

# PROGRAMIRANJE

ZBIRKA ZADATAKA

Ishod učenja 1

2023-2024.

## Zadatak 1

Napišite program koji od korisnika učitava podatke o dva stošca: polumjer baze (r) i visinu (h). Neka program izračuna volumene obaju stožaca koristeći formulu:  $V = \frac{1}{3}\pi r^2 h$ . Ispišite oba volumena i podatak o tome koji stožac ima veći volumen. Primjer rada programa:

```
Upisite polumjer baze prvog stosca: 0.9
Upisite visinu prvog stosca: 5.5
Upisite polumjer baze drugog stosca: 5.5
Upisite visinu drugog stosca: 0.9
Prvi stozac ima volumen 4.6629
Drugi stozac ima volumen 28.4955
Drugi stozac ima veci volumen
```

*Moguće rješenje:*

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

int main() {
    double pi = 3.14;

    double r1;
    cout << "Upisite polumjer baze prvog stosca: ";
    cin >> r1;

    double h1;
    cout << "Upisite visinu prvog stosca: ";
    cin >> h1;

    double r2;
    cout << "Upisite polumjer baze drugog stosca: ";
    cin >> r2;

    double h2;
    cout << "Upisite visinu drugog stosca: ";
    cin >> h2;

    double v1 = (1.0 / 3.0) * pi * pow(r1, 2) * h1;
    double v2 = (1.0 / 3.0) * pi * pow(r2, 2) * h2;

    cout << "Prvi stozac ima volumen " << v1 << endl;
    cout << "Drugi stozac ima volumen " << v2 << endl;
    if (v1 > v2) {
        cout << "Prvi stozac ima veci volumen" << endl;
    }
    else if (v1 < v2) {
        cout << "Drugi stozac ima veci volumen" << endl;
    }
    else {
        cout << "Stosci imaju jednake volumene" << endl;
    }

    return 0;
}
```

## Zadatak 2

Napišite program koji od korisnika učitava x i y koordinate neke točke. Pretpostavite da postoji pravokutnik čije su x i y koordinate lijevog gornjeg kuta jednak 10, 10 te čija je širina 8 i visina 4. Izračunajte i ispišite nalazi li se točka u koliziji s pravokutnikom ili ne. Točka se nalazi u koliziji s pravokutnikom ako su ispunjeni svi sljedeći uvjeti:

- x pozicija točke je desno od lijevog ruba pravokutnika.
- x pozicija točke je lijevo od desnog ruba pravokutnika.
- y pozicija točke je ispod gornjeg ruba pravokutnika.
- y pozicija točke je iznad donjeg ruba pravokutnika.

Pretpostavite da je ishodište 0, 0 u gornjem lijevom kutu.

*Moguće rješenje:*

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

int main() {
    double x;
    cout << "Upisite x: ";
    cin >> x;

    double y;
    cout << "Upisite y: ";
    cin >> y;

    double px = 10.0;
    double py = 10.0;
    double pw = 8.0;
    double ph = 4.0;

    if (x >= px and x <= px + pw
        and y >= py and y <= py + ph) {
        cout << "Postoji kolizija" << endl;
    }
    else {
        cout << "Nema kolizije" << endl;
    }

    return 0;
}
```

### Zadatak 3

Napišite program koji od korisnika učitava broj kartice i PIN. Ispišite mu stanje računa, pitajte ga koliko novca želi podići, ispišite da ste mu isplatili traženi novac (ili da nema dovoljno) te mu na kraju ispišite preostalo stanje računa. Ispravni brojevi kartica i PIN-ovi su:

- Ako je broj kartice 5399-8500-1722-0325 i PIN 1234, onda korisnik ima na računu 350.50 eura,
- Ako je broj kartice 5399-1159-2036-1151 i PIN 0752, onda korisnik ima na računu 9510.00 eura.

Ako korisnik unese neispravan broj kartice ili PIN, ispišite da ste mu uzeli karticu, da se javi u poslovnicu banke i završite program. Primjer rada programa kad korisnik ima dovoljno sredstava:

Upisite broj kartice: 5399-8500-1722-0325

Upisite PIN: 1234

Vase trenutno stanje je: 350.5 EUR

Koliko novca zelite podici: 200

Isplaceno vam je 200 EUR

Vase trenutno stanje je: 150.5 EUR

Primjer rada programa kad korisnik nema dovoljno sredstava:

Upisite broj kartice: 5399-1159-2036-1151

Upisite PIN: 0752

Vase trenutno stanje je: 9510 EUR

Koliko novca zelite podici: 10000

Nema dovoljno sredstava na racunu za trazenu isplatu

Vase trenutno stanje je: 9510 EUR

Primjer rada programa kad korisnik upiše krive podatke:

Upisite broj kartice: 5399-1159-2036-1151

Upisite PIN: 9999

Neispravan broj kartice ili PIN, vasa kartica je zadržana. Molimo, javite se u poslovnicu banke.

