



Природно-математички факултет  
Универзитет у Бањој Луци

# Итеративне наредбе

*Аутори: Драган Матић, Милана Грбић,  
Милан Предојевић, Ненад Вилендечић*

2025

# Садржај

<b>Понављање</b>	<b>3</b>
Гранање (If-else) . . . . .	3
<b>Задаци</b>	<b>4</b>
1   Задатак: Максимум два броја . . . . .	4
2   Задатак: Сума позитивних бројева . . . . .	5
<b>Увод</b>	<b>6</b>
<b>Задаци</b>	<b>8</b>
1   Задатак: Дјелиоци природног броја . . . . .	8
2   Задатак: Унос до -1 . . . . .	9
3   Задатак: Уклањање нула са десне стране . . . . .	10
4   Задатак: Извештај о годишњем пословању . . . . .	11
5   Задатак: Палиндром . . . . .	13
6   Задатак: Генерисање чудних бројева . . . . .	14
<b>Домаћи</b>	<b>15</b>
1   Задатак . . . . .	15
2   Задатак . . . . .	15
3   Задатак . . . . .	15
4   Задатак . . . . .	15
5   Задатак* . . . . .	15

# Понављање

## Гранање (If-else)

If-else структура омогућава доношење одлука у C++ програмима. Основни облик 'if-else' израза је:

```
1 if (uslov) {  
2     // Ako je uslov tacan, izvrsi ovaj blok koda  
3 } else {  
4     // Ako je uslov netacan, izvrsi ovaj blok koda  
5 }
```

## Пример коришћења

Следећи пример демонстрира употребу 'if-else' структуре у C++:

```
1 #include <iostream>  
2  
3 int main() {  
4     int broj;  
5     std::cout << "Unesite broj: ";  
6     std::cin >> broj;  
7  
8     if (broj % 2 == 0) {  
9         std::cout << "Broj je paran" << std::endl;  
10    } else {  
11        std::cout << "Broj je neparan." << std::endl;  
12    }  
13    return 0;  
14 }
```

# Задаци

## 1. Задатак: Максимум два броја

Написати програм који за унесена два реална броја исписује њихов максимум.

### Примјер извршавања

```
Unesite prvi realan broj: 5
Unesite drugi realan broj: 12
Maksimum unesenih brojeva je: 12
```

### Рјешење

```
1 #include <iostream>
2
3 using namespace std;
4
5 int main() {
6     // Deklaracija promenljivih
7     double broj1, broj2;
8
9     // Unos brojeva
10    cout << "Unesite prvi realan broj: ";
11    cin >> broj1;
12
13    cout << "Unesite drugi realan broj: ";
14    cin >> broj2;
15
16    // Odredjivanje i ispis maksimuma
17    if (broj1 >= broj2) {
18        cout << "Maksimum unesenih brojeva je: " << broj1 << endl;
19    } else {
20        cout << "Maksimum unesenih brojeva je: " << broj2 << endl;
21    }
22
23    return 0;
24 }
```

## 2. Задатак: Сума позитивних бројева

Написати програм који учитава три цијела броја са тастатуре и исписује збир позитивних унесених бројева.

### Примјер извршавања

```
Unesite prvi broj: 2
Unesite drugi broj: -3
Unesite treci broj: 9
Zbir pozitivnih brojeva je: 11
```

### Рјешење

```
1 #include <iostream>
2
3 using namespace std;
4
5 int main() {
6     // Deklaracija promjenljivih
7     int broj1, broj2, broj3;
8     int zbir = 0;
9
10    // Unos brojeva
11    cout << "Unesite prvi broj: ";
12    cin >> broj1;
13
14    cout << "Unesite drugi broj: ";
15    cin >> broj2;
16
17    cout << "Unesite treci broj: ";
18    cin >> broj3;
19
20    // Provjera i racunanje zbira pozitivnih brojeva
21    if (broj1 > 0) {
22        zbir += broj1;
23    }
24    if (broj2 > 0) {
25        zbir += broj2;
26    }
27    if (broj3 > 0) {
28        zbir += broj3;
29    }
30
31    // Ispis rezultata
32    cout << "Zbir pozitivnih brojeva je: " << zbir << endl;
33
34    return 0;
35 }
```

## Увод

Итеративне наредбе, познате као петље, омогућавају извршавање истог блока кода више пута док је неки услов испуњен. У C++ постоје три главне итеративне наредбе:

- **for** петља
- **while** петља
- **do-while** петља

Свака од њих се користи у различитим ситуацијама, у зависности од услова понављања.

