

Početna grupa

1. Naredba while

a. Opšti oblik naredbe while:

```
while(logicki_uslov)
{
    Naredbel
}
```

b. Logički uslov (engl. boolean expression) mora imati vrijednost true ili false. Blok Naredbel, se izvršava sve dok je logički uslov tačan (true); kada uslov postane netačan (false) izvršava se prva naredba iza naredbe while.

c. Obratite pažnju da poslije while **nema** simbola tačka-zapeta.

2. Primjeri upotrebe naredbe while.

```
/* Primjer 1
Stampati 10 puta broj 5, po jedan broj u redu
*/
int brojac; // broji koliko smo puta stampali
brojac = 0; // na pocetku, nijednom nismo stampali
while (brojac < 10)
{
    cout << 5 << endl;
    brojac = brojac + 1;
}
// vrijednost promjenljive brojac u ovom trenutku je 10
```

```
/* Primjer 2
* Stampati sve brojeve od 500 do 800, po jedan u redu
*/
int brojac;
brojac = 500; // na pocetku, brojac sadrzi prvi broj koji stampamo
while (brojac < 800)
{
    cout << brojac << endl;
    brojac = brojac + 1;
}
// vrijednost promjenljive brojac u ovom trenutku je 801
```

```
/* Primjer 2-a
* Ucitati broj n i stampati vase ime n puta.
*/
int n, brojac;
cin>>n;
brojac = 0; //
while (brojac < n)
{
    cout << "Goran" << endl;
    brojac = brojac + 1;
}
```

```

/* Primjer 2-b
* Ucitati broj n i stampati redni broj i vase ime n puta.
*/
    int n, brojac;
    cin>>n;
    brojac = 0; //
    while (brojac < n)
    {
        cout << i+1 << "." <<"Goran" << endl;
        brojac = brojac + 1;
    }

```

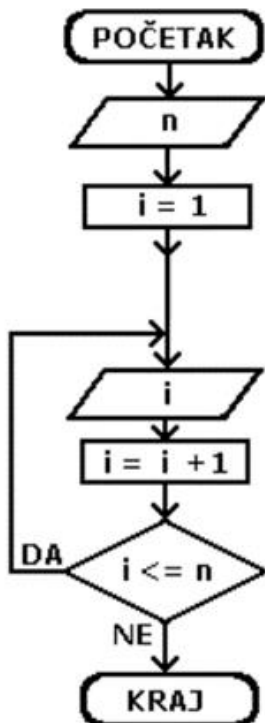
```

/* Primjer 3
Stampati svaki drugi broj od 500 do 800, po jedan u redu
*/
    int brojac;
    brojac = 500; // na pocetku, brojac sadrzi prvi broj koji stampamo
    while (brojac < 800)
    {
        cout << brojac << endl;
        brojac = brojac + 2;
    }
    // vrijednost promjenljive brojac u ovom trenutku je 802

```

Učitati prirodan broj n i štampati sve prirodne brojeve od 1 do n.

Prvi način



```

int n = 0;
cin >> n;

int i = 1;
while (i<=n)
{
    cout << i << endl;
    i = i + 1;
}

