2
1  2  3
1  2  3  4
1  2  3  4  5
```

8. Napišite program koji od korisnika traži učitavanje prirodnog broja n za koji mora vrijediti $2 < n < 10$. Ako korisnik unese broj koji nije iz intervala, prekida se izvođenje programa, a ako je upisao dobar broj na ekranu se treba ispisati dio kvadratne matrice (iznad sporedne dijagonale) s n redaka i n stupaca čiji su elementi brojevi kao što je prikazano u primjeru.

Test podaci: $n = 5$

```

1  2  3  4  5
2  3  4  5
3  4  5
4  5
5

```

9. Napišite program u koji korisnik unosi broj redaka/stupaca. Unos se ponavlja sve dok uneseni broj ne bude paran broj u intervalu $[2, 10]$. Nakon toga, ispisuje se kvadratna matrica koja na glavnoj dijagonali ima zvjezdice, na sporednoj nule, a na ostalim pozicijama točke.

Test podaci: $n = 5$

```

*  .  .  .  0
.  *  .  0  .
.  .  *  .  .
.  0  .  *  .
0  .  .  .  *

```

10. Napišite program koji od korisnika traži unos realne vrijednosti x . Program računa i ispisuje vrijednost funkcije $f(x)$. Funkcija je definirana kako slijedi:

- $f(x) = x * 10$ ako je x element segmenta $[1, 20]$
- $f(x) = x + 2$ ako je x element segmenta $[21, 40]$
- $f(x) = x/3$ ako je x element segmenta $[41, 60]$
- $f(x) = 0$ za sve ostale vrijednosti x

Program se ponavlja sve dok korisnik na upit želi li završiti program ne odgovori slovom d (neovisno je li uneseno veliko ili malo slovo). Prije izlaska iz programa, ispisuje se koliko puta se izvršila koja od aritmetičkih operacija. Na kraju se prikazuje i kvadratna matrica čije dimenzije ovise o ukupnom broju izvođenja izračuna. Na svim pozicijama u matrici prikazuje se znak *.

11. Napišite program koji od korisnika traži unos realne vrijednosti x . Program računa i ispisuje vrijednost funkcije $f(x)$. Funkcija je definirana kako slijedi:

- $f(x) = x * 5$ ako je x element segmenta $[1, 100]$
- $f(x) = x + 5$ ako je x element segmenta $[-100, -1]$
- $f(x) = x/5$ ako je x element segmenta $[150, 200]$
- $f(x) = 0$ za sve ostale vrijednosti x

Program se ponavlja sve dok korisnik na upit želi li završiti program odgovara sa slovom n (neovisno je li uneseno veliko ili malo slovo). Prije izlaska iz programa prikazuje se kvadratna matrica čije dimenzije čini prosječna cjelobrojna

vrijednost svih vrijednosti funkcije. Na svim pozicijama u matrici prikazuje se suma.

12. Napišite program koji od korisnika traži unos cjelobrojnih vrijednosti a i b . Program računa i ispisuje vrijednost c koju dobiva na sljedeći način:

- $c = a + b$ ako su a i b neparni brojevi koji pripadaju segmentu $[0, 50]$
- $c = a - b$ ako su a i b neparni brojevi koji ne pripadaju segmentu $[10, 100]$
- $y = 0$ inače

Program se ponavlja sve dok se ne učita 0. Prije izlaska iz programa, korisnik odgovara na upit želi li prikaz broja izvršenja svake od aritmetičkih operacija. Ukoliko unese slovo d (neovisno je li veliko ili malo slovo), prikazuje se kvadratna matrica čije dimenzije ovise o ukupnom broju izvođenja izračuna. Najprije se prikazuje onoliko redaka koliko puta se izvršila aritmetička operacija zbrajanja, a na svim pozicijama prikazuje se znak $+$, a zatim onoliko redaka koliko puta se izvršila aritmetička operacija oduzimanja, a na svim pozicijama prikazuje se znak $-$.

13. Napišite program za rješavanje linearne jednadžbe $a * x + b = 0$. Brojevi a i b su realni brojevi. Posebno razmatrati slučajeve:

- $a = 0, b = 0$ Svaki broj je rješenje linearne jednadžbe.
- $a = 0, b \neq 0$ Linearne jednadžbe nema rješenja.
- $a \neq 0, b \neq 0$ $x = -b/a$

Test podaci (a, b, x): $(1, -3, 3)$, $(-2, 5, 2.5)$, $(0, 0, \text{svaki broj})$, $(0, 10, \text{nema rješenja})$.

Program se ponavlja sve dok korisnik na upit želi li nastaviti izvođenje programa odgovara slovom d ili y (neovisno je li uneseno veliko ili malo slovo). Prije izlaska iz programa prikazuje se matrica u kojoj broj redaka predstavlja slučajeve bez rješenja, a broj stupaca slučajeve s rješenjem. Na svim pozicijama u matrici prikazuje se znak @.

14. Napišite program koji od korisnika traži unos cjelobrojne vrijednosti x . Program računa i ispisuje vrijednost y koju dobiva na sljedeći način:

- $y = x * 2$ ako je x neparan i djeljiv brojem 3
- $y = x/2$ ako je x paran broj i ne pripada segmentu $[10, 100]$

- $y = 0$ inače

Program se ponavlja sve dok se ne učita 0. Prije izlaska iz programa, korisnik odgovara na upit želi li odmah prekinuti izvođenje programa. Ukoliko korisnik unese slovo n (neovisno je li veliko ili malo slovo), prikazuju se dvije kvadratne matrice. Prva matrica ima dimenzije određene brojem izvršenja operacije množenja, a na svim pozicijama nalazi se znak $*$, a druga ima dimenzije određene brojem izvršenja operacije dijeljenja i na svim pozicijama nalazi se znak $/$.

15. Napišite program u koji korisnik unosi cijeli broj, a program računa sumu znamenki unesenog broja. Ukoliko je suma znamenki jednoznamenkast broj, suma se ispisuje, a inače se zbrajaju znamenke sume sve dok se suma ne svede na jednoznamenkasti broj.

16. Napišite program u koji korisnik unosi pozitivan cijeli broj, a ispisuju se svi savršeni brojevi od broja 2 do unesenog broja. Broj je savršen ako je jednak sumi svih svojih faktora.

Test podaci:

$$\begin{array}{rcll} 6 & = & 1 & + 2 + 3 \\ 28 & = & 1 & + 2 + 4 + 7 + 14 \end{array}$$

17. Program pronalazi i ispisuje sve Pitagorine brojeve od 1 do 100. Za brojeve x , y i z kažemo da su Pitagorini ako je $x^2 + y^2 = z^2$. Obratiti pozornost na sve moguće permutacije.

7 Polja i nizovi

1. Napišite program koji od korisnika traži unos prirodnog broja u intervalu $(0, 20]$. Unos se ponavlja sve dok unesena vrijednost ne bude u odgovarajućem intervalu. Nakon toga unosi se niz od n cijelih brojeva, a izračunava se i ispisuje novi niz definiran na sljedeći način: $c_i = \max(a_i, b_i)$.
2. Napišite program koji od korisnika traži unos prirodnog broja u intervalu $(0, 30]$. Unos se ponavlja sve dok unesena vrijednost ne bude u odgovarajućem intervalu. Nakon toga unosi se niz od n cijelih brojeva. Učitani niz ispisuje se na zaslone. Pronalazi se i ispisuje najveći i najmanji broj u nizu.
3. Napišite program koji od korisnika traži unos prirodnog broja u intervalu $(0, 30]$. Unos se ponavlja sve dok unesena vrijednost ne bude u odgovarajućem intervalu. Nakon toga unosi se niz od n prirodnih brojeva. Učitani niz ispisuje se na zaslon. Izračunava se te ispisuje suma i broj parnih i neparnih brojeva odvojeno.
4. Napišite program koji od korisnika traži unos prirodnog broja u intervalu $(0, 20]$. Unos se ponavlja sve dok unesena vrijednost ne bude u odgovarajućem intervalu. Nakon toga unosi se niz od n cijelih brojeva. Učitani niz ispisuje se na zaslon. Formira se i ispisuje novi niz koji se sastoji od kvadrata vrijednosti početnog niza ($b_i = a_i^2$).
5. Napišite program koji korisnika traži unos prirodnog broja u intervalu $(0, 30]$. Unos se ponavlja sve dok unesena vrijednost ne bude u odgovarajućem intervalu. Nakon toga unosi se dva niza od n realnih brojeva. Izračunava se i ispisuje novi niz definiran na sljedeći način: $c_i = a_i^4 + b_i^3$.
6. Napišite program koji od korisnika traži unos prirodnog broja u intervalu $(0, 50]$. Unos se ponavlja sve dok unesena vrijednost ne bude u odgovarajućem intervalu. Nakon toga unosi se niz od n cijelih brojeva. Učitani niz ispisuje se na zaslon. Ispisuje se dva najveća broja u nizu.
7. Napišite program koji od korisnika traži unos prirodnog broja u intervalu $(0, 20]$. Unos se ponavlja sve dok unesena vrijednost ne bude u odgovarajućem intervalu. Nakon toga unosi se niz od n realnih brojeva (niz X). Učitani niz ispisuje se na zaslon. Ispisuje se novi niz (niz Y) definiran na sljedeći način: $y_1 = 1$; $y_n = 1$; $y_i = \frac{x_{i-1} + x_{i+1}}{2}$; $i = 2, 3, 4, \dots, n-1$.
8. Napišite program koji od korisnika traži unos prirodnog broja u intervalu $(0, 20]$. Unos se ponavlja sve dok unesena vrijednost ne bude u odgovarajućem intervalu. Nakon toga unosi se niz od n cijelih brojeva. Učitani niz ispisuje se na zaslon. Ispisuje se broj pozitivnih elemenata, broj negativnih elemenata i

broj nula.

9. Napišite program koji od korisnika traži unos prirodnog broja u intervalu $(0, 10]$. Unos se ponavlja sve dok unesena vrijednost ne bude u odgovarajućem intervalu. Nakon toga unose se dva niza od n realnih brojeva. Izračunava se i ispisuje novi niz definiran na sljedeći način: $c_i = \frac{a_i + b_i}{2}$.

10. Napišite program koji od korisnika traži unos prirodnog broja u intervalu $(0, 10]$. Unos se ponavlja sve dok unesena vrijednost ne bude u odgovarajućem intervalu. Nakon toga unosi se niz od n cijelih brojeva. Učitani niz ispisuje se na zaslone. Ispisuje se aritmetička sredina dvoznamenkastih brojeva.

11. Napišite program koji od korisnika traži unos prirodnog broja u intervalu $(0, 20]$. Unos se ponavlja sve dok unesena vrijednost ne bude u odgovarajućem intervalu. Nakon toga unosi se niz od n prirodnih brojeva. Učitani niz ispisuje se na zaslon. Ispisuju se sume recipročnih vrijednosti parnih i neparnih brojeva odvojeno, te broj parnih i broj neparnih brojeva.

12. Napišite program koji od korisnika traži unos prirodnog broja u intervalu $(0, 40]$. Unos se ponavlja sve dok unesena vrijednost ne bude u odgovarajućem intervalu. Nakon toga unosi se niz od n prirodnih brojeva. Učitani niz ispisuje se na zaslon. Izračunava se i ispisuje aritmetička sredina jednoznamenkastih brojeva niza.

13. Napišite program koji od korisnika traži unos prirodnog broja u intervalu $(0, 50]$. Unos se ponavlja sve dok unesena vrijednost ne bude u odgovarajućem intervalu. Nakon toga učitava se niz od n realnih brojeva. Učitani niz ispisuje se na zaslon. Ispisuju se indeksi svih onih elemenata koji ne pripadaju segmentu $[1, 50]$.

14.♦ Napišite program koji od korisnika traži unos prirodnog broja u intervalu $(0, 40]$. Unos se ponavlja sve dok unesena vrijednost ne bude u odgovarajućem intervalu. Nakon toga učitava se niz od n realnih brojeva. Učitani niz ispisuje se na zaslon. Ispisuje se novi niz koji najprije sadrži sve nule, zatim sve negativne elemente i na kraju pozitivne elemente zadanog niza.

15. Napišite program koji od korisnika traži unos prirodnog broja u intervalu $(0, 50]$. Unos se ponavlja sve dok unesena vrijednost ne bude u odgovarajućem intervalu. Nakon toga unosi se niz od n cijelih brojeva. Učitani niz ispisuje se na zaslon. Ispisuje se novi niz čiji su elementi jednaki zbroju znamenaka odgovarajućeg elementa zadanog niza.

16. Napišite program koji od korisnika traži unos dva prirodna broja, $0 < n \leq 80$ i $0 < m \leq 50$. Unos se ponavlja sve dok unesena vrijednost ne bude u odgovarajućem intervalu. Nakon toga unose se uređeni (sortirani) nizovi od n i m realnih brojeva. Ispisuju se učitani nizovi i novi niz sastavljen od elemenata

dva učitana niza tako da i taj novi niz bude uređen.

17. Napišite program koji od korisnika traži unos prirodnog broja u intervalu $(0, 50]$. Unos se ponavlja sve dok unesena vrijednost ne bude u odgovarajućem intervalu. Nakon toga unosi se niz od n cijelih brojeva. Učitani niz ispisuje se na zaslون. Ispisuju se indeksi svih onih elemenata koji su pozitivni i nalaze se na parnim pozicijama.

18. Napišite program koji od korisnika traži unos prirodnog broja u intervalu $(0, 100]$. Unos se ponavlja sve dok unesena vrijednost ne bude u odgovarajućem intervalu. Nakon toga se učitava niz od n prirodnih brojeva (ukoliko korisnik pokuša upisati broj koji nije prirodan, mora ponoviti unos za taj element!). Učitani niz ispisuje se na zaslون. Izračunava se i ispisuje aritmetička sredina svih parnih elemenata na parnim pozicijama.

19. Napišite program koji od korisnika traži unos prirodnog broja u intervalu $(0, 100]$. Unos se ponavlja sve dok unesena vrijednost ne bude u odgovarajućem intervalu. Nakon toga se učitava niz od n prirodnih brojeva (ukoliko korisnik pokuša upisati broj koji nije prirodan, mora ponoviti unos za taj element!). Učitani niz ispisuje se na zaslون. Izračunava se i ispisuje aritmetička sredina svih neparnih elemenata na neparnim pozicijama.

20. Matin robot se pokvario. On će ga pokušati popraviti tako da nasumično presjeca žice od kojih se robot sastoji, ali želi prekinuti samo one žice čiji je napon manji od prosječnog jer se boji visokog napona. Napišite program koji će Mati reći koliko žica smije prekinuti.

21. Za pohranu velikih cjelobrojnih vrijednosti može se koristiti polje (svaka znamenka čini jedan element polja). Napisati program koji traži unos dva pozitivna cijela broja koji imaju 20 ili manje znamenki i ispisuje zbroj ta dva broja. Program treba zbrajati pomoću algoritma za ručno zbrajanje. Zbroj se sprema u polje veličine 20, a rezultat se ispisuje na zaslonu. Ako rezultat ima više od 20 znamenki, ispisuje se poruka «Nedozvoljeno veliki broj!». Maksimalni broj znamenki definiran je pomoću konstante *MAX*. Interakcija s korisnikom ponavlja se sve dok korisnik to želi.

22. Napišite program u kojem će se generirati određeni broj slučajnih brojeva iz segmenta $[0,9]$. Korisnik unosi maksimalni broj slučajnih brojeva ($n \leq 10000$), a potom se generiraju brojevi. Program treba raditi tako da za svaki od brojeva od 0 do 9 zapamti koliko puta je generiran. Na zaslonu se ispisuju svi generirani brojevi i koliko puta se svaki od njih generirao.

23. Napišite program u koji korisnik unosi polje realnih brojeva, a izračunava se i ispisuje standardna devijacija.

$$s = \sqrt{\sum_{i=1}^n \frac{(m_i - \bar{m})^2}{n-1}}$$

24. Napišite program koji će od korisnika tražiti neka najprije učitati rečenicu kao niz od najviše 80 znakova. Ispisuje se učitana rečenica, a potom se ispisuje i učitana rečenica obrnutim redoslijedom.

25. Napišite program koji će od korisnika tražiti unos rečenice kao niza od maksimalno 80 znakova. Najprije se ispisuje učitana rečenica, a potom program provjerava je li upisana rečenica palindrom (palindrom je rečenica koja se jednako čita sprijeda i otraga) i ispisuje odgovarajuću poruku. Obratiti pozornost na mala i velika slova, tj. malo slovo 'a' ne razlikuje se od velikog slova 'A'.

26. Napišite program koji će od korisnika tražiti neka najprije učitati rečenicu od najviše 200 znakova. Ispisuje se učitana rečenica, a potom i rečenica u kojoj su suglasnici izostavljeni.

27. Napišite program koji će od korisnika tražiti neka najprije učitati rečenicu od najviše 200 znakova. Ispisuje se učitana rečenica, a potom i rečenica u kojoj su mala slova zamijenjena velikim i obratno.

28. Napišite program koji će od korisnika tražiti neka najprije učitati rečenicu od najviše 200 znakova. Ispisuje se učitana rečenica, a potom i rečenica u kojoj je svaki znak koji je znamenka zamijenjen sa zvjezdicom.

29. Napišite program koji će od korisnika tražiti neka najprije učitati rečenicu od najviše 200 znakova. Ispisuje se učitana rečenica, a potom i brojevi pojavljivanja svakog samoglasnika posebno.

30. Napišite program koji će od korisnika tražiti neka najprije učitati rečenicu od najviše 200 znakova. Ispisuje se učitana rečenica, a potom i broj pojavljivanja velikih slova, malih slova i znamenki.

31. Napišite program koji će od korisnika tražiti učitavanje dva znakovna niza (najviše 50 znakova). Kreira se novi znakovni niz koji predstavlja spoj dva učitana niza. Ispisuju se sva tri niza na ekranu te koji je od prva dva znakovna niza duži.

32. Napišite program koji od korisnika traži učitavanje tri prezimena (najviše 50 znakova). Ispisuju se prezimena sortirana po abecedi.

33. Napišite program koji će od korisnika tražiti neka najprije učitati rečenicu od najviše 200 znakova. Ispisuje se učitana rečenica, a potom i rečenica u kojoj

su izostavljeni svi znakovi koji nisu slova.

34. Napišite program koji će od korisnika tražiti neka najprije učitati rečenicu od najviše 200 znakova. Ispisuje se učitana rečenica, a potom i rečenica u kojoj su izostavljeni svi znakovi koji nisu znamenke.

35. Napišite program koji će od korisnika tražiti neka najprije učitati rečenicu od najviše 200 znakova. Ispisuje se učitana rečenica, a potom i broj pojavljivanja znamenki u rečenici.

36. Napišite program koji će od korisnika tražiti neka najprije učitati rečenicu od najviše 200 znakova. Ispisuje se učitana rečenica, a potom i rečenica u kojoj su izostavljeni razmaci.

8 Dvodimenzionalna polja

1. • Napišite program koji će korisniku omogućiti analizu podataka o kretanju temperatura u hrvatskim gradovima za izabranu godinu promatranja. Podaci o temperaturi su cjelobrojni i pohranjuju se, za izabranu godinu, u dvodimenzionalnu matricu s 5 redaka (za svaki izabrani grad) i 12 stupaca (za svaki izabrani mjesec u godini). Korisniku treba omogućiti unos 5 gradova koji se pohranjuju u znakovne nizove, te unos godine i 12 prosječnih temperatura za unesenu godinu za svaki uneseni grad. Nakon toga ispisuje se sljedeći izbornik:

1. Ispis prosječne temperature za odabrani mjesec
2. Ispis prosječne godišnje temperature za odabrani grad
3. Ispis temperature u odabranom mjesecu i gradu
4. Ispis prosječne temperature u Hrvatskoj za zadanu godinu
5. Kraj interakcije

Realizirati svaku od navedenih opcija. Interakcija s programom ponavlja se sve dok korisnik ne odabere kraj rada.

2. Napišite program koji dodjeljuje putnicima mjesta u zrakoplovu. Pretpostavite da zrakoplov ima mjesta označena na sljedeći način:

1	A B	C D
2	A B	C D
3	A B	C D
4	A B	C D
5	A B	C D
6	A B	C D
7	A B	C D

Program treba prikazati tlocrt sjedećih mjesta gdje *X* označava zauzeto mjesto. Nakon što se mjesta 1A, 2B i 4C rezerviraju, tlocrt treba izgledati:

1	X B	C D
2	A X	C D
3	A B	C D
4	A B	X D
5	A B	C D
6	A B	C D
7	A B	C D

Nakon prikaza slobodnih mjesta, korisnik unosi željeno mjesto, tj. red i broj sjedala, a zatim se prikazuje ažurirani tlocrt. Program se nastavlja sve dok se ne popuni sjedeći kapacitet zrakoplova ili dok korisnik ne signalizira kraj interakcije. Ukoliko korisnik upiše mjesto koje je rezervirano, program treba tražiti novi unos jer je mjesto zauzeto.