*Moguće rješenje:*

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

int main() {
    string broj_kartice;
    cout << "Upisite broj kartice: ";
    getline(cin, broj_kartice);

    string pin;
    cout << "Upisite PIN: ";
    getline(cin, pin);

    double stanje;

    if (broj_kartice == "5399-8500-1722-0325" && pin == "1234") {
        stanje = 350.50;
    }
    else if (broj_kartice == "5399-1159-2036-1151" && pin == "0752") {
        stanje = 9510.00;
    }
    else {
```

```
        cout << "Neispravan broj kartice ili PIN, vasa kartica je zadržana.  
Molimo, javite se u poslovnicu banke." << endl;  
        return 0; // Završimo program  
    }  
  
    cout << "Vase trenutno stanje je: " << stanje << " EUR" << endl;  
  
    double isplata;  
    cout << "Koliko novca zelite podici: ";  
    cin >> isplata;  
  
    if (isplata <= stanje) {  
        stanje -= isplata;  
        cout << "Isplaceno vam je " << isplata << " EUR" << endl;  
    }  
    else {  
        cout << "Nema dovoljno sredstava na racunu za trazenu isplatu" << endl;  
    }  
  
    cout << "Vase trenutno stanje je: " << stanje << " EUR" << endl;  
  
    return 0;  
}
```

## Zadatak 4

Napišite igru u kojoj je korisnik udaljen 20 koraka od cilja. U svakoj rundi omogućite korisniku da baci kockicu na kojoj može dobiti broj između 1 i 6 te ga pomaknite toliko polja prema cilju. Korisnik može doći do cilja samo ako dobije točan broj (primjerice, ako je udaljen 4 polja od cilja, mora dobiti 4 da dođe do cilja). Nacrtajte svaku rundu. Primjer odigravanja partije:

```
Runda: 1
Dobili ste 2
-X-----
Pritisnite enter za sljedeću rundu

Runda: 2
Dobili ste 5
-----X-----
Pritisnite enter za sljedeću rundu

Runda: 3
Dobili ste 4
-----X-----
Pritisnite enter za sljedeću rundu

Runda: 4
Dobili ste 1
-----X-----
Pritisnite enter za sljedeću rundu

Runda: 5
Dobili ste 3
-----X-----
Pritisnite enter za sljedeću rundu

Runda: 6
Dobili ste 1
-----X-----
Pritisnite enter za sljedeću rundu

Runda: 7
Dobili ste 1
-----X-----
Pritisnite enter za sljedeću rundu

Runda: 8
Dobili ste 2
-----X-----
Pritisnite enter za sljedeću rundu

Runda: 9
Dobili ste 5
Dobili ste prevelik broj, morate čekati sljedeću rundu
-----X-----
Pritisnite enter za sljedeću rundu

Runda: 10
```

```

Dobili ste 6
Dobili ste prevelik broj, morate cekati sljedecu rundu
-----X-
Pritisnite enter za sljedecu rundu

Runda: 11
Dobili ste 4
Dobili ste prevelik broj, morate cekati sljedecu rundu
-----X-
Pritisnite enter za sljedecu rundu

Runda: 12
Dobili ste 6
Dobili ste prevelik broj, morate cekati sljedecu rundu
-----X-
Pritisnite enter za sljedecu rundu

Runda: 13
Dobili ste 1
Pobjeda!

```

*Moguće rješenje:*

```

#include <iostream>
#include <string>
#include <ctime>
using namespace std;

int main() {
    srand(time(nullptr));

    int udaljenost_cilja = 50;
    int polozaj_igraca = 0;
    int runda = 1;
    string temp;
    bool pobjeda = false;

    while (true) {
        cout << "\nRunda: " << runda << endl;

        int dobiven_broj = 1 + rand() % (6 - 1 + 1); // Bacu kockicu
        cout << "Dobili ste " << dobiven_broj << endl;

        if (polozaj_igraca + dobiven_broj < udaljenost_cilja) {
            polozaj_igraca += dobiven_broj;
        }
        else if (polozaj_igraca + dobiven_broj == udaljenost_cilja) {
            cout << "Pobjeda!" << endl;
            break;
        }
        else {
            cout << "Dobili ste prevelik broj, morate cekati sljedecu rundu"
<< endl;
        }

        // Crtanje
        for (int i = 0; i < polozaj_igraca - 1; i++) { // Predjeni put

```