```

```
/* Primjer 4
Stampati sve brojeve iz segmenta [0,100] koji su djeljivi sa 3, od najveceg ka
najmanjem
*/
    int brojac;
    brojac = 100; // na pocetku, brojac sadrzi prvi broj
    while (brojac >= 0)
    {
        if (brojac % 3 == 0) // ako je tekuci broj djeljiv sa 3
        {
            cout<< brojac << endl;
        }
        brojac = brojac - 1;
    }
}
```

```
/* Primjer 4 - drugo rjesenje
Prethodno rjesenje ima 101 prolazak kroz petlju while.
Ovo rjesenje ima samo 34 prolaska kroz petlju.
*/
    int brojac;
    // na pocetku, brojac sadrzi prvi broj koji je
    // manji od 100 i koji je djeljiv sa 3
    brojac = 99;
    while (brojac >= 0)
    {
        cout<< brojac << endl;
        brojac = brojac - 3;
    }
}
```

```
/* Primjer 5
Izracunati i stampati zbir svih brojeva iz segmenta [a,b]
*/
    int a = 500, b = 800, zbir = 0;
    brojac = a; // na pocetku, brojac sadrzi prvi sabirak
    while (brojac <= b)
    {
        zbir = zbir + brojac;
        brojac = brojac + 1; // prelazimo na sljedeci sabirak
    }
    cout<< zbir << endl;
```

```

/* Primjer 6
Ucitati dva cijela broja a i b i stampati sve cijele brojeve od a do b,
ukljucujuci i njih, u rastucem poretku (od najmanjeg ka najvećem).
*/
    int a, b, brojac;
    cin>>a>>b;
    int veci, manji;
    if (a<b)
    {
        manji = a;
        veci = b;
    }
    else
    {
        manji = b;
        veci = a;
    }
    brojac = manji; // na pocetku, brojac sadrzi najmanji broj
    while (brojac <= veci)
    {
        cout<< brojac << endl;
        brojac = brojac + 1; // prelazimo na sljedeci broj
    }

```

```

/* Primjer 7
Izracunati i stampati zbir svih brojeva iz segmenta [a,b] koji su djeljivi sa 7
*/
    zbir = 0, a= 100, b = 200;
    brojac = a; // na pocetku, brojac sadrzi prvi sabirak
    while (brojac <= b)
    {
        if (brojac % 7 == 0)
        {
            zbir = zbir + brojac;
        }
        brojac = brojac + 1; // prelazimo na sljedeci sabirak
    }
    cout<< zbir << endl;

```

Zadaci za vježbu (while)

1. Napisati program koji učitava cijele brojeve a i b i štampa sve cijele brojeve iz intervala $[a,b]$, od najmanjeg ka najvećem.
2. Napisati program koji učitava cijele brojeve a i b i štampa sve neparne cijele brojeve iz intervala $[a,b]$, od najvećeg ka najmanjem.
3. Napisati program koji učitava cijele brojeve a i b i štampa sve cijele brojeve iz intervala $[a,b]$ koji pri dijeljenju sa 7 daju ostatak 1 ili ostatak 4.
4. Napisati program koji učitava cijele brojeve a i b i štampa zbir svih cijelih brojeva iz intervala $[a,b]$.
5. Napisati program koji učitava cijele brojeve a i b i štampa zbir kvadrata svih neparnih cijelih brojeva iz intervala $[a,b]$.
6. Napisati program koji učitava prirodna broj n i štampa $n!$ ($n! = 1*2*...*n$)
7. Napisati kod koji učitava prirodan broj n i štampa sve njegove pozitivne djelioce.
8. Napisati kod koji učitava prirodan broj n i štampa zbir svih pozitivnih djelilaca broja n .
9. Prirodan broj n je savršen ako je jednak zbiru svih svojih pozitivnih djelilaca koji su manji od n . Npr. broj 6 je savršen, jer su djelioци broja 6 redom 1, 2 i 3 i važi $1+2+3=6$. Napisati program koji učitava prirodan broj n i provjerava da li je savršen, i ako jeste, štampa poruku "Savršen", a ako nije savršen, štampa "Nije savršen".
10. Unosi se cio broj n , a zatim n cijelih brojeva, po apsolutnoj vrijednosti manjih od 100000. Štampati njihov zbir.
11. Unosi se cio broj n , a zatim n cijelih brojeva, po apsolutnoj vrijednosti manjih od 100000. Štampati njihovu prosječnu vrijednost.
12. Unosi se cio broj n , a zatim n cijelih brojeva, po apsolutnoj vrijednosti manjih od 100000. Štampati najmanji od njih.
13. Brojeve x_1, x_2, \dots, x_n formiramo na sljedeći način: $x_1=1, x_2=3, \dots, x_n=2n^2-1$. Napisati program koji učitava prirodan broj n i štampa sve brojeve x_1, x_2, \dots, x_n , po jedan u redu.
14. Brojeve x_1, x_2, \dots, x_n formiramo na sljedeći način: $x_1=1, x_n=(2n-1)x_{n-1}-n$. Napisati program koji učitava prirodan broj n i štampa sve brojeve x_1, x_2, \dots, x_n , po jedan u redu.