3. Napišite program za rad s cjelobrojnom dvodimenzionalnom matricom A s n redaka i m stupaca u intervalu $[1,10]$. Korisnik najprije unosi broj redaka te broj stupaca matrice (učitavati dok se ne učitava ispravno), a potom se unose elementi matrice koji su cijeli brojevi. Potrebno je realizirati sljedeće opcije:

- ispis unesenih podatak u matičnom obliku (ispis $n \times m$)
- ispis zbroja apsolutnih vrijednosti svih elemenata matrice
- ispis indeksa najvećeg elementa matrice (ili više indeksa ako je tih elemenata više)
- ispis indeksa najmanjeg elementa matrice (ili više indeksa ako je tih elemenata više)
- ispis indeksa zadanog elementa kojeg unosi korisnik (ili više indeksa ako je tih elemenata više)
- ispis svih elemenata matrice koji su djeljivi s 3, a nisu djeljivi s 2
- ispis svih elemenata matrice koji se nalaze na glavnoj dijagonali matrice

4. Poznata *memory* igra funkcionira tako da ima špil karata koji sadrži identične parove. Primjerice, ako špil ima 6 karata, dvije karte imaju oznaku 1, dvije oznaku 2, a dvije oznaku 3. Karte se rotiraju i postavljaju na stol licem okrenute prema dolje. Igrač odabire dvije karte koje se tad okrenu i, ako su identične, one ostaju okrenute licem prema gore. Ako one ne čine par, vraćaju se u originalni položaj – licem prema dolje. Igra se nastavlja dok se ne otkriju svi parovi. Napišite program koji omogućava igranje te igre. Upotrijebite 16 karata koje se slože tako da čine kvadratnu matricu 4×4 , a označene su brojevima od 1 do 8. Program treba omogućiti korisniku odabir karata pomoću x i y koordinata. Pretpostavimo da su karte složene na sljedeći način:

	1	2	3	4
1	8	*	*	*
2	*	*	*	*
3	*	8	*	*
4	*	*	*	*

Sve karte su licem prema dolje osim para označenog brojem 8 koje su locirane koordinatama (1, 1) i (3, 2). Kako biste sakrili karte koje su privremeno otkrivene, ispišite velik broj novih redaka. Nakon svakog pokušaja, slučajno se odabiru dvije karte i zamjenjuju.

5. Izradite igricu kružić-križić. Na početku se unose imena igrača. Igrica uzastopce traži odabir polja od igrača X i igrača O. Na zaslonu se prikazuje:

1	2	3
4	5	6
7	8	9

Igrači unose broj pozicije koju žele označiti. Nakon svakog poteza, prikazuje se izmijenjeno polje, npr.:

X	X	O
4	5	6
O	8	9

Na kraju se ispiše pobjednik (1. ili 2. igrač) te ponudi ponovno pokretanje igre. Program možete nadograditi po želji.

6. Mali Mate našao se u šumi u kojoj se navodno nalazi blago. Mate je našao upute koje mu govore kako doći do blaga. Upute se nalaze u obliku niza znakova 'U', 'R', 'L', 'D' ('U' - gore, 'R' - desno, 'L' - lijevo, 'D' - dolje), koji neće sadržavati više od 100000 znakova. Mate je u šumu krenuo u novim cipelama. Šuma se sastoji od $n \times n$ kvadratića (čije su koordinate označene brojevima od 1 do n s lijeva na desno, odozgo prema dolje) koji Mati troše cipele čim on ugazi na njih. Broj redaka i stupaca n , kao i vrijednosti koliko koji kvadratić troši cipele unosi korisnik. Mate je vama dao upute i svoju početnu poziciju u šumi (X , Y) te kakvoću K svojih cipela na početku. Mate ne može po šumi hodati bez cipela, stoga zahtijeva od vas da mu kažete hoće li uspjeti doći do blaga. Šuma je okružena minskim poljem. Ako Mate izađe izvan šume ili ostane bez cipela, odustaje od daljnje potrage za blagom.

Test podaci:

ulaz			ulaz			ulaz		
3			3			3		
1	4	2	1	4	2	1	4	2
8	2	7	8	2	7	8	2	7
2	9	9	2	9	9	2	9	9
1	1	10	1	1	10	1	1	100
DRRL			DLL			DDRULR		
izlaz			izlaz			izlaz		
NE			NE			DA		
2 2			0 2			2 2		

7. Matina najdraža igračka je magični kvadrat u kojem se nalazi 9 prirodnih brojeva poredanih u 3 retka i 3 stupca. Magični kvadrat ima svojstvo da su zbrojevi brojeva u svakom retku, svakom stupcu i na obje dijagonale međusobno jednaki, a svaki broj u njemu je manji od ili jednak 20000. Dok je Mate spavao, lopovi su ukrali najviše tri broja iz njegovog magičnog kvadrata. Na mjestima brojeva koji su ukradeni nalazi se broj 0. Napišite program koji će za neki zadani (krnji) kvadrat odrediti brojeve koji nedostaju. Program ispisuje točno popunjeni magični kvadrat.

Test podaci:

ulaz			ulaz			ulaz		
4	9	2	0	12	12	495	468	0
3	0	7	16	10	0	0	522	414
8	1	6	8	8	14	441	0	549
izlaz			izlaz			izlaz		
4	9	2	6	12	12	495	468	603
3	5	7	16	10	4	630	522	414
8	1	6	8	8	14	441	576	549

9 Strukture

1. Napišite program koji će evidentirati podatke o dvjema osobama. U programu je potrebno definirati strukturu *osoba* koja će sadržavati sljedeće podatke: ime, prezime i osobni dohodak. Korisnik unosi podatke o osobama, a ispisuje se prosjek njihovih primanja.

2. • Napišite program koji će evidentirati podatke o studentu. U programu je potrebno definirati strukturu *student* koja će sadržavati sljedeće podatke: ime studenta, prezime studenta, broj indeksa i ocjenu. Korisnik unosi podatke o dva studenta, a ispisuje se prezime onog studenta koji ima bolju ocjenu. Ukoliko oba studenta imaju jednaku ocjenu, ispisuje se poruka «Studenti imaju jednaku ocjenu».

3. • Modificirati prethodni zadatak tako da se unosi niz od n studenata - formira se lista kao polje (elementi polja su tipa strukture), podaci za svih n studenata, a ispisuju se svi podaci za sve studente čija je ocjena jednaka ocjeni koju unese korisnik.

4. • Napišite program u kojem je definirana struktura *account* koja se sastoji od imena, prezimena i lozinke (voditi računa ima li korisnik 2 imena ili prezimena). Nakon unosa imena i prezimena generira se lozinka tako da se uzme prvo slovo imena i čitavo prezime, s tim da se svako pojavljivanje slova 'o' zamijeni s 0, a svako pojavljivanje slova 'i' s 1. Generirana lozinka se ispisuje.

Test podaci:

ULAZ
MARIJA MAJA
BRKIC

IZLAZ
mbrk1c

5. Napišite program u kojem je definirana struktura *box* koja sadrži ime proizvođača, visinu, širinu, duljinu i volumen kutije. Korisnik unosi ime, visinu, širinu i duljinu jedne kutije, a računa se i ispisuje volumen te kutije koji je jednak produktu preostale tri dimenzije.

6. Napišite program koji ocjenjuje studente na temelju sljedećih podataka:
- 2 su kviza, svaki ima 10 bodova
 - 1 ispit je na polugodištu, a drugi na kraju godine, svaki ima 100 bodova
 - završni ispit ima 50% udjela u ocjeni, ispit na polugodištu ima 25% udjela, a kvizovi zajedno imaju 25% udjela

90% ili više bodova je za izvrstan, [80% - 90%) vrlo dobar, [70% - 80%) je dobar, [60% - 70%) dovoljan, a sve ispod 60% je pad. Program učitava bodove studenata i ispisuje njihove ocjene te postotak postignutih bodova. Definirajte strukturu *student* koja se sastoji od prezimena studenta i polja s bodovima. Broj bodova u polju čine redom kviz1, kviz2, ispit na polugodištu i finalni ispit.

7. Napišite program koji će omogućiti evidenciju plaća za n radnika. Formira se lista kao polje (elementi polja su tipa strukture) za radnika sa sljedećim podacima: šifra radnika (integer), prezime (niz duljine 100), iznos plaće (iznos zaokružen na 2 decimale). Korisnik najprije unosi broj n ($0 < n < 100$) – za koliko osoba unosi podatke, a potom sve podatke za n osoba. Nakon toga na ekranu se nudi sljedeći izbornik:

1. pregled svih podataka za radnike – šifra, prezime te iznosi plaća
2. ispis svih podataka za radnike čije prezime počinje nekim zadanim slovom (slovo unosi korisnik)
3. ispis prezimena radnika čiji su iznosi plaća veći od nekog zadanog iznosa (iznos unosi korisnik)

Interakcija s programom završava kada korisnik odabere kraj.

8. Napišite program koji će omogućiti evidenciju podataka za n studenata. Formira se lista kao polje (elementi polja su tipa strukture) za studenta sa sljedećim podacima: broj indeksa (integer), ime (niz duljine 100), prezime (niz duljine 100), ocjena (integer 1,2,3,4 ili 5). Korisnik najprije unosi broj n ($0 < n < 50$) – za koliko studenata unosi podatke, a potom sve podatke za n studenata. Nakon toga na ekranu se ispisuje sljedeći izbornik:

1. pregled svih podataka za studente – broj indeksa, ime, prezime, ocjena
2. ispis svih podataka za studente čije prezime počinje nekim zadanim slovom (slovo unosi korisnik)
3. ispis prezimena studenata čija ocjena je manja od zadane ocjene (ocjenu unosi korisnik)

Interakcija s programom završava kada korisnik odabere kraj.

9. Napišite program koji će omogućiti evidenciju podataka za n osoba. Formira se lista kao polje (elementi polja su tipa strukture) za osobu sa sljedećim podacima: (redni) broj osobe, prezime i godina rođenja. Korisnik najprije unosi broj n ($0 < n < 70$) – za koliko osoba unosi podatke, a potom sve podatke za n osoba. Nakon toga na ekranu se ispisuje sljedeći izbornik:

1. pregled svih podataka za osobe – broj osobe, prezime, godina rođenja
2. ispis prezimena osoba čija je starost u intervalu [DG,GG], (podatke DG i GG unosi korisnik)

3. ispis svih podataka za osobe čije prezime počinje nekim zadanim slovom (slovo unosi korisnik)

Interakcija s programom završava kada korisnik odabere kraj.

10. Napišite program koji će omogućiti evidenciju podataka za n korisnika bankovnog računa. Formira se lista kao polje (elementi polja su tipa strukture) za korisnika bankovnog računa sa sljedećim podacima: broj_racuna (integer), prezime (niz duljine 90), stanje_racuna (iznos zaokružen na 2 decimale). Korisnik najprije unosi broj n ($0 < n < 99$) – za koliko korisnika unosi podatke, a potom sve podatke za n korisnika. Nakon toga na ekranu se ispisuje sljedeći izbornik:

1. pregled svih podataka za korisnike – broj računa, prezime, stanje računa
2. ispis podataka osoba čije je stanje računa u intervalu [DG,GG] (podatke DG i GG unosi korisnik)
3. ispis svih podataka za korisnike čije prezime počinje nekim zadanim slovom (slovo unosi korisnik)

Interakcija s programom završava kada korisnik odabere kraj.

11. Napišite program koji će omogućiti evidenciju podataka za n gostiju hotela. Formira se lista kao polje (elementi polja su tipa strukture) za gosta sa sljedećim podacima: prezime gosta, broj osobne iskaznice/putovnice, broj sobe, broj noćenja. Korisnik najprije unosi broj n ($0 < n < 99$) – za koliko korisnika unosi podatke, a potom unosi sve podatke za n korisnika. Nakon toga na ekranu se ispisuje sljedeći izbornik:

1. pregled svih podataka za goste – prezime gosta, broj osobne /putovnice, broj sobe, broj noćenja
2. ispis prezimena gostiju koji imaju više od nekog zadanog broja noćenja (broj noćenja unosi korisnik)
3. ispis svih podataka za goste čije prezime počinje nekim zadanim slovom (slovo unosi korisnik)

Interakcija s programom završava kada korisnik odabere kraj.

12. Napišite program koji će korisniku omogućiti evidenciju zarade web-shopa. Formira se lista kao polje (elementi polja su tipa strukture) za proizvod sa sljedećim podacima: šifra proizvoda, naziv proizvoda, cijena proizvoda, prodana količina. Nakon toga potrebno je omogućiti unos podataka za proizvode koji su prodani: najprije se unosi broj proizvoda za koji se upisuju podaci, a potom svi podaci. Potrebno je realizirati sljedeći izbornik:

1. ispis svih podataka za sve prodane proizvode

2. ispis ukupne zarade od svih prodanih proizvoda
3. ispis podataka za proizvod (ili više njih) koji je ostvario najveću prodaju

Interakcija s programom završava kada korisnik odabere kraj.

13. Napišite program koji će korisniku omogućiti praćenje rada na projektima. Formira se lista kao polje (elementi polja su tipa strukture) za projekt sa sljedećim podacima: šifra projekta, naziv projekta, broj sati rada na projektu. Potrebno je omogućiti unos broja n – koliko projekata imamo (broj n unosi korisnik uz ograničenje da mora biti iz segmenta $[1,80]$). Nakon toga unose se svi podaci za sve projekte te se ispisuje sljedeći izbornik:

1. ispis svih podataka za sve projekte
2. ispis podataka za projekt s nekom zadanom šifrom (šifru unosi korisnik)
3. ispis projekata koji imaju više od nekog određenog broja radnih sati (granicu unosi korisnik)

Interakcija s programom završava kada korisnik odabere kraj.

10 Funkcije

1.♦ Napišite program koji sadrži funkciju za kvadriranje broja. Unos vrijednosti i ispis rezultata vrši se u funkciji *main*.

2.♦ Nadopunite prethodni program u koji korisnik unosi broj *n*, a poziva se funkcija koja vraća apsolutnu vrijednost broja. Apsolutna vrijednost broja pohranjuje se u varijablu u funkciji *main*, a potom se ispisuje.

3. Napišite program u koji korisnik unosi broj *n*, a poziva se funkcija koja ima jedan formalni parametar tipa *int* i vraća vrijednost *true* ako je parametar parni broj, a *false* ako je neparan. Rezultat se ispisuje na zaslon računala.

4. Napišite program za ispis kvadratne matrice popunjene zvjezdicama. Broj redaka / stupaca unosi korisnik, a on mora biti u intervalu [2, 10]. Provjeru vrijednosti potrebno je realizirati u zasebnoj funkciji, a unos se ponavlja sve dok vrijednost ne bude u danom intervalu.

5. Napišite program koji sadrži jednostavnu funkciju *posnegative* koja provjerava je li broj pozitivan ili negativan te vraća vrijednost 'P' ili 'N'. Unos broja i ispis rezultata vrši se u funkciji *main*.

6. Napišite program koji sadrži funkciju koja uzima dva cjelobrojna parametra i vraća logičku vrijednost. U funkciji *main* ispisuje se logička vrijednost koja je istinita ako je kvadratni korijen iz prvog parametra jednak drugom parametru.

7. Modificirajte prethodni zadatak tako da napišete jednostavnu funkciju koja kao rezultat vraća apsolutnu vrijednost broja na temelju rezultata koje vraća funkcija *posnegative*. Rezultat se sprema u varijablu u funkciji *main*, a potom se i ispisuje sadržaj te varijable.

8. Napišite program u koji korisnik unosi broj ukupno postignutih bodova, a u zasebnoj funkciji se utvrđuje i vraća ocjena koju je student zaslužio na temelju postignutih bodova (<50 (1); 50-60 (2); 60-75 (3); 75-90 (4); 90-100 (5)).

9. Napišite program u koji korisnik unosi neki znak. Poziva se funkcija s jednim parametrom tipa *char* i vraća vrijednost *true* ako je parametar broj, a *false* ako nije. Rezultat se ispisuje na zaslon računala.

10. Napišite program u koji korisnik unosi tri znaka za redom (ne treba koristiti niz znakova, već možete i tri varijable tipa *char*) koji predstavljaju neku jednostavnu korisničku šifru. Pravilo kreiranje šifre je da ona ne smije sadržavati slova i brojeve, već mora biti kreirana od preostalih znakova. Potrebno je napisati funkciju *verifyCode* koja provjerava je li šifra korektno unesena i vraća

logičku vrijednost. U glavnom dijelu programa treba iskoristiti funkciju te ispisati poruku korisniku je li točno upisao podatke za kreiranje šifre.

11. • Napišite program koji će za upisane duljine stranica pravokutnika izračunati i ispisati površinu tog pravokutnika. Unos stranica pravokutnika i ispis rješenja realizirati u glavnoj funkciji. Potrebno je napisati dvije funkcije pomoću kojih će se realizirati traženo rješenje. Funkcija *f1* neka bude napisana tako da vraća vrijednost (rezultat) preko imena, a druga funkcija *f2* neka bude napisana tako da vraća vrijednost (rezultat) preko reference.

12. Napišite program u koji korisnik unosi stranicu kvadrata, a u funkciji se izračunava površinu i opseg kvadrata. Rezultati se ispisuju u glavnom dijelu programa.

13. Napišite program u kojem se simulira bacanje novčića. Korisnik najprije unosi broj bacanja *n*. Poziva se funkcija *flip* koja vraća vrijednost preko reference u kojoj se generira slučajni broj u rasponu od 1 do 100 i bilježi broj pobjeda glave i pisma. Ako je generirani broj paran, dobiva glava, a ako je neparan, dobiva pismo. U funkciji *main* ispisuje se je li pobijedila glava ili pismo.

14. • Napišite funkcije za unos i ispis varijable tipa korisnički definirane strukture *car* koja se sastoji od marke, tipa i godine proizvodnje. Korisnik unosi podatke o dva automobila, a potom se podaci ispisuju.

15. Napišite program koji sadrži funkciju za skraćivanje razlomka. Vrijednosti za brojnik i nazivnik unose se u jednoj funkciji, a potom se poziva druga funkcija i ispisuje skraćeni razlomak.

16. Napišite program koji konvertira 24-satni zapis u 12-satni zapis. Primjerice, vrijeme 14:25 treba se konvertirati u 2:25 PM. Ulaz u program čine 2 int vrijednosti. Program treba sadržavati najmanje tri funkcije – jednu za unos, drugu za konverziju i treću za ispis. Korisnik treba imati mogućnost izvođenja konverzije koliko god puta on to želi.

Test podaci:

12:30 → 12:30 PM

0:20 → 0:20 AM

13:00 → 1:00 PM

8:25 → 8:25 AM

17. Napišite program koji računa ratu inflacije u protekloj godini. Korisnik unosi cijenu nekog artikla prije godinu dana i danas. Program izračunava ratu inflacije tako da podijeli razliku cijena s cijenom protekle godine. Program treba omogućiti korisniku ponavljanje izračuna koliko god korisnik puta želi. Program treba sadržavati najmanje 3 funkcije, jednu za unos cijena, drugu za računanje

rate inflacije i treću za ispis rate inflacije.

$$rata = (novaCijena - staraCijena) / staraCijena \times 100$$

18. Napišite igricu u kojoj se generiraju slučajni brojevi od 1 do 6 u funkciji *genNum*. Nakon što se generira prvi broj, on se ispisuje na zaslon i korisnik mora upisati 'V' ili 'M' ako misli da će sljedeći generirani broj biti veći (V) ili manji (M) od prethodno generiranog broja. Igra se nastavlja sve dok korisnik ne skupi ukupno 5 bodova. Korisnik dobiva po 1 bod za svaki pogodak. U slučaju promašaja, igra je završena. Ako je sljedeći generirani broj jednak prethodnom, korisnik niti dobiva, niti gubi bodove, a igra se nastavlja.

11 Polja u funkcijama

1. Napišite program u koji korisnik unosi rečenicu i neki znak, a poziva se funkcija *occurNum* koja vraća koliko puta se dani znak pojavljuje u rečenici.

2. • Napišite program koji sadrži sljedeće funkcije:

- funkcija *num* ne vraća vrijednost, a u njoj se unosi broj elemenata nekog polja *n*
- u funkciji *fill* se od korisnika traži unos realne vrijednosti kojima se popunjava polje (popunjavanje prestaje kad je u polje uneseno *n* elemenata ili kad korisnik unese nenumeričku vrijednost)