```
        cout << '-';
    }

    if (polozaj_igraca > 0) {
        cout << 'x'; // Polozaj igraca
    }

    for (int i = 0; i < udaljenost_cilja - polozaj_igraca; i++) { //
Preostali put
        cout << '-';
    }
    cout << endl;

    cout << "Pritisnite enter za sljedecu rundu";
    getline(cin, temp);

    runda++;
}

return 0;
}
```



## Zadatak 5

Napišite program koji korisniku omogućuje rad s tekućim računom. Omogućite korisniku sljedeće akcije, dok god to želi:

- Ispis stanja računa.
- Dodavanje novca na račun.
- Isplatu novca s računa.
- Završetak programa.

Primjer rada programa:

A = ispis stanja

B = dodavanje novca na račun

C = isplata novca s računa

D = kraj

> A

Stanje racuna: 0 EUR

> B

Unesite iznos koji zelite dodati: 200

Iznos uspjesno dodan

> B

Unesite iznos koji zelite dodati: 100

Iznos uspjesno dodan

> C

Unesite iznos koji zelite isplatiti: 500

Nedovoljno sredstava

> A

Stanje racuna: 300 EUR

> C

Unesite iznos koji zelite isplatiti: 150

Iznos uspjesno isplacen

> C

Unesite iznos koji zelite isplatiti: 21

Iznos uspjesno isplacen

> A

Stanje racuna: 129 EUR

> D

Moguće rješenje:

---

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

int main() {
    double tekuci_racun = 0.0;
    double iznos;
    char operacija;

    cout << "A = ispis stanja" << endl;
    cout << "B = dodavanje novca na racun" << endl;
    cout << "C = isplata novca s racuna" << endl;
    cout << "D = kraj" << endl;

    do {
        cout << "\n> ";
        cin >> operacija;

        switch (operacija) {
            case 'A':
                cout << "Stanje racuna: " << tekuci_racun << " EUR" << endl;
                break;
            case 'B':
                cout << "Unesite iznos koji zelite dodati: ";
                cin >> iznos;
                tekuci_racun += iznos;
                cout << "Iznos uspjesno dodan" << endl;
                break;
            case 'C':
                cout << "Unesite iznos koji zelite isplatiti: ";
                cin >> iznos;
                if (tekuci_racun < iznos) {
                    cout << "Nedovoljno sredstava" << endl;
                }
                else {
                    tekuci_racun -= iznos;
                    cout << "Iznos uspjesno isplacen" << endl;
                }
                break;
            default:
                break;
        }
    } while (operacija != 'D');

    return 0;
}
```

## Zadatak 6

Napišite program koji simulira aparat za parking. Omogućite korisniku da upiše koliko novca je ubacio te da upiše broj sati parkiranja (jedan sat je 2 EUR). Ispišite korisniku sve novčanice i kovanice koje će mu aparat vratiti, uz pretpostavku da će uvijek vratiti najmanji mogući broj (primjerice, ako treba vratiti 8 EUR onda će vratiti novčanicu od 5 EUR te kovanice od 2 i 1 EUR). Pretpostavite da aparat vraća samo novčanice od 10 EUR i 5 EUR te kovanice od 2 EUR i 1 EUR. Napomena: neka vam tipovi podataka budu int. Tko želi, može ih postaviti na double pa vidjeti s kojim problemom se tada suočavamo. Primjer rada programa:

```
Ubacite novac: 50
Koliko sati ste ostavili auto na parking: 6
Trebam vratiti: 38
Vracam novčanicu od 10 EUR
Vracam novčanicu od 10 EUR
Vracam novčanicu od 10 EUR
Vracam novčanicu od 5 EUR
Vracam kovanicu od 2 EUR
Vracam kovanicu od 1 EUR
```

*Moguće rješenje:*

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

int main() {
    int ubaceno;
    cout << "Ubacite novac: ";
    cin >> ubaceno;

    int sati;
    cout << "Koliko sati ste ostavili auto na parking: ";
    cin >> sati;

    int cijena = sati * 2;

    if (ubaceno < cijena) {
        cout << "Neispravan odabir" << endl;
        return 0;
    }

    int za_vratiti = ubaceno - cijena;
    cout << "Trebam vratiti: " << za_vratiti << endl;

    do {
        if (za_vratiti > 10) {
            cout << "Vracam novčanicu od 10 EUR" << endl;
            za_vratiti -= 10;
        }
        else if (za_vratiti > 5) {
            cout << "Vracam novčanicu od 5 EUR" << endl;
            za_vratiti -= 5;
        }
        else if (za_vratiti > 2) {
            cout << "Vracam kovanicu od 2 EUR" << endl;
            za_vratiti -= 2;
        }
    } while (za_vratiti > 0);
}
```