Početna grupa

Zadatak 1

Napisati program koji učitava cio broj n i štampa koliko sadrži parnih cifara.

```
#include <iostream>

using namespace std;
int main()
{
    int n, cifra, brojParnih = 0, pn;
    cout << "Unesite cio broj:";
    cin >> n;
    pn = n; // zapamtimo vrijedost n
    if (n<0)
    {
        pn = -n;
    }
    // odvajamo cifre broja pn
    while(pn>0)
    {
        cifra = pn%10;
        if (cifra % 2 == 0)
        {
            brojParnih++;
            // brojParnih = brojParnih + 1;
            // brojParnih += 1;
        }
        pn = pn /10; // pn /= 10;
    }
    cout << "Broj parnih cifara broja " << n << " je " << brojParnih << endl;
    return 0;
}
```

Zadatak 2

Dva cijela broja x i y su prijateljski brojevi ako se zbir cifara broja x i zbir cifara broja y razlikuju najviše za 1. Napisati program koji učitava cijele brojeve x i y i štampa poruku „FRIENDS” ako su prijatelji i „FOES” ako nisu prijatelji.

```
#include <iostream>

using namespace std;
int main()
{
    int x, px, y, py, cifra, zbirX = 0, zbirY = 0;
    cout << "Unesite 2 cijela broja:";
    cin >> x >> y;
    px = x;
    py = y;
    if (x<0)
    {
        px = -x;
    }
    if (y<0)
    {
        py = -y;
    }

    while(px>0)
    {
        cifra = px%10;
        zbirX = zbirX + cifra;
        px = px /10;
    }
    while(py>0)
    {
        cifra = py%10;
        zbirY = zbirY + cifra;
        py = py /10;
    }
    int d = zbirX-zbirY;
    if (d == 0 || d == 1 || d == -1)
    {
        cout << "FRIENDS" << endl;
    }
    else
    {
        cout << "FOES" << endl;
    }
    return 0;
}
```

Zadatak 3

Napisati program koji učitava cio broj n i štampa broj koji se dobija brisanjem svih neparnih cifara iz broja n .

```
#include <iostream>

using namespace std;
int main()
{
    int n, pn, cifra, rez = 0, stepen = 1;
    cout << "Unesite cio broj:";
    cin >> n;
    pn = n; // zapamtimo vrijedost n
    if (n<0)
    {
        pn = -n;
    }

    while (pn>0)
    {
        cifra = pn%10;
        if (cifra % 2 == 0)
        {
            rez = rez + cifra * stepen;
            stepen = stepen * 10;
        }
        pn = pn /10; // pn /= 10;
    }
    if (n < 0)
    {
        rez = -rez;
    }
    cout << rez << endl;
    return 0;
}
```

Domaći zadatak

1. Napisati program koji učitava cijele brojeve sve dok se ne učita -9999 i štampa koji od učitanih brojeva ima najveću šifru. Šifra cijelog broja n se dobija kao razlika brojeva x i y , gdje je x broj koji se dobija brisanjem svih parnih cifara iz broja n , a y broj koji se dobija brisanjem svih neparnih cifara iz broja n . Ako se ne unese nijedan broj, štampati poruku „NONE”.
2. Cio broj x je izveden iz broja y ako se broj $|x|$ može dobiti brisanjem tačno jedne cifre iz broja y . Napisati program koji učitava cijele brojeve x i y i štampa poruku „GOOD” ako je x izveden iz y ili „BAD” ako nisu prijatelji.

Početna grupa

Zadatak 1

Napisati program koji učitava cio broj n i štampa sljedeću „piramidu” brojeva (dat je primjer za $n=5$).