### for петља

Користи се када знамо **тачан број понављања**. Синтакса је:

```
1 for (<<inicijalizacija>>; <<uslov>>; <<inkrement/dekrement>>) {
2     // Tijelo petlje
3 }
```

### Пример: Испис бројева од 1 до 5

```
1 #include <iostream>
2 using namespace std;
3
4 int main() {
5     for (int i = 1; i <= 5; i++) {
6         cout << i << " ";
7     }
8     return 0;
9 }
```

### while петља

Користи се када **не знамо унапријед** колико ће се пута извршити неки блок.

```
1 while (<<uslov>>) {
2     // Tijelo petlje
3 }
```

### Пример: Испис бројева од 1 до 5

```
1 #include <iostream>
2 using namespace std;
3
4 int main() {
```

```

5     int i = 1;
6     while (i <= 5) {
7         cout << i << " ";
8         i++;
9     }
10    return 0;
11 }

```

## do-while петља

Ова петља **увек изврши бар једну итерацију** јер се услов провјерава након прве итерације.

```

1 do {
2     // tijelo petlje
3 } while (<<uslov>>);

```

## Пример: Испис бројева од 1 до 5

```

1 #include <iostream>
2 using namespace std;
3
4 int main() {
5     int i = 1;
6     do {
7         cout << i << " ";
8         i++;
9     } while (i <= 5);
10
11    return 0;
12 }

```

## Закључак

Петља	Када се користи?
for	Када знамо број понављања
while	Када не знамо број понављања унапред
do-while	Када је потребно да се изврши бар једном

# Задаци

## 1. Задатак: Дјелиоци природног броја

Прави дјелиоци природног броја су сви дјелиоци осим јединице и самог тог броја. Написати програм који за унесен природан број  $n$  исписује све његове праве дјелиоце. У случају неисправног уноса, исписати одговарајућу поруку о грешци.

### Примјер извршавања

Unesi prirodan broj: 18

Pravi djelitelji broja 18 su: 2 3 6 9

### Изузеци

- Уколико корисник не унесе природан број прекинути извршење и исписати одговарајућу поруку.
- Уколико је унесени број прост исписати одговарајућу поруку.

### Рјешење

```
1 #include <iostream>
2 using namespace std;
3
4 int main() {
5     int n;
6     cout << "Unesi prirodan broj: ";
7     cin >> n;
8
9     if (n <= 0) {
10        cout << "Greska: Neispravan unos." << endl;
11        return 1;
12    }
13
14    cout << "Pravi djelitelji broja " << n << " su: ";
15    bool imaDjelitelj = false;
16    for (int i = 2; i <= n / 2; i++) {
17        if (n % i == 0) {
18            cout << i << " ";
19            imaDjelitelj = true;
20        }
21    }
22
23    if (!imaDjelitelj) {
24        cout << "(Nema pravih djelitelja, broj je prost)";
25    }
26
27    cout << endl;
28    return 0;
29 }
```



## 2. Задатак: Унос до -1

Написати програм који од корисника захтијева унос цијелих бројева све док корисник не унесе -1. Након што корисник унесе -1 програм треба да испише суму унесених бројева.

### Примјер извршавања

Unesi cijele brojeve (-1 za kraj):

4

4

12

-1

Suma unesenih brojeva: 20

### Рјешење

```
1 #include <iostream>
2 using namespace std;
3
4 int main() {
5     int broj, suma = 0;
6
7     cout << "Unesi cijele brojeve (-1 za kraj): ";
8
9     do {
10        cin >> broj;
11        suma += broj;
12    } while (broj != -1);
13
14    suma -= broj;
15
16    cout << "Suma unesenih brojeva: " << suma << endl;
17
18    return 0;
19 }
```

### 3. Задатак: Уклањање нула са десне стране

Написати програм који за унесени цијели број исписује број добијен уклањањем свих нула са десне стране унесеног броја.

#### Примјер извршавања

Unesi cijeli broj: 455100

Broj bez nula sa desne strane: 4551

#### Рјешење

```
1 #include <iostream>
2 using namespace std;
3
4 int main() {
5     int broj;
6
7     cout << "Unesi cijeli broj: ";
8     cin >> broj;
9
10    if (broj == 0) {
11        cout << "Rezultat: 0" << endl;
12        return 0;
13    }
14
15    while (broj % 10 == 0) {
16        broj /= 10;
17    }
18
19    cout << "Broj bez nula sa desne strane: " << broj << endl;
20
21    return 0;
22 }
```

## 4. Задатак: Извештај о годишњем пословању

Књиговођа води евиденцију о трансакцијама једне фирме и треба да напише извештај о годишњем пословању те фирме. Фирма је током године имала  $t$  трансакција. Трансакције су представљене цијелим бројевима и у случају да је вриједност трансакције позитивна, та трансакција означава приход фирме, а у случају да је негативна расход. Написати програм који учитава ненегативан цијели број  $t$  и податке о  $t$  трансакцијама и затим израчунава и исписује укупан приход, укупан расход и зараду током године. У случају неисправног уноса, исписати одговарајућу поруку о грешци.