```
cin>>temp;
if (!cin)
{
    cin.clear(); //resetira podatkovni član cin objekta koji opisuje stanje, kako
    bi se mogao nastaviti unos
    while (cin.get()!='\n') continue; //rješavanje nedozvoljenog unosa
}
```

- funkcija *write* ispisuje elemente polja.
- funkcija *reorder* preslaguje elemente polja tako da prvi element postaje zadnji, drugi element predzadnji, itd.

3. Napišite program koji od korisnika traži unos cijelog broja *n*, a potom i *n* cjelobrojnih vrijednosti. Izlaz iz programa je dvostupčani ispis – u 1. stupcu nalaze se jedinstvene vrijednosti, a u 2. stupcu broj pojavljivanja tih vrijednosti.

Test podaci: -12 3 -12 4 1 1 -12 1 -1 1 2 3 4 2 3 -12

n	broj
-12	4
3	3
4	2
1	4
-1	1
2	2

4. • Napišite program koji omogućava rad s dvodimenzionalnim poljem koje je kvadratna matrica - matrica s istim brojem redaka i stupaca. Program treba omogućiti sljedeće aktivnosti točno zadanim redoslijedom:

- unos broja *n* (najviše 10, učitavati dok se ne učitava ispravno) koji predstavlja broj redaka i stupaca
- unos elemenata matrice koji su cijeli brojevi

- ispis unesenih podataka u matričnom obliku (ispis $n \times n$).
- ispis zbroja elemenata na glavnoj dijagonali
- ispis zbroja elemenata na sporednoj dijagonali

Svaki dio koda koji se može izdvojiti kao zasebna cjelina potrebno realizirati primjenom funkcija.

5. Napišite program koji korisniku omogućava zbrajanje cjelobrojnih matrica. Unos dimenzija i unos elemenata matrice vrši se u funkciji *input*, a ispis elemenata u matričnom obliku tako da se iza svakog elementa ispiše još i znak *** u funkciji *write*. Zbrajanje matrica moguće je samo ako su im brojevi redaka jednaki, kao i brojevi stupaca. Uvjet se provjerava u funkciji *condition* koja vraća logičku vrijednost preko imena. Matrice se zbrajaju u funkciji *add* tako da se zbroji odgovarajući element matrice *A* s odgovarajućim elementom matrice *B* ($C[i, j] = A[i, j] + B[i, j]$), a rezultat se pohranjuje u matricu *C* i potom ispisuje. Korisnik može ponavljati interakciju s programom sve dok on to želi.

Test podaci:

	A		B		C
1	3 2		0 0 5		1 3 7
1	0 0	+	7 5 0	=	8 5 0
1	2 2		2 1 1		3 3 3

6. Napišite program koji korisniku omogućava množenje cjelobrojnih matrica. Unos dimenzija i unos elemenata matrice vrši se u funkciji *input*, a ispis elemenata u matričnom obliku u funkciji *write*. Množenje matrica moguće je samo ako je broj stupaca prve matrice jednak broju redaka druge matrice. Uvjet se provjerava u funkciji *condition* koja vraća slovo *d* ili *n* preko imena, ovisno o tome je li uvjet zadovoljen. Ukoliko je uvjet zadovoljen, matrice se množe u funkciji *multiply*, a rezultat se pohranjuje u matricu *C* i potom ispisuje. Korisnik može ponavljati interakciju s programom sve dok on to želi.

Test podaci:

	A		B		C
1	2	1 2 3		1*1+2*3 = 7	1*2+2*4 = 10 1*3+2*4=11
		*	=		
3	4	3 4 4		3*1+2*3 = 15	3*2+4*4 = 22 3*3+4*4 = 25

7. Napišite program u koji korisnik unosi elemente matrice koji su realni brojevi, a ispisuje se transponirana matrica. Ako matrica *A* ima dimenzije $m \times n$, onda transponirana matrica A^T ima dimenzije $n \times m$, a nastaje pretvaranjem stupaca u retke i redaka u stupce, tj. $A^T[i, j] = A[j, i]$. Unos dimenzija matrice vrši se u funkciji *main*, a unos elemenata matrice u funkciji *input*. Transponiranje matrice pri ispisu, odnosno izravni ispis transponirane matrice implementirati u funkciji *writeT*. U programu se najprije poziva funkcija za ispis, a potom za transponiranje matrice i ispis. Transponirana matrica generira se u funkciji

translate, a rezultat se pohranjuje u matricu B i potom ispisuje. Korisnik može ponavljati interakciju s programom sve dok on to želi. Unos korisnikova odgovora vrši se u funkciji *answer* koja vraća vrijednost preko imena, a provjera je li korisnik unio d ili n u funkciji *main*.

Test podaci:

$$\begin{array}{ccc} \mathbf{A} & & \mathbf{A}^T \\ \begin{array}{ccc} 1 & 2 & 3 \\ & & \\ 3 & 4 & 4 \end{array} & = & \begin{array}{cc} 1 & 3 \\ 2 & 4 \\ 3 & 4 \end{array} \end{array}$$

Napomena: Ispisom transponirane matrice dobije se ponovno originalna matrica jer se i prilikom ispisa matrica ponovno transponira.

8. • Napišite program koji korisniku omogućava evidenciju zarade web-shopa. Formira se lista kao polje (elementi polja su tipa strukture) za proizvod sa sljedećim podacima: šifra proizvoda, naziv proizvoda, cijena proizvoda, prodana količina. Nakon toga korisnik unosi podatke za proizvode koji su prodani (najprije se unosi broj proizvoda za koji se upisuju podaci, a potom svi podaci). Potrebno je realizirati sljedeći izbornik:

- ispis svih podataka za sve prodane proizvode
- ispis ukupne zarade od svih prodanih proizvoda
- ispis podataka za proizvod koji je ostvario najveću zaradu

Potrebno je omogućiti ponovni odabir iz izbornika nakon što je jedna opcija odabrana (ponuditi korisniku dvije opcije: povratak na izbornik / izlaz iz programa) i omogućiti samo ispravan unos vrijednosti (ograničiti unos odabira iz izbornika, cijena ne može biti negativna vrijednost). Svaki dio koda koji se može izdvojiti kao zasebna cjelina potrebno realizirati primjenom funkcija.

9. Napišite program koji će korisniku omogućiti praćenje rada na projektu. Definirajte strukturu *project* sa sljedećim podacima: šifra projekta, naziv projekta, predviđeni broj sati rada i realizirani broj sati rada na projektu. Potrebno je omogućiti unos svih podataka (funkcija *input*), a potom izračunati i ispisati preostali broj sati rada (funkcija *calculate*).

Test podaci:

Unesi šifru -> PIRSVES

Unesi naziv projekta -> Projekt izrade racunalnog sustava za vodjenje evidencije studija

Unesi broj predvidjenih sati rada -> 2400

Unesi broj realiziranih sati rada -> 1260

Preostali broj sati rada -> 1140

10. Napišite program koji će omogućiti evidenciju podataka za gosta hotela. Definirajte strukturu *guest* sa sljedećim podacima: prezime gosta, broj osobne iskaznice/putovnice, broj sobe, broj noćenja. Korisnik unosi sve podatke za gosta (funkcija *input*), a potom se ispisuje ukupan iznos koji gost mora platiti (broj noćenja x 720,00) (funkcija *checkout*).

Test podaci:

Unesi prezime -> Brkic
Unesi broj osobne iskaznice / putovnice -> 346432
Unesi broj sobe -> 403
Unesi broj nocenja -> 4

Zaduzenje -> 2880,00

11. Napišite program u kojem je definirana struktura *person* koja se sastoji od imena, prezimena i korisničkog imena (voditi računa ima li korisnik 2 imena ili prezimena). Nakon unosa imena i prezimena u funkciji *generate* generira se korisničko ime (username = Ime + . + Prezime). Korisničko ime ispisuje se na zaslon.

Test podaci:

Unesi ime -> MARIJA MAJA
Unesi prezime -> BRKIC

Ime: MARIJA MAJA
Prezime: BRKIC
Username: Marija maja.Brkic

12 Nadjačavanje funkcija

1. Je li dozvoljeno sljedeće nadjačavanje funkcije kvadrat? Objasni!

```
double kvadrat(float);  
float kvadrat(float);
```

2. Je li dozvoljeno sljedeće nadjačavanje funkcije kvadrat? Objasni!

```
long kvadrat(int);  
long kvadrat(long);
```

3. Pronađite greške u sljedećem programu i prokomentirajte zbog čega su one nastale te način na koji bi se mogle ispraviti!