```
    }  
    else {  
        cout << "Vracam kovanicu od 1 EUR" << endl;  
        za_vratiti -= 1;  
    }  
} while (za_vratiti > 0);  
return 0;  
}
```

## Zadatak 7

Napišite program koji zamisli jedan cijeli broj između 1 i 100. Neka program od korisnika u petlji učitava cijeli broj i ispisuje korisniku je li zamišljeni broj veći, manji ili jednak učitanoj broju. Petlja treba trajati sve dok korisnik ne pogodi zamišljeni broj. Na kraju, program treba ispisati u koliko pokušaja je korisnik pogodio broj. Primjer rada programa:

```
Pokusajte pogoditi broj: 50
Zamisljeni broj je veci
Pokusajte pogoditi broj: 75
Zamisljeni broj je veci
Pokusajte pogoditi broj: 88
Zamisljeni broj je manji
Pokusajte pogoditi broj: 81
Zamisljeni broj je veci
Pokusajte pogoditi broj: 85
Zamisljeni broj je manji
Pokusajte pogoditi broj: 83
Zamisljeni broj je veci
Pokusajte pogoditi broj: 84
Pogodili ste zamisljeni broj iz 7 pokusaja
```

*Moguće rješenje:*

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

int main() {
    // Zamišljanje broja
    srand(time(nullptr));
    int zamisljen = 1 + (rand() % (100 - 1 + 1));

    int broj_pokusaja = 0;
    bool pogodjeno = false;
    do {
        int pokusaj;
        cout << "Pokusajte pogoditi broj: ";
        cin >> pokusaj;
        broj_pokusaja++;

        if (pokusaj > zamisljen) {
            cout << "Zamisljeni broj je manji" << endl;
        }
        else if (pokusaj < zamisljen) {
            cout << "Zamisljeni broj je veci" << endl;
        }
        else {
            cout << "Pogodili ste broj iz " << broj_pokusaja << " pokusaja" << endl;
            pogodjeno = true;
        }
    } while (!pogodjeno);

    return 0;
}
```

## Zadatak 8

Napišite program koji od korisnika učitava jedan string Z i jedan broj B. Neka program ispiše string Z točno B puta tako da između svaka dva ispisa stringa stavi crticu, osim iza zadnjeg. Primjerice, ako je Z = "MMM" i B = 3, onda treba ispisati: MMM-MMM-MMM

*Moguće rješenje:*

---

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

int main() {
    string z;
    cout << "Upisite neki string: ";
    getline(cin, z);

    int b;
    cout << "Upisite neki prirodni broj: ";
    cin >> b;

    for (int i = 0; i < b; i++) {
        cout << z;

        if (i < b - 1) {
            cout << "-";
        }
    }

    return 0;
}
```

## Zadatak 9

Napišite program koji od korisnika učitava broj N veći od 1. Neka program zvjezdicama nacrtava prazni kvadrat sa stranicama duljine N. Primjerice, ako korisnik upiše N jednak 6, treba nacrtati ovakav kvadrat:

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

*Moguće rješenje:*

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

int main() {
    int n;
    cout << "Upisite n veci od 1: ";
    cin >> n;

    for (int redak = 1; redak <= n; redak++) {
        // Svaki redak zapocinje zvjezdicom
        cout << '*';

        // Prvi i zadnji redak imaju n-2 zvjezdice, a svi ostali reci imaju razmake
        for (int i = 0; i < n - 2; i++) {
            if (redak == 1 || redak == n) {
                cout << '*';
            }
            else {
                cout << ' ';
            }
        }

        // Svaki redak zavrшава zvjezdicom
        cout << '* << endl;
    }

    return 0;
}
```

## Zadatak 10

Napišite program koji od korisnika učitava neparni broj N veći ili jednak 3. Neka program zvjezdicama nacrtaj prazni kvadrat sa stranicama duljine N, okomito podijeljen na dva jednaka dijela znakom |.

Primjer za N=3	Primjer za N=5	Primjer za N=7
*** * * ***	***** *   * *   * *   * *****	***** *   * *   * *   * *   * *   * *****

Moguće rješenje:

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

int main() {
    int n;
    cout << "Upisite neparni n veci ili jednak 3: ";
    cin >> n;

    for (int redak = 1; redak <= n; redak++) {
        cout << '*'; // Svaki redak zapocinje zvjezdicom

        // Prvi i zadnji redak imaju (n-2)/2 zvjezdica, a ostali imaju razmake
        for (int i = 0; i < (n - 2) / 2; i++) {
            if (redak == 1 || redak == n) {
                cout << '*';
            }
            else {
                cout << ' ';
            }
        }

        // Prvi i zadnji redak imaju 1 zvjezdicu, svi ostali "|".
        if (redak == 1 || redak == n) {
            cout << '*';
        }
        else {
            cout << '|';
        }