### Примјер извршавања

```
Unesi broj transakcija: 7
Unesi transakciju 1: 8
Unesi transakciju 2: -50
Unesi transakciju 3: 45
Unesi transakciju 4: 2007
Unesi transakciju 5: -67
Unesi transakciju 6: -123
Unesi transakciju 7: 14
Ukupan приход: 2074
Ukupan rashod: -240
Zarada: 1834
```

### Рјешење

```
1 #include <iostream>
2 using namespace std;
3
4 int main() {
5     int t;
6
7     cout << "Unesi broj transakcija: ";
8     cin >> t;
9
10    if (t < 0) {
11        cout << "Greska: Neispravan unos." << endl;
12        return 1;
13    }
14
15    int приход = 0, rashod = 0, transakcija;
16
17    for (int i = 0; i < t; i++) {
18        cout << "Unesi transakciju " << i + 1 << ": ";
19        cin >> transakcija;
20
21        if (transakcija > 0) {
22            приход += transakcija;
23        } else {
```

```
24         rashod += transakcija;
25     }
26 }
27
28 int zarada = prihod + rashod;
29
30 cout << "Ukupan prihod: " << prihod << endl;
31 cout << "Ukupan rashod: " << rashod << endl;
32 cout << "Zarada: " << zarada << endl;
33
34 return 0;
35 }
```

## 5. Задатак: Палиндром

Број је палиндром уколико се исто чита и са лијеве и са десне стране. Написати програм који учитава позитиван цијели број и провјерава да ли је уčitани број палиндром. У случају неисправног уноса, исписати одговарајућу поруку о грешци.

### Примјер извршавања

```
Unesite pozitivan cijeli broj: 54845
Broj je palindrom.
```

### Рјешење

```
1 #include <iostream>
2 using namespace std;
3
4 int main() {
5     int broj, original, obrnuti = 0;
6
7     cout << "Unesite pozitivan cijeli broj: ";
8     cin >> broj;
9
10    if (broj <= 0) {
11        cout << "Greska: Unos mora biti pozitivan cijeli broj."
12    << endl;
13        return 1;
14    }
15
16    original = broj;
17
18    while (broj > 0) {
19        obrnuti = obrnuti * 10 + broj % 10;
20        broj /= 10;
21    }
22
23    if (original == obrnuti) {
24        cout << "Broj je palindrom." << endl;
25    } else {
26        cout << "Broj nije palindrom." << endl;
27    }
28
29    return 0;
30 }
```

## 6. Задатак: Генерисање чудних бројева

Чудни бројеви су они који се могу изразити као збир квадрата два броја. На пример, број 5 је чудан јер важи:  $5 = 1^2 + 2^2$ . Написати програм који генерише све чудне бројеве који су мењи од  $n$ , гдје је  $n$  број унесен са тастатуре.

### Примјер извршавања

```
Unesite broj n: 10
Cudni brojevi manji od 10:
1 = 0^2 + 1^2
2 = 1^2 + 1^2
4 = 0^2 + 2^2
5 = 1^2 + 2^2
8 = 2^2 + 2^2
9 = 0^2 + 3^2
```

### Рјешење

```
1 #include <iostream>
2 using namespace std;
3
4 int main() {
5     int n;
6     cout << "Unesite broj n: ";
7     cin >> n;
8
9     if (n <= 0) {
10        cout << "Greska: N mora biti pozitivan broj." << endl;
11        return 1;
12    }
13
14    cout << "Cudni brojevi manji od " << n << ":" << endl;
15    for (int i = 1; i < n; i++) {
16        for (int a = 0; a * a <= i; a++) {
17            for (int b = a; b * b + a * a <= i; b++) {
18                if (a * a + b * b == i) {
19                    cout << i << " = " << a << "^2 + " << b << "^2" <<
20                    endl;
21                    break;
22                }
23            }
24        }
25    }
26    return 0;
27 }
```

# Домаћи

## 1. Задатак

Написати програм који за унесени позитиван цијели број исписује да ли је он дјељив сумом својих цифара. У случају неисправног уноса, исписати одговарајућу поруку о грешци.

### Примјер извршавања 1

```
Unesite broj: 12
Broj 12 je deljiv sa 3.
```

### Примјер извршавања 2

```
Unesite broj: 2564
Broj 2564 nije deljiv sa 17.
```

## 2. Задатак

Написати програм који учитава позитиван цијели број  $n$ , а потом и  $n$  цијелих бројева. Израчунати и исписати збир оних бројева који су истовремено непарни и негативни. У случају неисправног уноса, исписати одговарајућу поруку о грешци.

### Примјер извршавања

```
Unesite broj n: 5
Unesite n brojeva:
1
-5
-6
3
-11
Zbir neparnih i negativnih: -16
```

## 3. Задатак

Написати програм који за унесени цијели број провјерава и исписује да ли се цифра 5 налази у његовом запису.

## 4. Задатак

Написати програм који учитава цијеле бројеве све до уноса броја нула, а затим израчунава и исписује аритметичку средину унесених бројева на четири децимале.

## 5. Задатак\*

Написати програм који за унети позитиван непаран цијели број  $n$  звездицама исцртава слово X димензије  $n$ .

## Примјер извршавања

Unesite broj n: 5

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