```
#include<iostream>  
using namespace std;
```

```
long kvadrat(long);  
float kvadrat(float);
```

```
int main()  
{  
    cout<<kvadrat(42);  
    cout<<kvadrat(3.4);  
    return 0;  
}
```

```
long kvadrat(long n)  
{  
    return n*n;  
}
```

```
float kvadrat(float n)  
{  
    return n*n;  
}
```

4. C biblioteke su već postojale kad je nastao C++. Budući da su se mogle jednostavno koristiti, one se nisu mijenjale kako bi se iskoristila karakteristika nadjačavanja funkcija. Iz tog razloga postoji nekoliko funkcija za apsolutnu vrijednost broja. Napišite nekoliko funkcija *abs* tako da mogući parametri budu tipa *int*, *float* ili *double*.

5. Napišite 2 funkcije *ascend*. Jedna funkcija ima dva, a druga tri formalna parametra tipa *int*. Funkcije vraćaju vrijednost *true* ako su njihovi parametri poredani rastuće, a vrijednost *false* ako nisu. Ispis rezultata vrši se u glavnoj

funkciji.

6. Napišite 3 funkcije *avg*. Jedna funkcija računa prosjek 2 broja, druga računa prosjek 3 broja, a treća računa prosjek 4 broja.

7.♦ Napišite 2 funkcije za kvadriranje – jedna ima cjelobrojni parametar, a druga realni parametar.

8.♦ Napišite funkciju koja obično uzima jedan argument (znakovni niz) i ispisuje ga jednom. Ako se pri pozivu funkcije proslijedi i drugi argument *integer* n i on je veći od 0, tada se znakovni niz ispisuje n puta.

9. Napiši 2 funkcije *pocetak*. Jedna funkcija ispisuje prvih n znamenki nekog broja, a druga funkcija ispisuje prvih n slova neke riječi. Korisnik unosi grad i poštanski broj, a zatim i broj znamenki i slova koje želi vidjeti. Na zaslonu se ispisuje rezultat.

10.♦ Napišite program koji pretražuje datoteku *numbers.txt* koja sadrži *integer* vrijednosti i ispisuje najveću i najmanju vrijednost na zaslon.

13 Datoteke

1. Napišite program koji pretražuje datoteku *floats.txt* koja sadrži *double* vrijednosti i ispisuje njihov prosjek na zaslon.
2. Program učitava 3 broja iz datoteke *ulaznaDatoteka.txt*, zbraja brojeve te sumu zapisuje u datoteku *izlaznaDatoteka.txt*.
3. Napišite program pomoću kojeg korisnik dodaje cjelobrojne vrijednosti na kraj datoteke *ulaznaDatoteka.txt* sve dok ne unese vrijednost 0, koja služi kao oznaka za kraj unosa.
4. • Modificirajte prethodni zadatak tako da se zbrajaju svi brojevi iz datoteke *ulaznaDatoteka.txt* u funkciji *suma*. U programu neka se izvrši provjera je li datoteka otvorena bez greške. Ime datoteke unosi korisnik.
5. Napišite program za evidenciju prodaje (evidencija prodaje je u ovom zadatku potpuno pojednostavljena, prati se samo broj prodanih računala u periodu od 12 mjeseci). Program treba omogućiti korisniku unos broja prodanih računala za točno 12 mjeseci (računa se kao godina dana od 1. do 12. mjeseca). Podaci se bilježe i u datoteci *prodaja.txt*. Nakon unosa potrebno je izračunati koliki je prosjek prodaje te u kojim mjesecima je prodaja natprosječna. Ispisati sve podatke u datoteci *obracun.txt*.
6. Napišite program u kojem će se generirati određeni broj slučajnih brojeva iz segmenta $[0,9]$. U programu je potrebno upisati željeni broj slučajnih brojeva, max 10000, a potom generirati brojeve. Program treba raditi tako da za svaki od brojeva od 0 do 9 zapamti koliko puta je broj generiran. Na kraju treba na ekranu ispisati za svaki od brojeva koliko puta je generiran. Rezultati se pohranjuju u datoteci *rezultati.txt*. Prokomentirajte jesu li svi brojevi jednako zastupljeni.
7. Napišite program koji kopira ulaz s tipkovnice u datoteku čije ime unosi korisnik. Ulaz je omogućen sve dok korisnik ne unese znak '@', koji služi kao oznaka za kraj.
8. • Napišite program koji u funkciji *copyFile* kopira sadržaj jedne datoteke u drugu datoteku. Ime ulazne i izlazne datoteke unosi korisnik.
9. Napišite program koji otvara dvije tekstualne datoteke za unos i jednu za ispis. Imena datoteka unosi korisnik. Program treba u funkciji *mergeFiles* spojiti odgovarajuće linije ulaznih datoteka (pri čemu razmak služi kao separator) i sadržaj pohraniti u izlaznu datoteku. Ako je jedna datoteka duža od druge,

njene linije dodaju se na kraj izlazne datoteke.

Test podaci:

Ulazna datoteka 1:

jabuke kruške
paprika

Ulazna datoteka 2:

banane
krastavci kupus
šećer sol

Izlazna datoteka:

jabuke kruške banane
paprika krastavci kupus
šećer sol

10. Napišite program koji otvara dvije tekstualne datoteke za unos i jednu za ispis. Imena datoteka unosi korisnik. Program treba u funkciji *mergeUniqueLines* spojiti ulazne datoteke i sadržaj pohraniti u izlaznu datoteku. U ulaznim datotekama nalaze se liste imena i to posložena po abecednom redoslijedu. Izlazna datoteka treba sadržavati spoјenu listu imena posloženu po abecednom redu bez duplih imena.

Test podaci:

Ulazna datoteka 1:

ana
bojan
marta
tin

Ulazna datoteka 2:

marta
nina
tea

Izlazna datoteka:

ana
bojan
marta
nina
tea
tin

11. Napišite program koji u funkciji *clearFile* čita sadržaj jedne datoteke i pohranjuje uređenu verziju te datoteke pod drugim imenom. Ime ulazne datoteke unosi korisnik, a ime izlazne tvori se tako da se na ime ulazne doda nastavak

«-clean». Sadržaj ulazne datoteke uređuje se tako da se izbacuju suvišne praznine u tekstu (kada program nađe na više od jednog razmaka, odbacuje sve praznine do sljedećeg znaka). Ime ulazne i izlazne datoteke unosi korisnik.

12. Napišite program koji u funkciji *avgWordLen* izračunava prosječnu duljinu riječi (prosječan broj znakova u riječi) za datoteku u kojoj se nalazi neki tekst. Riječ je niz simbola koja završava jednim od sljedećih znakova: praznina, zarez, točka, kraj linije. Ime ulazne datoteke unosi korisnik.

13. Napišite program koji u funkciji *correctSourceCode* ispravlja C++ program koji ima greške kod korištenja operatora « ili » sa cin i cout tokovima. Program ispravlja svako pojavljivanje cin « sa cin » i cout » sa cout «. Pretpostavka je da se nakon cout i cin objekta nalazi točno jedna praznina. Ime ulazne i izlazne datoteke unosi korisnik.

14. Napišite program koji u funkciji *countLines* broji linije u nekoj datoteci. Svaka linija datoteke ispisuje se na zaslon numerirana. Broj linije ispisuje se u polju širokom 3 znaka. Ime ulazne datoteke unosi korisnik.

15. Napišite program koji u funkciji *addContents* dodaje sadržaj datoteke u *izlaz.txt* datoteku. Korisnik unosi ime ulazne datoteke.

16.♦ Definirajte strukturu *student* koja se sastoji od imena, prezimena i matičnog broja indeksa. Korisnik unosi podatke za *n* studenata. Podaci o studentima unose se u funkciji *write* i trajno pohranjuju u datoteku *students.txt* u sljedećem formatu:

```
indeks1 – prezime1, ime1  
indeks2 – prezime2, ime2  
...
```

17. Napišite program koji u funkciji *processStudents* izračunava prosječan broj bodova studenata iz određenog kolegija. Ulazna datoteka u sljedećem formatu: ime prezime broj_bodova. Podaci se učitavaju kao instance strukture *student*. Broj bodova cjelobrojna je vrijednost i ukupno ih je deset u svakom retku. Izlazna datoteka sadrži kopiju svih podataka iz ulazne datoteke, a na kraju svakog retka nalazi se i prosječan broj bodova studenta. Ime ulazne i izlazne datoteke unosi korisnik.

18. Definirajte strukturu *student* koja se sastoji od imena, prezimena i općeg uspjeha učenika. Korisnik unosi odgovarajuće podatke za *n* učenika, a unos se vrši i u datoteci *list.txt* u sljedećem formatu:

```
PREZIME, Ime
```

```
> +3.00
```

Širina polja opći uspjeh je 10. Opći uspjeh se prikazuje s predznakom na 2

decimale (točka i decimalne nule se također prikazuju).

19. Svaki zapis postojeće direktne neformatirane (binarne) datoteke *studenti.dat* sadrži podatke o jednom studentu: matični broj studenta (int), ime i prezime studenta (niz znakova duljine 50) i ukupni broj bodova koje je stekao (*float*). Redni broj zapisa datoteke *studenti.dat* odgovara matičnom broju studenta. Nakon unosa studenata i broja bodova, učitava se jedan matični broj studenta. Ako zapis s učitanim matičnim brojem postoji u datoteci, broj bodova u tom zapisu datoteke uvećava se za 10%.

20. Napišite program koji će ispisati koliko bajtova sadrži određena datoteka.

21. Napišite program koji će ispisati sadržaj datoteke onako kako je on zapisan u datoteci.

22. Napišite program koji će ispisati koji je najčešće korišten znak engleske abecede u nekoj slijednoj formatiranoj datoteci. Ukoliko ih ima više, potrebno ih je sve ispisati.

23. Napišite program iz kojeg se poziva funkcija koja čita sve znamenke koje se nalaze u nekoj slijednoj formatiranoj datoteci i zapisauje ih u drugu takvu datoteku. Ulazna datoteka može sadržavati i tekst. Funkcija treba vratiti broj pročitanih znamenaka ili nulu ukoliko se datoteka ne može otvoriti.

24. U slijednoj formatiranoj datoteci *podaci.txt* se nalaze podaci o nekoj obitelji. Prvi podatak u datoteci je broj zapisa sljedećeg oblika: Ime (25+1), Prezime(25+1), Godine (int). Napišite program iz kojeg se poziva funkcija koja će vratiti prosječnu starost članova obitelji. Ukoliko nije moguće otvoriti datoteku, funkcija vraća 0. Koliko iznosi veličina datoteke ako su u njoj pohranjena 2 zapisa?

25. U direktnu neformatiranu datoteku *podaci.txt* se spremaju podaci o stanju skladišta neke tvrtke. Struktura zapisa je sljedeća: Ime artikla (50+1), Proizvođač (50+1), Količina (int). Napišite program koji zapisuje n zapisa u datoteku, a zatim ispisuje sve zapise u datoteci.

26. U direktnoj neformatiranoj datoteci se nalaze podaci o učenicima nekog razreda. Struktura zapisa je sljedeća: Ime (25+1), Prezime (25+1), Uspjeh (int). Napišite funkciju koja će svim učenicima s ocjenom 1 ispraviti ocjenu na 2. Funkcija vraća broj ispravljenih ocjena. Ukoliko nema negativnih ocjena (jedinica) ili ako se datoteka ne može otvoriti, funkcija vraća nulu.

14 Rekurzija

1. Napišite program u kojem se poziva rekurzivna funkcija za određivanje je li broj prost. Prototip funkcije je:

short jeProst(int broj, int djelitelj);

Funkcija vraća 1 ako je broj prost, inače vraća 0. Pri prvom pozivu funkcije djelitelj je postavljen na 2.

Napomena: Nerekurzivno rješenje neće se priznavati.

2. Napišite program u kojem se poziva rekurzivna funkcija za izbacivanje znamenki iz zadanog niza. Prototip funkcije treba biti:

*int IzbaciZnamenke (char *niz);*

Funkcija vraća broj izbačenih znamenki.

Napomena: Nerekurzivno rješenje neće se priznavati.

3. Neka je zadan tip podatka *Stog* za koji treba definirati funkcije za inicijalizaciju stoga, dodavanje elementa na stog te brisanje elementa sa stoga. Elementi stoga su podaci tipa *Student* koji sadrže prezime studenta, ime studenta te broj položenih ispita. Prototipovi navedenih funkcija su:

*void initStog(Stog *ℓstog);*
*int dodaj(Student element, Stog *ℓstog);*
*int skini(Student *element, Stog *ℓstog);*

Funkcije *dodaj* i *skini* vraćaju 1 ako je operacija dodavanja ili skidanja uspjela, a 0 inače. Napišite rekurzivnu funkciju koja će sa stoga maknuti sve studente koji nisu položili niti jedan ispit. Funkcija vraća broj studenata maknutih sa stoga. Prototip funkcije treba biti:

*int makniStudente (Stog *ℓstog);*

Nakon unosa podataka, poziva se funkcija *makniStudente* i ispisuje se ažurirana lista.

Napomena: Nerekurzivno rješenje neće se priznavati.

4. Za pripremu, napišite program koji će u tekstualnu datoteku zapisati niz koji sadrži između 500 i 4000 slučajno odabranih brojeva iz intervala između 0 i 10 tisuća. Svaki broj zapisan je u novom retku. Napišite rekurzivnu funkciju koja će sortirati zadano polje brojeva algoritmom odabira/razmjene (*selection*). Zadano polje pročitajte u glavnom programu iz datoteke (memoriju alocirajte

dinamički, a koristite datoteku iz pripremnog dijela zadatka). U rekurzivnoj funkciji sortirajte elemente polja. U glavnoj funkciji ispišite sortirano polje i spremite ga u novu datoteku. Pri pozivu funkcije naznačite želite li sortirati uzlazno ili silazno.

Napomena: Nerekurzivno rješenje neće se priznavati.

5. Za pripremu, napišite program koji će u tekstualnu datoteku zapisati niz koji sadrži između 500 i 4000 slučajno odabranih brojeva iz intervala između 0 i 10000. Svaki broj zapisan je u novom retku. Napišite rekurzivnu funkciju koja će sortirati zadano polje brojeva algoritmom biranja (*bubble*). Zadano polje pročitajte u glavnom programu iz datoteke (memoriju alocirajte dinamički, a koristite datoteku iz pripremnog dijela zadatka). U rekurzivnoj funkciji sortirajte elemente polja. U glavnoj funkciji ispišite sortirani niz i spremite ga u novu datoteku. Pri pozivu funkcije naznačite želite li sortirati uzlazno ili silazno.

Napomena: Nerekurzivno rješenje neće se priznavati.

6. Napišite program u kojem se poziva rekurzivna funkcija za ispis binarnog ekvivalenta nekog dekadskog broja n . Znamenke binarnog ekvivalenta ispisuju se obrnutim redoslijedom. Korisnik unosi novi broj sve dok ne upiše 0, a ispisuju se njihovi binarni ekvivalenti.

Upute: Podijeliti decimalni broj s 2. Ako je djeljiv bez ostatka, zapisati 0, a ako je djeljiv s ostatkom, zapisati 1. Postupak nastaviti s rezultatom dijeljenja. Nove znamenke upisivati s lijeve strane.

Napomena: Nerekurzivno rješenje neće se priznavati.

7. Napisati rekurzivnu funkciju kojom se dekadskom broju n ispisuju znamenke njegovog binarnog ekvivalenta. Prikažite primjenu funkcije pozivanjem iz glavnog programa.

Napomena: Nerekurzivno rješenje neće se priznavati.

8. Napišite program koji ima rekurzivnu funkciju za računanje faktoriijela nekog broja koji unosi korisnik. Ukoliko korisnik ne unese vrijednost u rasponu $[0,10]$, traži se ponovni unos. Nakon što se unese broj iz zadanog intervala, ispisuje se faktoriijel tog broja.

UPUTA:

$$0!=1$$

$$1!=1$$

$$2!=1*2=2$$

$$3!=1*2*3=6$$

...

Napomena: Nerekurzivno rješenje neće se priznavati.

9. Napišite program koji ima rekurzivnu funkciju koja za učitani prirodni broj n ispisuje znamenke tog broja jednu ispod druge u obrnutom poretku ($123 \leftarrow 3$ 2 1).

Napomena: Nerekurzivno rješenje neće se priznavati.

10. Napišite program koji ima rekurzivnu funkciju koja za učitani prirodni broj n ispisuje znamenke tog broja jednu ispod druge.

Napomena: Nerekurzivno rješenje neće se priznavati.

11. Napišite rekurzivnu funkciju koja računa zbroj znamenki za učitani prirodni broj n . Prikažite primjenu funkcije pozivanjem iz glavnog programa. Potrebno je napisati tri verzije funkcije:

- s korištenjem lokalne varijable
- bez korištenja lokalne varijable
- rezultat se pohranjuje u parametar

Napomena: Nerekurzivno rješenje neće se priznavati.

12. Napišite program koji ima rekurzivnu funkciju koja pronalazi najmanji element nekog zadanog niza cijelih brojeva.

Napomena: Nerekurzivno rješenje neće se priznavati.

13. Napišite rekurzivnu funkciju koja računa sumu aritmetičkog niza od člana A do člana B s razmakom D . Niz mora biti uzlazni ($A < B$), inače funkcija vraća 0. Prototip funkcije glasi:

int suma(int a, int b, int d);

Funkcija se poziva u glavnoj funkciji.

Napomena: Nerekurzivno rješenje neće se priznavati.

14. Napišite program koji sadrži rekurzivnu funkciju *neparneZnam* koja za svaki uneseni broj ispisuje koliko on sadrži neparnih znamenki. Interakcija s programom odvija se sve dok korisnik to želi.

Napomena: Nerekurzivno rješenje neće se priznavati.

15. Nadopunite kôd datoteke s novom rekurzivnom funkcijom za binarno pretraživanje polja.

***Napomena:** Nerekurzivno rješenje neće se priznavati.*

16. Nadopunite kôd datoteke s definicijama rekurzivnih funkcija za ispis elemenata povezane liste i obrnuti ispis elemenata povezane liste.

***Napomena:** Nerekurzivno rješenje neće se priznavati.*

17. Nadopunite kôd datoteke s funkcijom koja vraća broj elemenata povezane liste.

***Napomena:** Nerekurzivno rješenje neće se priznavati.*

18. Napišite program za praćenje stanja na nekoj listi čekanja. Definirajte odgovarajuću strukturu koja će omogućiti implementaciju reda preko povezane liste. U red se zapisuju cijeli brojevi (šifre osoba koje čekaju). Potrebno je omogućiti sljedeće operacije – za svaku definirajte funkciju:

- dodavanje novog elementa u red
- izlaz elementa iz reda – brisanje iz reda
- ispis elemenata reda
- implementirati rekurzivni algoritam koji računa i vraća duljinu povezane liste, odnosno broj osoba

***Napomena:** Nerekurzivno rješenje neće se priznavati.*

19. Napišite program koji ima rekurzivnu funkciju koja će sadržaj jednostruko povezane liste prepisati u novu, dvostruko povezanu listu zadržavajući poredak iz jednostruko povezane liste. U postojeću jednostruko povezanu listu upisani su elementi cjelobrojnog tipa. Funkcija vraća 1 ukoliko je uspješno obavila posao, a 0 inače i ima prototip:

```
int prepisi(cvorj *glavaJednostruka, cvord *EdvostrukaGlava, cvord
            *EdvostrukaRep);
```

Nakon unosa podataka ispisuju se obje liste.

***Napomena:** Nerekurzivno rješenje neće se priznavati.*

15 Najčešće pogreške

1. Ispravite grešku u danom izvornom kôdu:

```
#include <iostream>
using namespace std;

int main()
{
    int n1, n2;
    cout<<"Unesi dva broja: ";
    cin>>n1>>n2;
    if (n2!=0)
        cout << "Rezultat: " <<n1/n2 <<endl;
    else
        cout << "Dijeljenje s nulom nije dozvoljeno\n";
    return 0;
}
```

2. Ispravite grešku u danom izvornom kôdu:

```
#include <iostream>
using namespace std;

int main()
{
    // dug korisnika
    int amount;
    cout<<"Unesite iznos duga: ";
    cin>>amount;
    if (amount=0)
        cout<<"Vas dug je podmiren\n";
    else
        cout<<"Duzni ste "<<amount<<"\n";
    return 0;
}
```

3. Ispravite grešku u danom izvornom kôdu:

```
#include<iostream>
using namespace std;

int main() {
    int value;
    cout<<"Unesi broj: ";
```

```
    cin>>value;
    switch (value%2)
    {
    case 0:
        cout << "Parni_broj" << endl;
    case 1:
        cout << "Neparni_broj" << endl;
    }
    return 0;
}
```

4. Ispravite grešku u danom izvornom kôdu:

```
#include <iostream>
using namespace std;

int main()
{
    int a,b,c;
    a = 7;
    b = 5;
    c = 3;
    if (a>b>c)
        cout << "a,b,c_su_poredani\n";
    else
        cout << "a,b,c_nisu_poredani\n";
    return 0;
}
```

5. Ispravite grešku u danom izvornom kôdu:

```
#include <iostream>
using namespace std;

int main()
{
    long int one_million;
    one_million = 1,000,000;
    cout<<"One_million_" <<one_million<<endl;
    return 0;
}
```

6. Ispravite grešku u danom izvornom kôdu:

```
#include <iostream>
```

```
using namespace std;

int main()
{
    int index;
    for (index=1; index<=10; ++index);
        cout<<index<<"_squared_"<<(index * index)<<'\n';
    return 0;
}
```

7. Ispravite grešku u danom izvornom kôdu:

```
#include <iostream>
using namespace std;

int main()
{
    cout<<"The_Amazing_Accounto_aplikacija_zbraja_i_
        racuna_prosjek_";
    cout<<"pet_vrijednosti.\n";
    cout<<"Molimo_unesite_vrijednosti:\n";
    double number;
    double sum=0.0;
    for (int i=1; i<=5; i++)
        cout<<" Value_"<<i<<":_";
        cin>>number;
        sum+=number;
    cout<<" Zbroj:_ "<<sum<<"\n";
    cout<<" Prosjek:_ "<<sum/5<<".\n";
    cout<<" Hvala_sto_ste_koristili_aplikaciju!\n";
    return 0;
}
```

8. Ispravite grešku u danom izvornom kôdu:

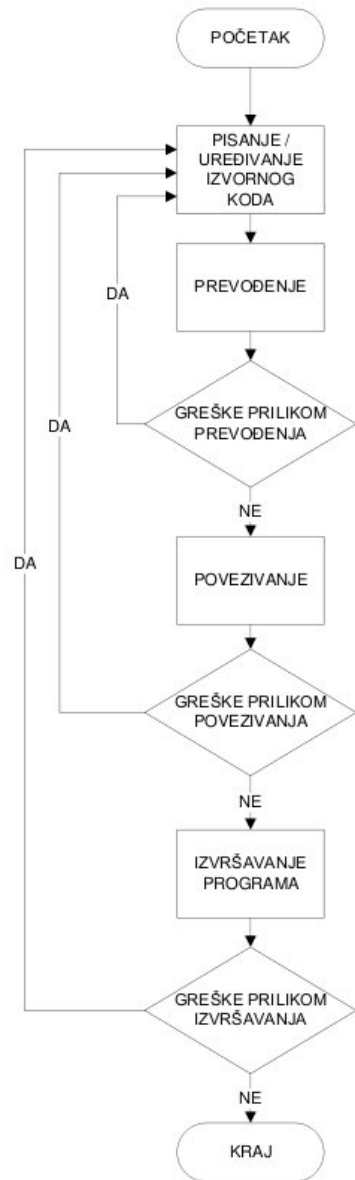
```
#include <iostream>
using namespace std;

int main()
{
    cout<<" Unesite_vrijednost_od_koje_treba_zapoceti_
        odbrojanje:_";
    int limit;
    cin>>limit;
    int i;
    for (i=limit; i; i-)
        cout<<" i=_ "<<i<<"\n";
}
```

```
    cout<<" Gotovo_buduci_da_je_vrijednost_varijable_i_="<<i<<"\n";  
    return 0;  
}
```

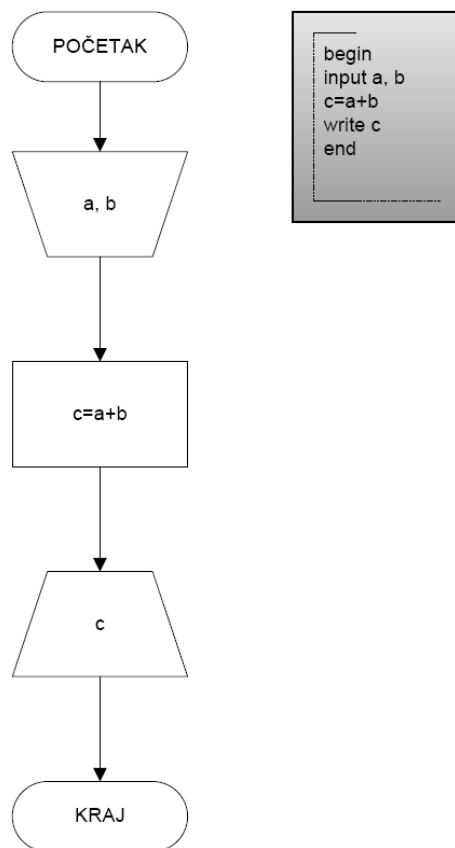
Vježba 1

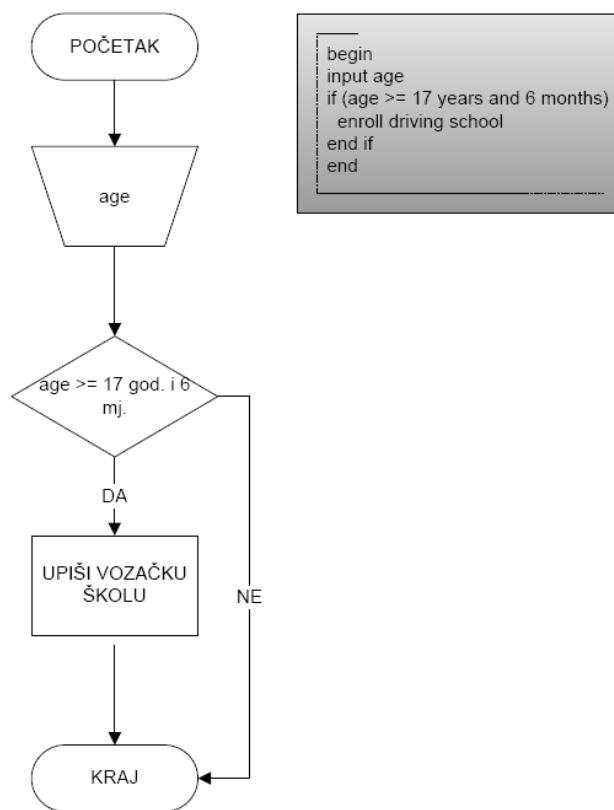
1. Razvojni ciklus C++ programa.

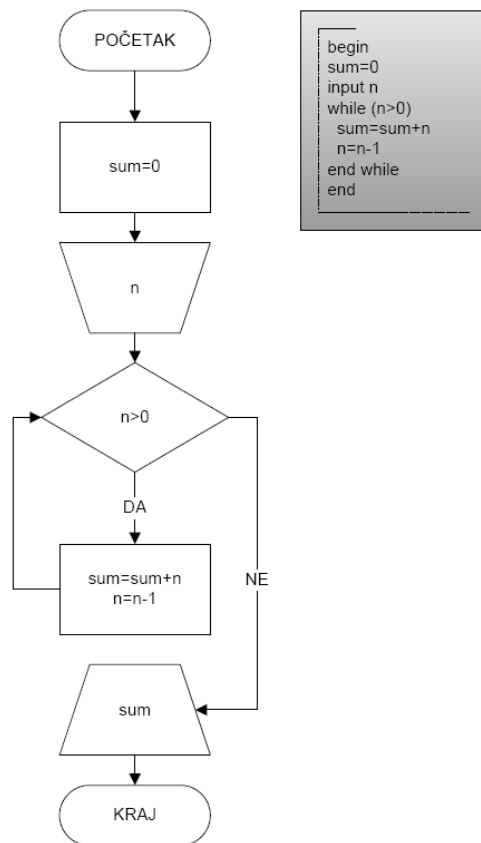


```
begin
while (not executed)
while (not linked)
while (not compiled)
write/edit code
end while
end while
end while
end
```

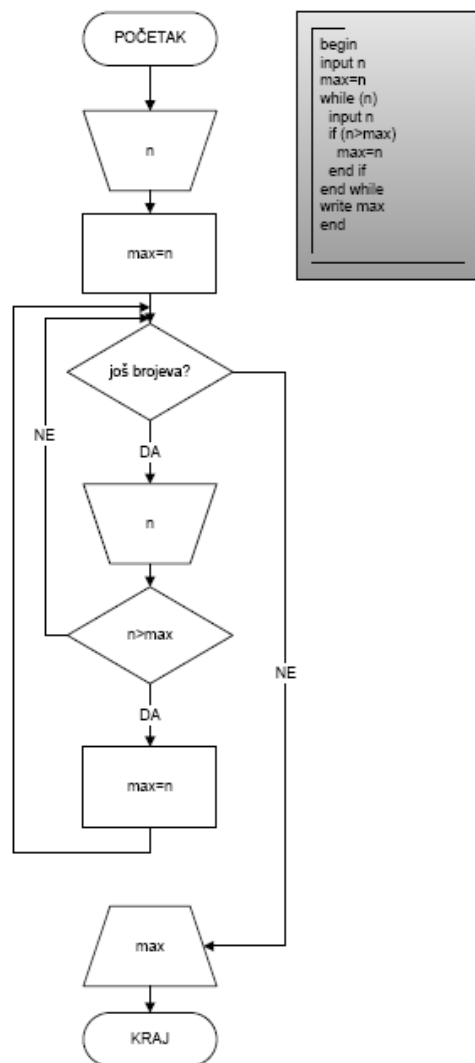
2. Zbrajanje dviju vrijednosti koje unosi korisnik



3. Naredba za grananje

5. Suma prvih n vrijednosti prirodnih brojeva.

8. Maksimum



Vježba 2

1. helloWorld.cpp

```
#include<iostream>
using namespace std;

int main()
{
    int age;
    cout << "Hello world!: -)\nYour age: ";
    cin>>age;
    return 0;
}
```

2. myInfo.cpp

```
#include<iostream>
using namespace std;

int main()
{
    cout<<"Ivo Ivic\nKralja Tomislava 15\n";
    return 0;
}
```

4. tellTime.cpp

```
#include<iostream>
using namespace std;

int main()
{
    int h, m;
    cout<<"Sat: ";
    cin>>h;
    cout<<"Minute: ";
    cin>>m;
    cout<<"Vrijeme: "<<h<<": "<<m<<endl;
    return 0;
}
```

6. codeVerification.cpp

```
#include <iostream>
using namespace std;

int main()
{
    int kod;
```

```

        cout<<" Agentu , unesi kod : _____\b\b\b\b\b\b\b\b\b\b
        ";
        cin>>kod;
        cout<<" Verifikacija koda ... "<<endl;
        cout<<"Kod verificiran !"<<endl;
        cout<<" Operacija \"Z3\" je aktivirana !"<<endl;
        return 0;
    }

```

8. checkingAccounts.cpp

```

#include<iostream>
#include<climits>
using namespace std;

int main()
{
    long ana;
    short mate;
    mate=SHRT_MAX;
    ana=mate;
    cout<<" Ana ima " <<ana<<" kn na racunu. \nMate ima "
        <<mate<<" kn na racunu. \n";
    ana++;
    mate++;
    cout<<" Ana ima " <<ana<<" kn na racunu. \nMate ima "
        <<mate<<" kn na racunu. \n";
    return 0;
}

```

10. lightyearsToAstronomicUnits.cpp

```

#include <iostream>
using namespace std;

const int AI=63240;

int main()
{
    double svjetlosne_godine , rezultat;
    cout<<" Alat za pretvorbu : svjetlosne godine -> atomske jedinice "<<endl;
    cout<<" Unesite broj svjetlosnih godina : ";
    cin>>svjetlosne_godine;
    rezultat=svjetlosne_godine*AI;
    cout<<svjetlosne_godine<<" svjet. god. = " <<rezultat;
    cout<<" astronom. jed. "<<endl;
    return 0;
}

```

11. fuelConsumption.cpp

```

#include <iostream>
using namespace std;

int main()
{
    double put, litre, potrosnja;
    cout<<"Ovo je program za izracunavanje prosjecne
        potrosnje goriva Vaseg vozila."<<endl;
    cout<<"\nUnesite udaljenost prijedjenu u kilometrima
        (km): ";
    cin>>put;
    cout<<"Unesite kolicinu potrosenog goriva u litrama (
        L): ";
    cin>>litre;
    potrosnja=(100*litre)/put;
    cout.setf(ios::fixed);
    cout.setf(ios::showpoint);
    cout.precision(2);
    cout<<"\nPotrosnja Vaseg vozila je "<<potrosnja<<
        " L/100km"<<endl<<endl;
    return 0;
}

```

12. deadline.cpp

```

#include<iostream>
using namespace std;

int main()
{
    const int s_u_d=24, m_u_s=60;
    long x, dan, sat, min;
    cout<<"Unesi broj sekundi: ";
    cin>>x;
    dan=x/(m_u_s*m_u_s*s_u_d);
    x=x%(m_u_s*m_u_s*s_u_d);
    sat=x/(m_u_s*m_u_s);
    x=x%(m_u_s*m_u_s);
    min=x/m_u_s;
    x=x%m_u_s;
    cout<<dan<<"d, " <<sat<<"h, " <<min<<"m, i " <<x<<
        "s\n";
    return 0;
}

```

13. integerDivision.cpp

```

#include <iostream>

```

```
using namespace std;

int main()
{
    int a, b;
    double c;
    cout<<"DIJELJENJE_DVIJE_CJELOBROJNE_VRIJEDNOSTI\n";
    cout<<"Prvi_operand: ";
    cin>>a;
    cout<<"Drugi_operand: ";
    cin>>b;
    cout<<"Unijeli ste brojeve: "<<a<<" i "<<b;
    c=static_cast<double>(a)/b;
    cout<<"\nKolicnik unesenih brojeva je: "<<c;
    cout<<endl;
    return 0;
}
```

Vježba 3

1. evenNums.cpp

```
#include <iostream>
using namespace std;

int main()
{
    int x, y;
    bool log;
    cout<<"Unesi 1. broj: ";
    cin>>x;
    cout<<"Unesi 2. broj: ";
    cin>>y;
    log = x%2==0 && y%2==0;
    cout<<"Oba broja su parna: "<<boolalpha<<log<<endl;
    return 0;
}
```

2. evenOrOdd.cpp

```
#include <iostream>
using namespace std;

int main()
{
    int x;
    cout<<"Unesi broj: ";
    cin>>x;
    if (x%2!=0) cout<<"Broj je neparan!\n";
    else cout<<"Broj je paran!\n";
    return 0;
}
```

3. logValue.cpp

```
#include <iostream>
using namespace std;

int main()
{
    int a, b, c;
    bool log;
    cout<<"Unesi 1. broj: ";
    cin>>a;
    cout<<"Unesi 2. broj: ";
    cin>>b;
    cout<<"Unesi 3. broj: ";
```

```

        cin>>c;
        log = a==2*b || c<1 || c>15;
        cout<<"log_="<<boolalpha<<log<<endl;
        return 0;
    }

```

4. selection.cpp

```

#include<iostream>
using namespace std;

int main()
{
    bool log;
    int d, m, g;
    cout<<"Unesi dan, mjesec i godinu rođenja i  

        odvojeno razmacima (d m g) : ";
    cin>>d>>m>>g;
    log = g>=1990 && g<=2001 && d%2!=0 && m==9;
    cout<<"Osoba zadovoljava uvjete : "<<boolalpha<<
        log<<endl;
    return 0;
}

```

15. roomCapacity.cpp