        // Ponovimo zvjezdice ili razmake, kao i za prvu polovicu
        for (int i = 0; i < (n - 2) / 2; i++) {
            if (redak == 1 || redak == n) {
                cout << '*';
            }
            else {
                cout << ' ';
            }
        }
        cout << '* << endl; // Svaki redak zavrшава zvjezdicom
    }
    return 0;
}
```



## Zadatak 11

Napišite program koji od korisnika učitava neparni broj  $N$  veći ili jednak 3. Neka program zvjezdicama nacrtá krunu sastavljenu od dva pravokutna trokuta s obje katete jednake  $N$ , okrenuta hipotenuzama jedan prema drugome. Upute:

- Prvi redak je iznimka jer ima zvjezdice na početku i kraju.
- Zadnji redak je iznimka jer ima sve zvjezdice.
- Svi ostali reci imaju po četiri zvjezdice i razmake.

Primjer za N=3	Primjer za N=5	Primjer za N=7
* * ** ** *****	* * ** ** * * * * * * * * *****	* * ** ** * * * * * * * * * * * * * * * * *****

*Moguće rješenje:*

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

int main() {
    int n;
    cout << "Upisite neparni n veci ili jednak 3: ";
    cin >> n;

    int razmaka_izmedju = 2 * n - 4; // Racunam od drugog retka
    int razmaka_unutar = 0;

    for (int redak = 1; redak <= n; redak++) {
        if (redak == 1) { // Prvi redak je iznimka
            cout << '*';
            for (int i = 0; i < 2 * n - 2; i++) {
                cout << ' ';
            }
            cout << '*';
        }
        else if (redak == n) { // Zadnji redak je iznimka
            for (int i = 0; i < 2 * n; i++) {
                cout << '*';
            }
        }
        else { // Svi ostali reci imaju jednak oblik
            cout << '*'; // Prvo zvjezdica

            for (int i = 0; i < razmaka_unutar; i++) { // Unutarnji razmaci
                cout << ' ';
            }

            cout << '*'; // Opet zvjezdica

            for (int i = 0; i < razmaka_izmedju; i++) { // Vanjski razmaci
                cout << ' ';
            }
        }
    }
}
```

```
    }  
    cout << '*'; // Opet zvjezdica  
    for (int i = 0; i < razmaka_unutar; i++) { // Unutarnji razmaci  
        cout << ' ';  
    }  
    cout << '*'; // Kraj  
    razmaka_unutar++;  
    razmaka_izmedju -= 2;  
    }  
    cout << endl;  
    }  
    return 0;  
}
```

## Zadatak 12

Napišite program koji od korisnika učitava broj N i zvjezdicama nacrtava dijamant visine  $2*N - 1$ . Prvi i zadnji redi su iznimke jer imaju samo jednu zvjezdicu. Svi ostali redi se crtaju po istom principu.

Primjer za N=3	Primjer za N=4	Primjer za N=5
<pre>*  * * *   *  * *   *</pre>	<pre>  *  * * *   *  *   *   * *    *</pre>	<pre>    *    * *   *   *  *     * *       *  *     *   *   *    * *     *</pre>

Moguće rješenje:

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

int main() {
    int n;
    cout << "Upisite n: ";
    cin >> n;

    int razmaka_ispred = n - 1;
    int razmaka_unutar = -1;

    for (int redak = 1; redak <= 2 * n - 1; redak++) {

        // Uvijek zapocinjemo s razmacima
        for (int i = 0; i < razmaka_ispred; i++) {
            cout << ' ';
        }

        // Uvijek ide prva zvjezdica
        cout << '*';

        if (redak > 1 && redak < 2 * n - 1) { // Prvi i zadnji red su iznimka

            for (int i = 0; i < razmaka_unutar; i++) {
                cout << ' ';
            }

            cout << '*';
        }

        cout << endl;

        if (redak < n) {
            razmaka_ispred--;
            razmaka_unutar += 2;
        }
        else {
            razmaka_ispred++;
        }
    }
}
```

```
        razmaka_unutar -= 2;
    }
}
return 0;
```

## Zadatak 13

U prirodi, s obzirom na DNA sekvencu, RNA polimeraza prepisuje sve nukleotide (T, C, G i A) u novu RNA sekvencu. Napišite program koji simulira ovaj proces prepisivanja za DNA sekvencu koju unese korisnik. Neka korisnik unosi znakove koji predstavljaju nukleotide, dok god to želi. Svaki put kada korisnik unese ispravni nukleotid, ispišite u konzolu prepisani nukleotid prema sljedećim pravilima:

- T se prepisuje u U
- G se prepisuje u G
- C se prepisuje u C
- A se prepisuje u A

Ako korisnik unese neispravan nukleotid (vrijednost koja nije T, C, G ili A), ispišite odgovarajuću poruku o pogrešci i dopustite korisniku da unese novi nukleotid.

*Moguće rješenje:*

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

int main() {
    char nukleotid;

    do {
        cout << "Unesite nukleotid (T, G, C ili A) ili X za izlaz: ";
        cin >> nukleotid;

        switch (nukleotid) {
            case 'T':
                cout << "Prepisan u: U" << endl;
                break;
            case 'G':
                cout << "Prepisan u: G" << endl;
                break;
            case 'C':
                cout << "Prepisan u: C" << endl;
                break;
            case 'A':
                cout << "Prepisan u: A" << endl;
                break;
            case 'X':
                break;
            default:
                cout << "Neispravan nukleotid" << endl;
                break;
        }
    } while (nukleotid != 'X');

    return 0;
}
```