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

```
#include <iostream>

#include <iostream>

using namespace std;
int main()
{
    int n;
    int red; // oznacava posljednji broj koji stampamo u trenutnom redu
    int cnt; // trenutni broj koji stampamo u redu
    cout << "Unesite prirodan broj:";
    cin >> n;
    red = n;
    while(red >= 1)
    {
        cnt = 1;
        while(cnt <= red)
        {
            cout << cnt << " ";
            cnt++;
        }
        red--; // red -= 1; red = red - 1;
        cout << endl; // odstampam svi brojevi reda - predji u novi red
    }
    return 0;
}
```

Zadatak 2

Napisati program koji učitava cio broj n i štampa sljedeću „piramidu” brojeva (dat je primjer za $n=5$).

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

```
#include <iostream>

using namespace std;
int main()
{
    int n;
    int red; // oznacava posljednji broj koji stampamo u trenutnom redu
    int cnt; // trenutni broj koji stampamo u redu
    cout << "Unesite prirodan broj:";
    cin >> n;
    red = n;
    while(red >= 1)
    {
        cnt = n;
        while(cnt >= red)
        {
            cout << cnt << " ";
            cnt--;
        }
        red--; // red -= 1; red = red - 1;
        cout << endl; // odstampani svi brojevi reda - predji u novi red
    }

    // zavrsono stampanje prvih n redova, prelazimo na poslednjih n-1 redova

    red = 2;
    while(red <= n)
    {
        cnt = n;
        while(cnt >= red)
        {
            cout << cnt << " ";
            cnt--;
        }
        red++; // red -= 1; red = red - 1;
        cout << endl; // odstampani brojevi u redu - predji u novi red
    }
    return 0;
}
```

Domaći zadatak

1. Cio broj x je izveden iz broja y ako se broj $|x|$ može dobiti brisanjem tačno jedne cifre iz broja y . Napisati program koji prvo učitava jedan cio broj x , a zatim učitava cijele brojeve sve dok se ne učita broj koji je po apsolutnoj vrijednosti veći od 2000000000 i za svaki od učitanih brojeva y štampa poruku „GOOD” ako je y izveden iz x ili „BAD” ako se y ne može izvesti iz x .
2. Koristeći samo dva karaktera ‘.’ (tačku) i ‘*’ (zvjezdicu) štampati “šahovsku tablu”. Prvi simbol treba da bude *. Program učitava dva pozitivna cijela broja m i n koji predstavljaju broj redova i broj kolona šahovske table. Vidi tabelu sa primjerima.

	Primjer 1	Primjer 2	Primjer 3
Ulaz	3 1	4 4	2 5
Izlaz	* . *	*.*. *.*. *.*. *.*.	*.*.* *.*.*

3. Koristeći samo dva karaktera ‘.’ (tačku) i ‘*’ (zvjezdicu) štampati oblik “okvira za sliku”. Program učitava dva pozitivna cijela broja m i n koji predstavljaju broj redova i broj kolona treba da ima okvir. Vidi tabelu sa primjerima.

	Primjer 1	Primjer 2	Primjer 3
Ulaz	3 1	4 4	2 5
Izlaz	* * *	**** *.*.* *.*.* ****	***** *****

4. (*) Koristeći samo dva karaktera ‘.’ (tačku) i ‘*’ (zvjezdicu) štampati oblik “mreže” (engl. grid-like pattern). Program učitava dva pozitivna cijela broja m i n koji predstavljaju broj redova i broj kolona mreže. Svako polje mreže ima 4 unutrašnja polja (tj. sadrži po 4 tačke). Vidi tabelu sa primjerima.

	Primjer 1	Primjer 2	Primjer 3
Ulaz	3 1	4 4	2 5
Izlaz	**** *.*.* *.*.* **** *.*.* *.*.* **** *.*.* *.*.* ****	***** *.*.*.*.*.*.* *.*.*.*.*.*.* ***** *.*.*.*.*.*.* *.*.*.*.*.*.* ***** *.*.*.*.*.*.* *.*.*.*.*.*.* ***** *.*.*.*.*.*.* *.*.*.*.*.*.* *****	***** *.*.*.*.*.*.* *.*.*.*.*.*.* ***** *.*.*.*.*.*.* *.*.*.*.*.*.* *****