```

#include <iostream>
using namespace std;

int main()
{
    int kapacitet, broj;
    cout<<"PROVJERA POPUNJENOSTI KAPACITETA\n";
    cout<<"Unesite kapacitet sobe : ";
    cin>>kapacitet;
    cout<<"Unesite broj uzvanika : ";
    cin>>broj;
    if (broj>kapacitet)
        cout<<broj-kapacitet<<" uzvanik/a je  

            visak. "<<endl;
    else
        cout<<"Mozete pozvati jos "<<kapacitet-broja<<
            " osoba/e/a. "<<endl;
    return 0;
}

```

Vježba 4

2. digitsNum.cpp

```
#include<iostream>
using namespace std;

int main()
{
    long num;
    cout<<"Unesi broj: ";
    cin>>num;
    if (num>=-9 && num<=9) cout<<"Jednoznamenkasti broj!\n";
    else if (num>=-99 && num<=99) cout<<"Dvoznamenkasti broj!\n";
    else if (num>=-999 && num<=999) cout<<"Troznamenkasti broj!\n";
    else if (num>=-9999 && num<=9999) cout<<"Cetveroznamenkasti broj!\n";
    else cout<<"Broj ima više od 4 znamenke!\n";
    return 0;
}
```

3. analyzeChar.cpp

```
#include<iostream>
#include<cctype>
using namespace std;

int main()
{
    char znak;
    cout<<"Unesi znak: ";
    cin>>znak;
    if (isdigit(znak)) cout<<"Uneseni znak je brojka!\n";
    else if (isupper(znak)) cout<<"Uneseni znak je veliko slovo!\n";
    else if (islower(znak)) cout<<"Uneseni znak je malo slovo!\n";
    else cout<<"Uneseni znak nije brojka niti slovo!\n";
    return 0;
}
```

4. days.cpp

```
#include<iostream>
using namespace std;
```

```
enum DANI{PON=1, UTO, SRI, CET, PET, SUB, NED};

int main()
{
    int n;
    cout<<"Unesi broj: ";
    cin>>n;
    switch (n)
    {
        case PON:
            cout<<"PONEDJELJAK\n";
            break;
        case UTO:
            cout<<"UTORAK\n";
            break;
        case SRI:
            cout<<"SRIJEDA\n";
            break;
        case CET:
            cout<<"CETVRTAK\n";
            break;
        case PET:
            cout<<"PETAK\n";
            break;
        case SUB:
        case NED:
            cout<<"VIKEND\n";
            break;
        default:
            cout<<"Tjedan ima 7 dana!\n";
    }
    return 0;
}
```

5. calculatorPairity.cpp

```
#include <iostream>
using namespace std;

int main()
{
    int a, b;
    double c;
    cout<<"Unesite dvije cjelobrojne vrijednosti."<<endl;
    cout<<"Prvi(a) broj: ";
    cin>>a;
    cout<<"Drugi(b) broj: ";
    cin>>b;
    if (a%2==1 && b%2==1)
```

```

    {
        c=a+b;
        cout<<c<<endl;
    }
    else if (a%2==1 && b%2==0)
    {
        c=a-b;
        cout<<c<<endl;
    }
    else if (a%2==0 && b%2==0)
    {
        c=a*b;
        cout<<c<<endl;
    }
    else
    {
        c=static_cast<double>(a)/b;
        cout<<c<<endl;
    }
    return 0;
}

```

8. functionValueRanges-8.cpp

```

#include <iostream>
using namespace std;

int main()
{
    double x, f;
    cout<<"Program racuna i ispisuje vrijednost funkcije f(x). "<<endl;
    cout<<"Unesite realnu vrijednost x: ";
    cin>>x;
    if (x>=1 && x<=20)
    {
        f=x*10;
        cout<<" f(x)=x*10 -> f ("<<x<<" )="<<x<<"
            *10="<<f<<endl;
    }
    else if (x>=21 && x<=40)
    {
        f=x+2;
        cout<<" f(x)=x+2 -> f ("<<x<<" )="<<x<<" +2="
            <<f<<endl;
    }
    else if (x>=41 && x<=60)
    {
        f=x/3;
    }
}

```

```

        cout<<" f(x)=x/3-> f ("<<x<<")="<<x<<"/3="
        <<f<<endl;
    }
    else
    {
        f=0;
        cout<<" f(x)=0-> f ("<<x<<")="<<f<<endl;
    }
    return 0;
}

```

9. functionValueRanges-9.cpp

```

#include <iostream>
using namespace std;

int main()
{
    double x, f;
    cout<<"Program racuna i ispisuje vrijednost funkcije f(x). "<<endl;
    cout<<" Unesite realnu vrijednost x: ";
    cin>>x;
    if (x>=1 && x<=100)
    {
        f=x*5;
        cout<<" f(x)=x*5-> f ("<<x<<")="<<x<< "*5="
        <<f<<endl;
    }
    else if (x>=-100 && x<=-1)
    {
        f=x+5;
        cout<<" f(x)=x+5-> f ("<<x<<")="<<x<< "+5="
        <<f<<endl;
    }
    else if (x>=150 && x<=200)
    {
        f=x/5;
        cout<<" f(x)=x/5-> f ("<<x<<")="<<x<< "/5="
        <<f<<endl;
    }
    else
    {
        f=0;
        cout<<" f(x)=0-> f ("<<x<<")="<<f<<endl;
    }
    return 0;
}

```

10. calculatorDivisionRange.cpp

```
#include <iostream>
using namespace std;

int main()
{
    int a, b;
    double c;
    cout<<" Unesite dvije cjelobrojne vrijednosti. "<<endl;
    cout<<" Prvi(a) broj: ";
    cin>>a;
    cout<<" Drugi(b) broj: ";
    cin>>b;
    if (a%10==0)
    {
        if (b>=10 && b<=20)
        {
            c=a+b;
            cout<<c<<endl;
        }
        else
        {
            c=a*b;
            cout<<c<<endl;
        }
    }
    else if (a%10!=0 && b>=10 && b<=20)
    {
        c=a;
        cout<<c<<endl;
    }
    else
    {
        c=a/b;
        cout<<c<<endl;
    }
    return 0;
}
```

13. quadraticEquation.cpp

```
#include <iostream>
#include <cmath>
using namespace std;

int main()
{
    double a, b, c, x1, x2, d;
    cout<<" Unesi tri broja: "<<endl;
```

```

    cin>>a;
    cin>>b;
    cin>>c;
    cout<<a<<"x^2+"<<b<<"x+"<<c<<"=0"<<endl;
    d=sqrt(b*b-4*a*c);
    if(d>0)
    {
        x1=(-b+d)/2*a;
        x2=(-b-d)/2*a;
        cout<<"x1="<<x1<<endl;
        cout<<"x2="<<x2<<endl;
    }
    else if (d==0)
    {
        x1=-b/(2*a);
        cout<<"x="<<x1<<endl;
    }
    else
        cout<<"Nema realnih x1 i x2 rjesenja."<<endl;

    return 0;
}

```

15. triangle.cpp

```

#include <iostream>
using namespace std;

int main()
{
    int a, b, c;
    bool trokut;
    cout<<"Unesi 3 prirodna broja manja od 30:\n";
    cin>>a;
    cin>>b;
    cin>>c;
    if(a>=1 && a<=30 && b>=1 && b<=30 && c>=1 && c<=30)
    {
        trokut = a+b>c && a+c>b && b+c>a;
        cout<<"Zadane stranice mogu biti duljine "
              <<stranica_trokuta<<endl;
    }
    else
        cout<<"Neispravan unos brojeva!"<<endl;

    return 0;
}

```

17. order.cpp

```

#include<iostream>
using namespace std;

int main()
{
    int a, b, c;
    cout<<"Unesi brojeve iz intervala [1,100]: ";
    cin>>a>>b>>c;
    if (a<1 || a>100 || b<1 || b>100 || c<1 || c>100)
        cout<<"Broj nije iz zadanog intervala!\n";
    else
    {
        if (a>b && a>c)
        {
            cout<<a<<" ";
            if (b>c) cout<<b<<" "<<c<<endl;
            else cout<<c<<" "<<b<<endl;
        }
        else if (b>a && b>c)
        {
            cout<<b<<" ";
            if (a>c) cout<<a<<" "<<c<<endl;
            else cout<<c<<" "<<a<<endl;
        }
        else
        {
            cout<<c<<" ";
            if (a>b) cout<<a<<" "<<b<<endl;
            else cout<<b<<" "<<a<<endl;
        }
    }
    return 0;
}

```

18. calculator.cpp

```

#include <iostream>
using namespace std;

int main()
{
    int izbor;
    double rezultat;
    int a, b;
    cout<<"KALKULATOR\n\nUnesi dva cijela broja (a i b). "
        <<endl;
    cout<<"a=";
    cin>>a;
    cout<<"b=";

```

```

cin>>b;
cout<<" 1. Zbrajanje (+)"<<endl;
cout<<" 2. Oduzimanje (-)"<<endl;
cout<<" 3. Mnozenje (*)"<<endl;
cout<<" 4. Dijeljenje (/)"<<endl;
cout<<" Koji opciju zelis? Opcija broj ( upisi broj bez
    tocke): ";
cin>>izbor;
switch (izbor)
{
    case 1:
        rezultat=a+b;
        cout<<a<<"+"<<b<<"="<<rezultat<<endl;
        break;
    case 2:
        rezultat=a-b;
        cout<<a<<"-"<<b<<"="<<rezultat<<endl;
        break;
    case 3:
        rezultat=a*b;
        cout<<a<<"*"<<b<<"="<<rezultat<<endl;
        break;
    case 4:
        if (b==0)
        {
            cout<<" Dijeljenje s 0 nije dozvoljeno!"<<
                endl;
        }
        else
        {
            rezultat=static_cast<double>(a)/b;
            cout<<a<<"/"<<b<<"="<<rezultat<<endl;
        }
        break;
    default:
        cout<<" Uneseni izbor nije ponudjen!"<<endl;
}
return 0;
}

```

19. calculatorFlexibleInput.cpp

```

#include <iostream>
using namespace std;

int main()
{
    char izbor;
    double rezultat;
    int a, b;

```

```

    cout<<"KALKULATOR\n\nUnesi_dva_cijela_broja_(a_i_b)."
        <<endl;
    cout<<"a=";
    cin>>a;
    cout<<"b=";
    cin>>b;
    cout<<"1._Zbrajanje_(+)"<<endl;
    cout<<"2._Oduzimanje_(-)"<<endl;
    cout<<"3._Mnozenje_(*)"<<endl;
    cout<<"4._Dijeljenje_(/)"<<endl;
    cout<<"Koji_opciju_zelis?_Opcija_broj_(upisi_broj_bez
        _tocke):_";
    cin>>izbor;
    switch (izbor)
    {
        case '1':
        case '+':
            rezultat=a+b;
            cout<<a<<"+"<<b<<"="<<rezultat<<endl;
            break;
        case '2':
        case '-':
            rezultat=a-b;
            cout<<a<<"-"<<b<<"="<<rezultat<<endl;
            break;
        case '3':
        case '*':
            rezultat=a*b;
            cout<<a<<"*"<<b<<"="<<rezultat<<endl;
            break;
        case '4':
        case '/':
            if (b==0)
            {
                cout<<"Dijeljenje_s_0_nije_dozvoljeno!"<<
                    endl;
            }
            else
            {
                rezultat=static_cast<double>(a)/b;
                cout<<a<<"/"<<b<<"="<<rezultat<<endl;
            }
            break;
        default:
            cout<<"Uneseni_izbor_nije_ponudjen!"<<
                endl;
    }
    return 0;
}

```

20. ageCategory.cpp

```
#include <iostream>
using namespace std;

int main()
{
    int god;
    cout<<" Koliko imate godina? ";
    cin>>god;
    if (god>=0 && god<=6)
        cout<<" Vi ste malo dijete: " <<endl;
    else if (god>=7 && god<=14)
        cout<<" Vi ste osnovnoskolac/osnovnoskolka " <<endl;
    else if (god>=15 && god<=18)
        cout<<" Vi ste srednjoskolac/srednjoskolka " <<endl;
    else
        cout<<" Vi ste odrasla osoba " <<endl;
    return 0;
}
```

21. drivingLicence.cpp

```
#include <iostream>
using namespace std;

int main()
{
    int god;
    cout<<" Koliko imate godina? ";
    cin>>god;
    if (god>=1 && god<=110) //provjera moguceg intervala
    {
        if (god>=0 && god<=17)
            cout<<" Premladi ste za vozacku dozvolu. " <<endl;
        else if (god>=18 && god<=80)
            cout<<" Mozete imati vozacku dozvolu. " <<endl;
        else if (god>=81 && god<=110)
            cout<<" Prestari ste za vozacku dozvolu. " <<endl;
    }
    else cout<<" Krivi unos godina! " <<endl;
    return 0;
}
```

22. whoIsOlder.cpp

```
#include <iostream>
using namespace std;

int main()
{
    int god_a, god_b, mj_a, mj_b;
    cout<<" Unesite mjesece i godinu rođenja 2 osoba (mj
        razmak god) : "<<endl;
    cout<<" 1. osoba: ";
    cin>>mj_a>>god_a;
    cout<<" 2. osoba: ";
    cin>>mj_b>>god_b;
    if (mj_a>=1 && mj_a<=12 && mj_b>=1 && mj_b<=12 &&
        god_a>=1950 && god_a<=2004 && god_b>=1950 && god_b
        <=2004)
    {
        if (god_a>god_b)
            cout<<" 2. osoba je starija od 1.
                osobe "<<endl;
        else if (god_b>god_a)
            cout<<" 1. osoba je starija od 2.
                osobe "<<endl;
        else
        {
            if (mj_a==mj_b)
                cout<<" Osobe su jednako
                    stare "<<endl;
            else if (mj_b>mj_a)
                cout<<" 1. osoba je
                    starija od 2. osobe "<<
                    endl;
            else
                cout<<" 2. osoba je
                    starija od 1. osobe "<<
                    endl;
        }
    }
    else cout<<" Unesene vrijednosti nisu iz zadanog
        intervala. "<<endl;
    return 0;
}
```

Vježba 5

2. factorial.cpp

```
#include<iostream>
using namespace std;

int main()
{
    int n, f=1;
    do
    {
        cout<<" Unesi broj u intervalu [1,10]: ";
        cin>>n;
    }
    while (n<1 || n>10);
    for (int i=1; i<=n; i++)
    {
        f*=i;
        cout<<i<<"!="<<f<<endl;
    }
    return 0;
}
```

5. powerOf.cpp

```
#include<iostream>
#include<cmath>
using namespace std;

int main()
{
    int x;
    cout<<" Unesi broj: ";
    cin>>x;
    for (int i=0; i<x; i++)
    {
        cout<<"2^"<<i<<"="<<pow(2., i)<<endl;
    }
    return 0;
}
```

```
#include<iostream>
#include<cmath>
using namespace std;

int main()
{
    int x;
    cout<<" Unesi broj: ";
```

```
    cin>>x;
    while (x>0)
    {
        cout<<"2^"<<x-1<<"="<<pow(2.,x-1)<<endl;
        x--;
    }
    return 0;
}
```

25. feature.cpp

```
#include<iostream>
#include<cmath>
using namespace std;

int main()
{
    int x, a, c=1;
    bool feature=true;
    cout<<"Unesi broj: ";
    cin>>a;
    x=a;
    while (x!=0)
    {
        cout<<"Unesi broj: ";
        cin>>x;
        c++;
        if (x!=0 && c%3==0 && a!=x) feature=false;
    }
    if (feature) cout<<"Niz ima svojstvo!\n";
    else cout<<"Niz nema svojstvo!\n";
    return 0;
}
```

43. sumDigits.cpp

```
#include<iostream>
using namespace std;

int main()
{
    int n, suma=0;
    cout<<"Unesi broj: ";
    cin>>n;
    while (n>0)
    {
        suma+=n%10;
        n/=10;
    }
}
```

```
        cout<<"Suma znamenki je : "<<suma<<endl;
        return 0;
    }
```

47. randomNum.cpp

```
#include<iostream>
#include<ctime>
#include<cstdlib>
using namespace std;

int main()
{
    srand(static_cast<unsigned int>(time(NULL)));
    int x, y;
    x=rand()%100+1;
    do
    {
        cout<<"Pokusaj pogoditi broj u intervalu [1,100]: ";
        cin>>y;
        if (x<y) cout<<"Generirani broj je manji od unesenog broja!\n";
        else if (x>y) cout<<"Generirani broj je veci od unesenog broja!\n";
        else cout<<"Generirani broj je jednak unesenom broju!\n";
    }
    while (x!=y);
    return 0;
}
```

Vježba 6

1. multiplicationTable.cpp

```
#include<iostream>
#include<iomanip>
using namespace std;

int main()
{
    for (int i=1; i<=10; i++)
    {
        for (int j=1; j<=10; j++)
            cout<<setw(5)<<i*j;
        cout<<endl;
    }
    return 0;
}
```

2. matrix.cpp