## Zadatak 14

Neka korisnik prvo unese širinu i visinu ploče, a zatim i x i y koordinate jedne točke. Nacrtajte ploču zadane širine i visine. Koristite znak 'x' za ispis zadane točke te '\_' za ispis svih ostalih praznih ćelija. Pretpostavimo da prvi znak u prvom redu ima koordinate (0, 0).

*Moguće rješenje:*

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

int main() {

    int sirina;
    cout << "Upisite sirinu ploce: ";
    cin >> sirina;

    int visina;
    cout << "Upisite visinu ploce: ";
    cin >> visina;

    int x;
    cout << "Upisite x tocke: ";
    cin >> x;

    int y;
    cout << "Upisite y tocke: ";
    cin >> y;

    for (int redak = 0; redak < visina; redak++) {
        for (int stupac = 0; stupac < sirina; stupac++) {
            if (stupac == x && redak == y) {
                cout << 'x';
            }
            else {
                cout << '_';
            }
        }
        cout << endl;
    }

    return 0;
}
```

## Zadatak 15

Omogućuje korisniku da unese krvnu grupu donora i primatelja (A, B, AB, 0) te odredite i ispišite može li donor dati krv primatelju. Ovo su pravila:

- Tip 0 može donirati bilo kojem tipu.
- Tip A može donirati A i AB.
- Tip B može donirati B i AB.
- Tip AB može donirati samo AB.

*Moguće rješenje:*

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

int main() {

    string donor;
    cout << "Upisite donorovu krvnu grupu: ";
    getline(cin, donor);

    string primatelj;
    cout << "Upisite primateljevu krvnu grupu: ";
    getline(cin, primatelj);

    if (donor == "0") {
        cout << "Moze donirati" << endl;
    }
    else if (donor == "A") {
        if (primatelj == "A" || primatelj == "AB") {
            cout << "Moze donirati" << endl;
        }
        else {
            cout << "Ne moze donirati" << endl;
        }
    }
    else if (donor == "B") {
        if (primatelj == "B" || primatelj == "AB") {
            cout << "Moze donirati" << endl;
        }
        else {
            cout << "Ne moze donirati" << endl;
        }
    }
    else if (donor == "AB") {
        if (primatelj == "AB") {
            cout << "Moze donirati" << endl;
        }
        else {
            cout << "Ne moze donirati" << endl;
        }
    }
    else {
        cout << "Neispravna krvna grupa" << endl;
    }
}
```

```
}    return 0;
```



## Zadatak 16

U biologiji je moguće predvidjeti rast populacije neke vrste. Napišite program koji od korisnika traži unos početne veličine populacije ( $N_0$ ), stope rasta ( $r$ ) i proteklog vremena ( $t$ ). Izračunajte broj jedinki nakon određenog vremena pomoću sljedeće formule:

$$N = N_0 * e^{rt}$$

Gdje je  $e$  približno jednako 2,71828, a  $t$  je izražen u godinama. Na primjer, u nekoj bari je u početku 85 žaba, tako da je  $N_0=85$ . Njihova populacija raste za 18% svake godine, tako da je  $r=0,18$ . To znači da će njihova populacija nakon tri godine ( $t=3$ ) biti 145 žaba.

*Moguće rješenje:*

---

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

int main() {
    const double e = 2.71828;

    int no;
    cout << "Unesite pocetnu populaciju: ";
    cin >> no;

    double r;
    cout << "Unesite stopu rasta: ";
    cin >> r;

    double t;
    cout << "Unesite proteklo vrijeme u godinama: ";
    cin >> t;

    int finalna_population = no * pow(e, r * t);

    cout << "Nakon " << t << " godina, populacija ce biti: " <<
    finalna_population << endl;