```
#include<iostream>
#include<iomanip>
using namespace std;

int main()
{
    int n, a=1;
    cout<<"Unesi broj: ";
    cin>>n;
    if (n>2 && n<10)
    {
        for (int i=0; i<n; i++)
        {
            for (int j=0; j<n; j++)
            {
                if (i==j)
                {
                    cout<<setw(5)<<a;
                    a*=-1;
                }
                else if (i<j) cout<<setw(5)<<'x';
                else cout<<setw(5)<<'0';
            }
            cout<<endl;
        }
    }
    return 0;
}
```

10. matrixFunction.cpp

```
#include<iostream>
#include<iomanip>
using namespace std;

int main()
{
    double x, f;
    char odg;
    int c1=0, c2=0, c3=0, c4=0;
    do
    {
        cout<<" Unesi x: ";
        cin>>x;
        if (x>=1 && x<=20)
        {
            f=x*10;
            c1++;
        }
        else if (x>20 && x<=40)
        {
            f=x+2;
            c2++;
        }
        else if (x>40 && x<=60)
        {
            f=x/3;
            c3++;
        }
        else
        {
            f=0;
            c4++;
        }
        cout<<" f( "<<x<<" )="<<f<<endl;
        cout<<" Zelis li ponoviti izracun: ";
        cin>>odg;
    }
    while (odg=='d' || odg=='D');
    cout<<"Mnozenje="<<c1<<"\nZbrajanje="<<c2<<"\nDijeljenje="<<c3<<"\nDodjela="<<c4<<endl;
    for (int i=0; i<c1+c2+c3+c4; i++)
    {
        for (int j=0; j<c1+c2+c3+c4; j++)
            cout<<setw(5)<<'*';
        cout<<endl;
    }
    return 0;
}
```

}



Vježba 7

14. categorizeElements.cpp

```
#include<iostream>
#include<iomanip>
#include<cstring>
#include<cctype>
using namespace std;

int main()
{
    float polje[40], poljeNew[40];
    int n, count=0;
    do
    {
        cout<<"Unesi broj elemenata polja u intervalu (0,40]: ";
        cin>>n;
    }
    while (n<=0 || n>40);
    for (int i=0; i<n; i++)
    {
        cout<<i<<" . element = ";
        cin>>polje[i];
        if (polje[i]==0) count++;
    }
    int i=0;
    while (count>0)
    {
        poljeNew[i]=0;
        i++;
        count--;
    }
    for (int j=0; j<n; j++)
    {
        if (polje[j]<0)
        {
            poljeNew[i]=polje[j];
            i++;
        }
    }
    for (int j=0; j<n; j++)
    {
        if (polje[j]>0)
        {
            poljeNew[i]=polje[j];
            i++;
        }
    }
}
```

```

    for (int j=0; j<n; j++)
    {
        cout<<setw(5)<<polje[j];
    }
    cout<<endl;
    for (int j=0; j<n; j++)
    {
        cout<<setw(5)<<poljeNew[j];
    }
    cout<<endl;
    return 0;
}

```

30. countFrequency.cpp

```

#include<iostream>
using namespace std;

int main()
{
    char rec[200], odg;
    int v=0, m=0, d=0;
    do
    {
        cout<<"Unesi recenicu: ";
        cin.get(rec, sizeof(rec));
        cout<<"Recenica: "<<rec<<endl;
        for (unsigned int i=0; i<strlen(rec); i++)
        {
            if (isupper(rec[i])) v++;
            else if (islower(rec[i])) m++;
            else if (isdigit(rec[i])) d++;
        }
        cout<<"Velika slova: "<<v<<endl;
        cout<<"Mala slova: "<<m<<endl;
        cout<<"Znamenke: "<<d<<endl;
        cout<<"Zelis li ponoviti izracun: ";
        cin>>odg;
        cin.ignore();
    }
    while (odg=='d' || odg=='D');
    return 0;
}

```

1. temperatures.cpp

91

```

double s=0;
int m, i;
char g[15];
switch (choice)
{
case '1':
    cout<<" Unesite mjesece: ";
    cin>>m;
    for (i=0; i<NUM; i++)
    {
        s+=temp[i][m-1];
    }
    cout<<" Prosjecna temperatura za "
        mjesec<<m<<" iznosi "<<s/NUM
        <<endl;
    break;
case '2':
    cout<<" Unesite grad: ";
    cin.get(g, sizeof(g));
    for (i=0; i<NUM; i++)
    {
        if (strcmp(g, city[i])
            ==0) break;
    }
    for (int j=0; j<12; j++)
    {
        s+=temp[i][j];
    }
    cout<<" Prosjecna god. temperatura
        za grad "<<g<<" iznosi "<<s
        /12<<endl;
    break;
case '3':
    cout<<" Unesite grad: ";
    cin.get(g, sizeof(g));
    cout<<" Unesite mjesece: ";
    cin>>m;
    for (i=0; i<NUM; i++)
    {
        if (strcmp(g, city[i])
            ==0) break;
    }
    cout<<" Temperatura za grad "
        "u "<<m<<" mjesece iznosi "
        <<temp[i][m-1]<<endl;
    break;
case '4':
    for (int i=0; i<NUM; i++)
    {
        for (int j=0; j<12; j++)

```

```
                {
                    s+=temp[i][j];
                }
            }
            cout<<"Prosjecna temperatura u
                Hrvatskoj za godinu "<<god<<"
                iznosi "<<s/(NUM*12)<<endl;
            break;
        case '5':
            cout<<"Odabrali ste kraj
                interakcije!\n";
            break;
        default:
            cout<<"Krivi odabir!\n";
        }

    }
    while (choice=='1' || choice=='2' || choice=='3'
        || choice=='4');
    return 0;
}
```

Vježba 9

2. compareStudents.cpp

```
#include<iostream>
using namespace std;

struct student
{
    char name[50], surname[50], index[10];
    int grade;
};

int main()
{
    student a, b;
    cout<<"Unesi podatke 1. studentu->\nIme: ";
    cin.getline(a.name, sizeof(a.name));
    cout<<"Prezime: ";
    cin.getline(a.surname, sizeof(a.surname));
    cout<<"Index: ";
    cin.getline(a.index, sizeof(a.index));
    cout<<"Ocjena: ";
    cin>>a.grade;
    cin.ignore();
    cout<<"\n\nUnesi podatke 2. studentu->\nIme: ";
    cin.getline(b.name, sizeof(b.name));
    cout<<"Prezime: ";
    cin.getline(b.surname, sizeof(b.surname));
    cout<<"Index: ";
    cin.getline(b.index, sizeof(b.index));
    cout<<"Ocjena: ";
    cin>>b.grade;
    if (a.grade>b.grade) cout<<"Bolju ocjenu ima " <<a
        .surname<<endl;
    else if (a.grade<b.grade) cout<<"Bolju ocjenu ima "
        <<b.surname<<endl;
    else cout<<"Studenti imaju jednaku ocjenu!\n";
    return 0;
}
```

3. students.cpp

```
#include<iostream>
using namespace std;

struct student
{
    char name[50], surname[50], index[10];
```

```

        int grade;
    }p[80];

    int main()
    {
        int n, q;
        cout<<"Unesi broj upisanih studenata: ";
        cin>>n;
        for (int i=0; i<n; i++)
        {
            cin.ignore();
            cout<<"Unesi podatke o " <<i+1<<" .
                studentu -> \nIme: ";
            cin.getline(p[i].name, sizeof(p[i].name))
                ;
            cout<<"Prezime: ";
            cin.getline(p[i].surname, sizeof(p[i].
                surname));
            cout<<"Index: ";
            cin.getline(p[i].index, sizeof(p[i].index
                ));
            cout<<"Ocjena: ";
            cin>>p[i].grade;
        }
        cout<<"Unesi ocjenu za koju zelis dobiti popis
            studenata: ";
        cin>>q;
        for (int i=0; i<n; i++)
        {
            if (p[i].grade==q) cout<<p[i].surname<<"
                "<<p[i].name<<endl;
        }
        return 0;
    }

```

4. account.cpp

```

#include<iostream>
#include<cstring>
using namespace std;

struct account
{
    char name[50], surname[50], password[50];
};

int main()
{
    account I;
    char a;

```

```
    cout<<"Ime: ";
    cin.getline(I.name, sizeof(I.name));
    cout<<"Prezime: ";
    cin.getline(I.surname, sizeof(I.surname));
    a=tolower(I.name[0]);
    if (a=='o') a='0';
    else if (a=='i') a='1';
    I.password[0]=a;
    for (int i=0; i<=strlen(I.surname); i++)
    {
        a=tolower(I.surname[i]);
        if (a=='o') a='0';
        else if (a=='i') a='1';
        I.password[i+1]=a;
    }
    cout<<"Lozinka: "<<I.password<<endl;
    return 0;
}
```

Vježba 10

1. squaredNum.cpp

```
#include<iostream>
using namespace std;

int sqr(int n);

int main()
{
    int n;
    cout<<"Unesi broj: ";
    cin>>n;
    cout<<n<<"^2="<<sqr(n)<<endl;
    return 0;
}

int sqr(int n)
{
    return n*n;
}
```

2. sqrAbsNum.cpp

```
#include<iostream>
using namespace std;

int sqr(int n);
int absVal(int n);

int main()
{
    int n;
    cout<<"Unesi broj: ";
    cin>>n;
    cout<<n<<"^2="<<sqr(n)<<endl;
    cout<<' | '<<n<<"|="<<absVal(n)<<endl;
    return 0;
}

int sqr(int n)
{
    return n*n;
}

int absVal(int n)
{
    return n>0 ? n : -n;
}
```

11. rectangle.cpp

```
#include<iostream>
using namespace std;

int f1(int a, int b);
void f2(int a, int b, int &p);

int main()
{
    int a, b, p;
    cout<<"a=";
    cin>>a;
    cout<<"b=";
    cin>>b;
    p=f1(a,b);
    cout<<"Povrsina pravokutnika izracunata u
           funkciji koja vraca vrijednost preko imena
           iznosi "<<p<<endl;
    f2(a,b,p);
    cout<<"Povrsina pravokutnika izracunata u
           funkciji koja vraca vrijednost preko reference
           iznosi "<<p<<endl;
    return 0;
}

int f1(int a, int b)
{
    return a*b;
}

void f2(int a, int b, int &p)
{
    p=a*b;
    return;
}
```

14. carsInFunction.cpp

```
#include<iostream>
using namespace std;

struct car
{
    char make[30], model[30];
    int year;
};

// void input(car &a);
car input();
```

```
void write(car a);

int main()
{
    car myCar, yourCar;
    // input(myCar);
    // input(yourCar);
    myCar=input();
    yourCar=input();
    write(myCar);
    write(yourCar);
    return 0;
}

/* void input(car &a)
{
    cout<<"Marka: ";
    cin>>a.make;
    cout<<"Tip: ";
    cin>>a.model;
    cout<<"Godina: ";
    cin>>a.year;
}*/

car input()
{
    car a;
    cout<<"Marka: ";
    cin>>a.make;
    cout<<"Tip: ";
    cin>>a.model;
    cout<<"Godina: ";
    cin>>a.year;
    return a;
}

void write(car a)
{
    cout<<"Marka: "<<a.make<<endl;
    cout<<"Tip: "<<a.model<<endl;
    cout<<"Godina: "<<a.year<<endl;
}
```

Vježba 11

2. fillArray.cpp

```
#include<iostream>
#include<iomanip>
using namespace std;

int num();
void fill(double p[], int &n);
void write(double p[], int n);
void reorder(double p[], int n);

int main()
{
    int n;
    double p[50];
    n=num();
    fill(p,n);
    write(p,n);
    reorder(p,n);
    write(p,n);
    return 0;
}

int num()
{
    int n;
    cout<<"Unesi broj elemenata polja: ";
    cin>>n;
    return n;
}

void fill(double p[], int &n)
{
    cout<<"Unesi elemente polja:\n";
    for (int i=0; i<n; i++)
    {
        double t;
        cin>>t;
        if (!cin)
        {
            cin.clear();
            while (cin.get()!='\n')
                continue;
            n=i;
        }
        else p[i]=t;
    }
}
```

```

void write(double p[], int n)
{
    for (int i=0; i<n; i++)
        cout<<setw(5)<<p[i];
    cout<<endl;
}

void reorder(double p[], int n)
{
    double t;
    int end=n-1;
    for (int i=0; i<n/2; i++, end--)
    {
        t=p[i];
        p[i]=p[end];
        p[end]=t;
    }
}

```

4. 2DinFunctions.cpp

```

#include<iostream>
#include<iomanip>
using namespace std;

int dims();
void input(int mat[][10], int n);
void write(int mat[][10], int n);
int mainDiag(int mat[][10], int n);
int sideDiag(int mat[][10], int n);

int main()
{
    int n, mat[10][10];
    n=dims();
    input(mat,n);
    write(mat,n);
    cout<<"Zbroj_ elemenata_na_ glavnoj_ dijagonali_ je_ "<<
        mainDiag(mat,n)<<endl;
    cout<<"Zbroj_ elemenata_na_ sporednoj_ dijagonali_ je_ "<<
        sideDiag(mat,n)<<endl;
    return 0;
}

int dims()
{
    int n;
    do
    {

```

```
        cout<<"Unesi_broj_redaka/stupaca: ";
        cin>>n;
    }
    while (n<=0 || n>10);
    return n;
}

void input(int mat[][10], int n)
{
    cout<<"Unesi_elemente_matrice: \n";
    for (int i=0; i<n; i++)
    {
        for (int j=0; j<n; j++)
            cin>>mat[i][j];
    }
}

void write(int mat[][10], int n)
{
    for (int i=0; i<n; i++)
    {
        for (int j=0; j<n; j++)
            cout<<setw(5)<<mat[i][j];
        cout<<endl;
    }
}

int mainDiag(int mat[][10], int n)
{
    int sum=0;
    for (int i=0; i<n; i++)
    {
        sum+=mat[i][i];
    }
    return sum;
}

int sideDiag(int mat[][10], int n)
{
    int sum=0;
    for (int i=0; i<n; i++)
    {
        for (int j=0; j<n; j++)
            if (i+j==n-1) sum+=mat[i][j];
    }
    return sum;
}
```

8. webShop.cpp

```
#include<iostream>
#include<iomanip>
using namespace std;

struct product
{
    char ID[10], name[50];
    double price, quantity;
};

int num();
void input(product p[], int n);
void menu();
void end();
int choose(int lim);
void write(product p[], int n);
void writeIncome(product p[], int n);
void bestSeller(product p[], int n);

int main()
{
    int n, choice;
    product p[50];
    n=num();
    input(p,n);
    do
    {
        menu();
        choice=choose(3);
        switch (choice)
        {
            case 1:
                write(p,n);
                break;
            case 2:
                writeIncome(p,n);
                break;
            case 3:
                bestSeller(p,n);
                break;
        }
        end();
        choice=choose(2);
    }
    while (choice!=2);
    return 0;
}

int num()
```

```

{
    int n;
    do
    {
        cout<<"Unesi_broj_proizvoda:_";
        cin>>n;
    }
    while (n<=0 || n>50);
    return n;
}

void input(product p[], int n)
{
    for (int i=0; i<n; i++)
    {
        cin.ignore();
        cout<<"ID_>_";
        cin.getline(p[i].ID, sizeof(p[i].ID));
        cout<<"Naziv_proizvoda_>_";
        cin.getline(p[i].name, sizeof(p[i].name));
        do
        {
            cout<<"Cijena_>_";
            cin>>p[i].price;
        }
        while (p[i].price <=0);
        do
        {
            cout<<"Prodana_kolicina_>_";
            cin>>p[i].quantity;
        }
        while (p[i].quantity <=0);
    }
}

void menu()
{
    cout<<"\n\n*_**_*I_Z_B_O_R_N_I_K*_***\n\n";
    cout<<"1._Ispis_svih_podataka_za_sve_prodane_proizvode\n";
    cout<<"2._Ispis_ukupne_zarade_od_svih_prodanih_proizvoda\n";
    cout<<"3._Ispis_podataka_za_proizvod_(ili_svie_njih)_koji_je_ostvario_najveu_prodaju\n";
}

void end()
{
    cout<<"\n\n*_**_*I_Z_B_O_R_N_I_K*_***\n\n";
    cout<<"1._Povratak_na_izbornik\n";
}

```

```

    cout<<" 2. Izlaz iz programa\n";
}

int choose(int lim)
{
    int n;
    do
    {
        cout<<"\n\nOdabir_> ";
        cin>>n;
    }
    while (n<1 || n>lim);
    return n;
}

void write(product p[], int n)
{
    for (int i=0; i<n; i++)
    {
        cout<<"ID_> " <<setw(10)<<p[i].ID<<endl;
        cout<<"Naziv_> " <<setw(10)<<p[i].name<<endl;
        cout<<"Cijena_> " <<setw(10)<<p[i].price<<endl;
        cout<<"Prodana_kolicina_> " <<setw(10)<<p[i].quantity<<endl;
    }
}

void writeIncome(product p[], int n)
{
    double sum=0;
    for (int i=0; i<n; i++)
    {
        sum+=p[i].price*p[i].quantity;
    }
    cout<<"Dohodak_> " <<sum<<endl;
}

void bestSeller(product p[], int n)
{
    double sum=0, max=0;
    int index;
    for (int i=0; i<n; i++)
    {
        sum=p[i].price*p[i].quantity;
        if (sum>max)
        {
            max=sum;

```