    return 0;
}
```

## Zadatak 17

Napišite jednostavnu igru u kojoj korisnik može držati dva predmeta, po jedan u svakoj ruci. U beskonačnoj petlji ponudite sljedeće opcije korisniku:

- Uzmi predmet u lijevu ruku. Ako korisnik već ima predmet u toj ruci, pitajte ga želi li baciti postojeći predmet.
- Uzmi predmet u desnu ruku. Ako korisnik već ima predmet u toj ruci, pitajte ga želi li baciti postojeći predmet.
- Zamijeni predmete iz lijeve u desnu i iz desne u lijevu.

Nakon svake operacije, ispišite predmete iz obje ruke. Primjer mogućeg toka programa:

A = uzmi predmet u lijevu ruku

B = uzmi predmet u desnu ruku

C = zamijeni predmete

> A

Unesite predmet: copper sword

Lijeva ruka: copper sword

Desna ruka:

> A

Vec drzite predmet - zelite li ga odbaciti (1/0)? 1

Unesite predmet: iron sword

Lijeva ruka: iron sword

Desna ruka:

> A

Vec drzite predmet - zelite li ga odbaciti (1/0)? 0

Lijeva ruka: iron sword

Desna ruka:

> B

Unesite predmet: wooden shield

Lijeva ruka: iron sword

Desna ruka: wooden shield

> C

Lijeva ruka: wooden shield

Desna ruka: iron sword

*Moguće rješenje:*

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

int main() {

    string lijeva_ruka = "";
    string desna_ruka = "";
    char akcija;
    bool baci;
    string temp;
```

```

cout << "A = uzmi predmet u lijevu ruku" << endl;
cout << "B = uzmi predmet u desnu ruku" << endl;
cout << "C = zamijeni predmete" << endl;

while (true) {
    cout << "\n> ";
    cin >> akcija;
    cin.ignore();

    switch (akcija) {
    case 'A':
        if (lijeva_ruka != "") {
            cout << "Vec drzite predmet - zelite li ga odbaciti (1/0)?";

            cin >> baci;
            cin.ignore();
            if (baci) {
                lijeva_ruka = "";
            }
        }
        if (lijeva_ruka == "") {
            cout << "Unesite predmet: ";
            getline(cin, lijeva_ruka);
        }
        break;

    case 'B':
        if (desna_ruka != "") {
            cout << "Vec drzite predmet - zelite li ga odbaciti (1/0)?";

            cin >> baci;
            cin.ignore();
            if (baci) {
                desna_ruka = "";
            }
        }
        if (desna_ruka == "") {
            cout << "Unesite predmet: ";
            getline(cin, desna_ruka);
        }
        break;

    case 'C':
        temp = lijeva_ruka;
        lijeva_ruka = desna_ruka;
        desna_ruka = temp;
        break;
    }

    cout << "Lijeva ruka: " << lijeva_ruka << endl;
    cout << "Desna ruka: " << desna_ruka << endl;
}

return 0;
}

```

## Zadatak 18

Napišite program koji od korisnika traži unos broja generacija. Zatim izračunajte i ispišite koliko će se mutacija dogoditi u tom broju generacija, ako znamo da postoji 10% šanse da se mutacija dogodi u svakoj generaciji. Primjer mogućeg toka programa:

Koliko generacija: 1000  
Bilo je 94 mutacija u 1000 generacija.

*Moguće rješenje:*

---

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

int main() {
    srand(time(nullptr));

    int generacija;
    cout << "Koliko generacija: ";
    cin >> generacija;

    int mutacija = 0;
    for (int i = 0; i < generacija; i++) {

        // Generiramo slučajni broj između 1 i 100
        int n = 1 + rand() % (100 - 1 + 1);

        // Provjerimo je li u 10%
        if (n > 90) {
            mutacija++;
        }
    }

    cout << "Bilo je " << mutacija << " mutacija u " << generacija << "
generacija." << endl;

    return 0;
}
```

## Zadatak 19

Pretpostavimo da kolegij ima četiri ishoda učenja, od kojih svaki ima 25 bodova dostupnih za prikupljanje. Tražite od korisnika da unese prikupljene bodove za svaki od ishoda učenja i zatim prikažite konačnu ocjenu korisnika za taj kolegij. Provjerite je li korisnik skupio najmanje 50% dostupnih bodova za svaki ishod učenja. Ocjene su sljedeće:

- Najmanje 50,01 za dovoljan (2)
- Najmanje 58,01 za dobar (3)
- Najmanje 75,01 za vrlo dobar (4)
- Najmanje 92,01 za izvrstan (5)

Primjer mogućeg toka programa:

```
Upisite bodove za ishod 1: 13
Upisite bodove za ishod 2: 21
Upisite bodove za ishod 3: 19.5
Upisite bodove za ishod 4: 22.75
Vrlo dobar (4)
```

*Moguće rješenje:*

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

int main() {

    double i1;
    cout << "Upisite bodove za ishod 1: ";
    cin >> i1;

    double i2;
    cout << "Upisite bodove za ishod 2: ";
    cin >> i2;

    double i3;
    cout << "Upisite bodove za ishod 3: ";
    cin >> i3;

    double i4;
    cout << "Upisite bodove za ishod 4: ";
    cin >> i4;

    double min_prolaz = 25 / 2.0;

    if (i1 < min_prolaz || i2 < min_prolaz || i3 < min_prolaz || i4 < min_prolaz)
    {
        cout << "Pali ste" << endl;
    }
    else {
        double total = i1 + i2 + i3 + i4;

        if (total >= 92.01) {
            cout << "Izvrstan (5)" << endl;
        }
        else if (total >= 75.01) {
            cout << "Vrlo dobar (4)" << endl;
        }
    }
}
```

```
    }
    else if (total >= 58.01) {
        cout << "Dobar (3)" << endl;
    }
    else if (total >= 50.01) {
        cout << "Dovoljan (2)" << endl;
    }
    else {
        cout << "Pali ste" << endl;
    }
}
return 0;
}
```



## Zadatak 21

U igri Dungeons and Dragons igrači koriste kockice s različitim stranama kako bi odredili ishod akcije (na primjer, ako kockica ima 20 strana, tada su mogući ishodi brojevi od 1 do 20). Napišite program koji simulira bacanje kockica tražeći od korisnika da unese broj strana za sve kockice i broj kockica. Zatim ispišite rezultate bacanja za svaku kockicu. Primjer:

```
Upisite broj strana: 20
Koliko kockica bacate: 3
Kockica 1 je pokazala 14
Kockica 2 je pokazala 7
Kockica 3 je pokazala 5
```

*Moguće rješenje:*

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

int main() {
    srand(time(nullptr));

    int strana;
    cout << "Upisite broj strana: ";
    cin >> strana;

    int kockica;
    cout << "Koliko kockica bacate: ";
    cin >> kockica;

    for (int k = 1; k <= kockica; k++) {
        int bacanje = 1 + rand() % (strana - 1 + 1);
        cout << "Kockica " << k << " je pokazala " << bacanje << endl;
    }

    return 0;
}
```



## Zadatak 22

Neka korisnik unese broj N koji određuje visinu trokuta. U svakom retku ispišite brojeve od 1 do broja retka, potencirane brojem retka. Na primjer:

- u prvom redu ispišite broj 1 na potenciju 1
- u drugom redu ispišite brojeve 1 2 na potenciju 2, dakle: 1 4
- u trećem redu ispišite 1 2 3 na potenciju 3, dakle: 1 8 27
- itd.

Primjerice, kada je N=5, treba ispisati:

```
1
1 4
1 8 27
1 16 81 256
1 32 243 1024 3125
```

*Moguće rješenje:*

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

int main() {
    int n;
    cout << "Upisite n: ";
    cin >> n;

    for (int redak = 1; redak <= n; redak++) {
        for (int i = 1; i <= redak; i++) {
            cout << pow(i, redak) << " ";
        }
        cout << endl;
    }

    return 0;
}
```



```
        int n = 1 + rand() % (100 - 1 + 1);
        if (n > 98) {
            cout << "o";
        }
        else {
            cout << "-";
        }
    }
    cout << endl;
}

// More
for (int redak = 1; redak <= visina_more; redak++) {
    for (int stupac = 1; stupac <= sirina; stupac++) {
        cout << "^";
    }
    cout << endl;
}

return 0;
}
```

## Zadatak 24

Napišite program koji ispisuje sve datume u 2023. godini u hrvatskom formatu: dan.mjesec.godina. Koristite sljedeće informacije:

- Siječanj, ožujak, svibanj, srpanj, kolovoz, listopad i prosinac imaju 31 dan.
- Travanj, lipanj, rujanj i studeni imaju 30 dana.
- Veljača ima 28 dana.

Ispis:

```
1.1.2023.  
2.1.2023.  
3.1.2023.
```

...

```
29.1.2023.  
30.1.2023.  
31.1.2023.
```

```
1.2.2023.  
2.2.2023.  
3.2.2023.
```

...

```
31.12.2023.
```

*Moguće rješenje:*

```
#include <iostream>  
using namespace std;  
  
int main() {  
    int max_dan;  
  
    for (int mjesec = 1; mjesec <= 12; mjesec++) {  
        if (mjesec == 2) {  
            max_dan = 28;  
        }  
        else if (mjesec == 1 || mjesec == 3 || mjesec == 5 || mjesec == 7 ||  
mjesec == 8 || mjesec == 10 || mjesec == 12) {  
            max_dan = 31;  
        }  
        else {  
            max_dan = 30;  
        }  
  
        for (int dan = 1; dan <= max_dan; dan++) {  
            cout << dan << "." << mjesec << ".2023." << endl;  
        }  
        cout << endl;  
    }  
  
    return 0;  
}
```