```
        index=i ;  
    }  
}  
cout<<" Bestseller -> "<<p [ index ] . name<<endl ;  
}
```

Vježba 12

5. ascend.cpp

```
#include<iostream>
#include<iomanip>
using namespace std;

bool ascend(int, int);
bool ascend(int, int, int);

int main()
{
    cout<<boolalpha<<ascend(1,2)<<endl;
    cout<<ascend(3,1,2)<<endl;
    return 0;
}

bool ascend(int a, int b)
{
    return a<b ? true : false;
}

bool ascend(int a, int b, int c)
{
    return a<b && b<c ? true : false;
}
```

7. squared.cpp

```
#include<iostream>
using namespace std;

long pow(int);
double pow(double);

int main()
{
    cout<<"5^2="<<pow(5)<<endl;
    cout<<"2.5^2="<<pow(2.5)<<endl;
    return 0;
}

long pow(int n)
{
    return n*n;
}

double pow(double n)
{

```

```
        return n*n;
    }
```

8. write.cpp

```
#include<iostream>
using namespace std;

void writeArray(char s[], int n=1);

int main()
{
    char sentence[] = "Naucio sam osnove programiranja!";
    cout<<"Poziv s jednim parametrom ... \n";
    writeArray(sentence);
    cout<<"Poziv s dva parametra ... \n";
    writeArray(sentence, 5);
    return 0;
}

void writeArray(char s[], int n)
{
    for (int i=0; i<n; i++)
        cout<<s[i]<<endl;
}
```

10. findFileMinMax.cpp

```
#include<iostream>
#include<fstream>
using namespace std;

int main()
{
    ifstream input;
    input.open("numbers.txt");
    int n, min, max;
    if (input>>min)
    {
        max=min;
        while (input>>n)
        {
            if (n<min) min=n;
            else if (n>max) max=n;
        }
        cout<<"min="<<min<<endl<<"max="<<max<<endl;
    }
    else cout<<"Datoteka je prazna!\n";
}
```

```
        input.close();  
        return 0;  
    }
```

Vježba 13

4. writeInFile.cpp

```
#include<iostream>
#include<fstream>
#include<cstdlib>
using namespace std;

int add(ifstream &input);

int main()
{
    ofstream output;
    ifstream input;
    int n, s=0;
    char name[50];
    cout<<"Ime datoteke: ";
    cin>>name;
    output.open(name, ios::app);
    if (output.fail())
    {
        cout<<"Izlazna datoteka nije kreirana!\n";
        exit(1);
    }
    cout<<"Unesi niz brojeva - oznaka za kraj je 0 ->
        \n";
    while (cin>>n && n!=0)
    {
        output<<n<<" ";
    }
    output.close();
    input.open(name);
    if (input.fail())
    {
        cout<<"Ulazna datoteka nije kreirana!\n";
        exit(1);
    }
    cout<<"Suma = " << add(input) << endl;
    input.close();
    return 0;
}

int add(ifstream &input)
{
    int s=0, n;
    while (input>>n)
        s+=n;
    return s;
}
```

```
}
```

8. copyFileContents.cpp

```
#include<iostream>
#include<cstdlib>
#include<fstream>
using namespace std;

int main()
{
    ifstream input;
    ofstream output;
    char inputFile[50], outputFile[50], line[255];
    cout<<"Ime ulazne datoteke: ";
    cin>>inputFile;
    input.open(inputFile);
    if (input.fail())
    {
        cout<<"Izlazna datoteka nije kreirana!\n";
        exit(1);
    }
    cout<<"Ime izlazne datoteke: ";
    cin>>outputFile;
    output.open(outputFile);
    if (output.fail())
    {
        cout<<"Izlazna datoteka nije kreirana!\n";
        exit(1);
    }
    while (!input.eof())
    {
        input.getline(line, sizeof(line));
        output<<line<<endl;
    }
    input.close();
    output.close();
    return 0;
}
```

16. studentRecords.cpp

```
#include<iostream>
#include<cstdlib>
#include<fstream>
using namespace std;

struct student
```

```
{
    char name[50], surname[50];
    long id;
};

void write(ofstream &output, int n);

int main()
{
    int n;
    do
    {
        cout<<"Unesi broj studenata: ";
        cin>>n;
    }
    while(n<0 || n>50);
    ofstream output;
    output.open("studenti.txt");
    if (output.fail())
    {
        cout<<"Izlazna datoteka nije kreirana!\n";
        exit(1);
    }
    write(output,n);
    output.close();
    return 0;
}

void write(ofstream &output, int n)
{
    student x;
    for (int i=0; i<n; i++)
    {
        cin.ignore();
        cout<<"Ime-> ";
        cin.getline(x.name, sizeof(x.name));
        cout<<"Prezime-> ";
        cin.getline(x.surname, sizeof(x.surname));
        cout<<"Id-> ";
        cin>>x.id;
        output<<x.id<<"- "<<x.surname<<" , "<<x.name<<endl;
    }
}
```

Kolokviji i ispiti

1. exam_1_1.cpp

```
#include <iostream>
using namespace std;

int main()
{
    int izbor, broj, prva, zadnja;
    bool parne;
    cout<<"Odaberite opciju:\n1. Pokreni interakciju \
n2. Ispisi pozdravnu poruku\n3. Prekini \
izvođenje"<<endl;
    cin>>izbor;
    switch (izbor)
    {
        case 1:
            cout<<"unesi petero znamenkasti broj: ";
            cin>>broj;
            prva=broj/10000;
            zadnja=broj%10;
            parne=(prva%2==0) && (zadnja%2==0);
            if (parne!=0)
                cout<<"Prva i zadnja znamenka su \
parni brojevi\n";
            break;
        case 2:
            cout<<"Dobrodošli u program!\n";
    }
    return 0;
}
```

2. exam_1_2.cpp

```
#include <iostream>
using namespace std;

int main()
{
    int broj;
    char izbor;
    bool log;
    cout<<"Unesite broj: ";
    cin>>broj;
    cout<<"(a) Ispisi broj\n(b) Ispisi apsolutnu \
vrijednost broja\n(c) Prekini izvođenje \
programa\n";
```

```

        cout<<" Odaberite opciju a, b ili c.\n";
        cin>>izbor;
        switch ( izbor )
        {
            case 'a':
                cout<<" Unesen je broj: " << broj << endl;
                break;
            case 'b':
                log= broj > 0;
                if ( log )
                    cout<<" Uneseni broj je pozitivan .\n" << endl;

                else
                {
                    broj=broj*(-1);
                    cout<<" Apsolutna vrijednost unesenog broja je: " << broj << endl;
                }
                break;
        }
        return 0;
    }
}

```

3. exam_1_3.cpp

```

#include<iostream>
using namespace std;

int main ()
{
    const int b=20, c=80;
    int n, a;
    bool log;
    cout<<"\nOdaberite opciju 1-3:\n";
    cout<<"\n1. Pokreni igru!\n";
    cout<<" 2. Unesi broj!\n";
    cout<<" 3. Prekini igru!\n";
    cin>>n;
    switch (n)
    {
        case 1:
            cout<<"Da vidimo kako te idu grešne srecu!\n";
            break;
        case 2:
            cout<<"Unesi broj koji nije u intervalu [20-80]!\n";
            cin>>a;
            log = (a<b || a>c);

```

```
        if (log==0)
            cout<<"Ni blizu!\n";
        break;
    }
    return 0;
}
```

4. exam_1_4.cpp

```
#include<iostream>
#include<iomanip>
using namespace std;

int main()
{
    int opcija = 0;
    bool logic;
    cout<<"Odaberite opciju:\n1. Unesi brojeve\n2. Unesi slova\n3. Prekini izvodjenje programa\n";
    ;
    cin>>opcija;
    switch (opcija)
    {
        case 1:
            cout<<"Unesi 3 realna broja:\n";
            double broj1, broj2, broj3;
            cin>>broj1>>broj2>>broj3;
            logic=(broj1<broj2)&&(broj2<broj3)
                &&(broj1>=1&&broj1<=100)&&(broj2>=1&&broj2<=100)&&(broj3>=1&&broj3<=100);
            if (logic)
                cout <<"Brojevi zadovoljavaju uvjet!"
                <<endl;
            break;
        case 2:
            cout<<"Unesi 3 slova:\n";
            char slovo1, slovo2, slovo3;
            cin>>slovo1>>slovo2>>slovo3;
            logic=(slovo1<slovo2)&&(slovo2<slovo3);
            cout<<boolalpha<<logic<<endl;
            break;
    }
    return 0;
}
```

5. exam_1_5.cpp

```
#include<iostream>
using namespace std;

int main()
{
    int d, m, g, razlika;
    char odabir;
    bool mirovina;
    cout<<"Unesite datum rođenja: "<<endl;
    cout<<"-dan: ";
    cin>>d;
    cout<<"-mjesec: ";
    cin>>m;
    cout<<"-godina: ";
    cin>>g;
    mirovina = g<1946 || (g==1946 && m<11) || (d<17
        && m==11 && g==1946);
    cout<<"IZBORNIK"<<endl;
    cout<<"Unesite: "<<endl;
    cout<<"1 ili a- Uvjet za mirovinu "<<endl;
    ;
    cout<<"2 ili b- Godina do mirovine "<<
        endl;
    cout<<"3 ili c- Prekini izvođenje
        programa"<<endl;
    cout<<"Vas odabir: ";
    cin>>odabir;
    switch(odabir)
    {
        case 'a':
        case '1':
            if(mirovina)
                cout<<"Osoba ima uvjet za
                    mirovinu. "<<endl;
            else
                cout<<"Osoba nema uvjet
                    za mirovinu. "<<endl;
            break;
        case 'b':
        case '2':
            if (mirovina)
                cout<<"Zasto Vi jos
                    uvijek radite?"<<endl;
            else
            {
                razlika=g-1946;
                if(razlika==0)
                    cout<<"U mirovinu
                        mozete ove
```

```

        godine.<<endl
        ;
    else
        cout<<"Do
        mirovine
        _jos_<<
        razlika<<"_
        godin-u/-e/-a\
        n";
    }
    return 0;
}

```

5. exam_2_1.cpp

```

#include<iostream>
#include<iomanip>
using namespace std;

const int MAX=20;

void input(int mat[][MAX], int &m, int &n);
void write(int mat[][MAX], int m, int n);
bool condition(int m1, int m2, int n1, int n2);
void add(int mat1[][MAX], int mat2[][MAX], int mat[][MAX],
        int m, int n);

int main()
{
    int m1, m2, n1, n2, A[MAX][MAX], B[MAX][MAX], C[
        MAX][MAX];
    char choice;
    do
    {
        input(A, m1, n1);
        input(B, m2, n2);
        write(A, m1, n1);
        write(B, m2, n2);
        if (condition(m1, m2, n1, n2))
        {
            add(A, B, C, m1, n1);
            write(C, m1, n1);
        }
        else cout<<"Matrice_se_ne_mogu_zbrajati!\n";
    }
}

```

```

        cout<<"Zelite li zavrsiti izvodjenje
        programa(d/n)? ";
        cin>>choice;
    }
    while (choice=='n' || choice=='N');
    return 0;
}

void input(int mat[][MAX], int &m, int &n)
{
    do
    {
        cout<<"Broj redaka: ";
        cin>>m;
    }
    while (m<=0 || m>MAX);
    do
    {
        cout<<"Broj stupaca: ";
        cin>>n;
    }
    while (n<=0 || n>MAX);
    cout<<"Unesi elemente matrice "<<m<<"x"<<n<<"
    ->\n";
    for (int i=0; i<m; i++)
    {
        for (int j=0; j<n; j++)
            cin>>mat[i][j];
    }
}

void write(int mat[][MAX], int m, int n)
{
    cout<<"\n\nElemente matrice "<<m<<"x"<<n<<"
    ->\n";
    for (int i=0; i<m; i++)
    {
        for (int j=0; j<n; j++)
            cout<<setw(2)<<mat[i][j]<<setw(2)
            <<'*';
        cout<<endl;
    }
    cout<<endl<<endl;
}

bool condition(int a1, int a2, int b1, int b2)
{
    return (a1==a2 && b1==b2) ? true : false;
}

```

```

void add(int mat1[][MAX], int mat2[][MAX], int mat[][MAX]
        ], int m, int n)
{
    for (int i=0; i<m; i++)
    {
        for (int j=0; j<n; j++)
            mat[i][j]=mat1[i][j]+mat2[i][j];
    }
}

```

6. exam_2_2.cpp

```

#include<iostream>
#include<iomanip>
using namespace std;

const int MAX=20;

void input(int mat[][MAX], int &m, int &n);
void write(int mat[][MAX], int m, int n);
char condition(int m2, int n1);
void multiply(int mat1[][MAX], int mat2[][MAX], int mat
            [][MAX], int m1, int m2, int n2);

int main()
{
    int m1, m2, n1, n2, A[MAX][MAX], B[MAX][MAX], C[
        MAX][MAX];
    char choice, log;
    do
    {
        input(A, m1, n1);
        input(B, m2, n2);
        log=condition(m2, n1);
        write(A, m1, n1);
        write(B, m2, n2);
        if (log=='d')
        {
            for (int i=0; i<m1; i++)
            {
                for (int j=0; j<n2; j++)
                    C[i][j]=0;
            }
            multiply(A, B, C, m1, m2, n2);
            write(C, m1, n2);
        }
        else cout<<"Matrice se ne mogu mnoziti!\n
            ";
        cout<<"Zelite li završiti izvođenje
            programa (d/n)? ";
    }
}

```

```

        cin>>choice;
    }
    while (choice!='d' && choice!='D');
    return 0;
}

void input(int mat[][MAX], int &m, int &n)
{
    do
    {
        cout<<"Broj redaka: ";
        cin>>m;
    }
    while (m<=0 || m>MAX);
    do
    {
        cout<<"Broj stupaca: ";
        cin>>n;
    }
    while (n<=0 || n>MAX);
    cout<<"Unesi elemente matrice "<<m<<"x"<<n<<"
        ->\n";
    for (int i=0; i<m; i++)
    {
        for (int j=0; j<n; j++)
            cin>>mat[i][j];
    }
}

void write(int mat[][MAX], int m, int n)
{
    cout<<"\n\nElemente matrice "<<m<<"x"<<n<<"
        ->\n";
    for (int i=0; i<m; i++)
    {
        for (int j=0; j<n; j++)
            cout<<setw(5)<<mat[i][j];
        cout<<endl;
    }
    cout<<endl<<endl;
}

char condition(int m2, int n1)
{
    char cond;
    if (m2==n1)
    {
        cond='d';
    }
    else cond='n';
}

```

```

        return cond;
    }

    void multiply(int mat1[][MAX], int mat2[][MAX], int mat
       [][MAX], int m1, int m2, int b2)
    {
        for (int i=0; i<m1; i++)
        {
            for (int j=0; j<b2; j++)
            {
                for (int k=0; k<m2; k++)
                    mat[i][j]+=mat1[i][k]*
                    mat2[k][j];
            }
        }
    }
}

```

7. exam_2_3.cpp

```

#include<iostream>
#include<iomanip>
using namespace std;

const int MAX=20;

void input(int mat[][MAX], int m, int n);
void writeT(int mat[][MAX], int m, int n);
char answer();
void translate(int mat1[][MAX], int mat2[][MAX], int m,
    int n);

int main()
{
    int m, n, A[MAX][MAX], B[MAX][MAX];
    char choice;
    do
    {
        do
        {
            cout<<"Broj redaka: ";
            cin>>m;
        }
        while (m<=0 || m>MAX);
        do
        {
            cout<<"Broj stupaca: ";
            cin>>n;
        }
        while (n<=0 || n>MAX);
        input(A, m, n);
    }
}

```

```

        writeT(A, m, n);
        translate(A, B, m, n);
        writeT(B, n, m);
        choice=answer();
    }
    while (choice=='n' || choice=='N');
    return 0;
}

void input(int mat[][MAX], int m, int n)
{
    cout<<"Unesi elemente matrice "<<m<<"x"<<n<<"
    ->\n";
    for (int i=0; i<m; i++)
    {
        for (int j=0; j<n; j++)
            cin>>mat[i][j];
    }
}

void writeT(int mat[][MAX], int m, int n)
{
    cout<<"\n\nElementi matrice "<<n<<"x"<<m<<"->\n";
    for (int i=0; i<n; i++)
    {
        for (int j=0; j<m; j++)
            cout<<setw(5)<<mat[j][i];
        cout<<endl;
    }
    cout<<endl<<endl;
}

void translate(int mat1[][MAX], int mat2[][MAX], int m,
               int n)
{
    for (int i=0; i<n; i++)
    {
        for (int j=0; j<m; j++)
            mat2[i][j]=mat1[j][i];
    }
}

char answer()
{
    char choice;
    cout<<"Zelite li zavrsiti izvodjenje programa (d/
    n)? ";
    cin>>choice;
    return choice;
}

```

}

Literatura

- [1] Savitch, W.: Problem Solving in C++. Pearson Publishing, 2006.