

Наредби за повторување

1. Структура за повторување на циклус до исполнување на услов

Наједноставниот од сите циклуси е while циклусот. Во овој циклус прво се проверува дали некој услов е исполнет. Ако условот е исполнет, циклусот се повторува, ако условот не е исполнет тогаш не се влегува во циклусот.

Главен дел:

Основна форма на while циклусот е:

```
while (uslov)
{
    blok_od_iskazi;
}
iskaz_po_ciklusot;
```

- Условот во циклусот може да биде споредбен или логички израз кој може да добие вредност точно (true, 1) или неточно (false, 0).

- Блокот од искази во циклусот се повторува сè додека условот е исполнет. Бидејќи условот се проверува на почетокот може да се случи телото на циклусот да не се изврши ниту еднаш.

- Кога условот нема да биде исполнет, блокот од искази во циклусот се прескокнува и програмата ќе продолжи со извршување на првата наредба по телото на циклусот.

- Структурата while мора да биде напишана така што во конечен број на повторувања логичкиот израз ќе добие вредност неточно. Така се обезбедува да се излезе од циклусот по конечен број на повторувања. Во спротивно циклусот би се повторувал бесконечно.

По изразот while (uslov) не се става точка и записка (;).

Пр. Програмата испишува броеви наназад. Корисникот внесува број од кој ќе започне испишувањето.

```
1  #include <cstdlib>
2  #include <iostream>
3  using namespace std;
4  int main()
5  {
6      int n;
7      cout<<"Ke brojам nanazad!"<<endl;
8      cout<<"Od koj broj da zapocnam? ";
9      cin >>n;
10     while (n > 0)
11     {
12         cout<<n<<endl;
13         n--;
14     }
15     system ("PAUSE");
16     return 0;
17 }
```

Да видиме што се случува при извршување на програмата:

n--;	(n>0)	cout<<n;	Број на повторувања
5	1	5	1
4	1	4	2
3	1	3	3
2	1	2	4
1	1	1	5
0	0		

Што ќе се случи ако корисникот внесе 0 или негативен број? Циклусот нема да се изврши ниту еднаш, затоа што прво се проверува условот и во телото на циклусот се влегува само ако условот е исполнет, односно ако логичкиот израз даден во условот има вредност вистина (true, 1).

Ако бројот во секое повторување не се намалува за 1 тогаш логичкиот израз даден во условот (n>0) секогаш ќе биде точен (true, 1) и циклусот ќе се повторува бесконечен број пати, односно додека програмата не ја истроши работната меморија со која располага. Коментирај го исказот n--; и провери!

пр. Програмата пресметува збир на првите n природни броеви. Бројот n го внесува корисникот

```
1 #include <iostream>
2 #include <cstdlib>
3 using namespace std;
4 int main()
5 {
6     int n;
7     int i = 1;
8     int zbir = 0;
9     cout<<"Do koj broj da sobiram? ";
10    cin>>n;
11    while (i<= n)
12    {
13        zbir = zbir + i;
14        i++;
15    }
16    cout<<"Zbirot na broevite od "<<1;|
17    cout<<" do "<<n<<" iznesuva "<<zbir<<endl;
18    system("PAUSE");
19    return 0;
20 }
```

Пр. Корисник внесува n цели броеви. Да се најде нивниот збир.

```
1 #include <cstdlib>
2 #include <iostream>
3 using namespace std;
4 int main()
5 {
6     int n, broj;
7     int i = 1;
8     int zbir = 0;
9     cout<<"Kolku broevi ke vnesete? ";
10    cin>>n;
11    while (i<= n)
12    {
13        cout<<"Vnesi broj: "<<"\t";
14        cin>>broj;
15        zbir = zbir + broj;
16        i++;
17    }
18    cout<<"Zbirot e "<<zbir<<endl;
19    system ("PAUSE");
20    return 0;
21 }
```

Пр. Измени ја претходната програма така да се собираат само позитивни броеви! Ако корисникот внесе негативен број тој нема да влезе во збирот.

```
# include <iostream>
using namespace std;
int main()
{
    int n,i,x,s;
    cin>>n;
    s=0;
    i=1;
    while (i<=n)
    {
        cin>>x;
```

```

    if (x>0) s+=x;
    i++;
}
cout<<s<<endl;
return 0;
}

```

Пр. Корисник внесува броеви, внесувањето се прекинува кога ќе внесе бројот 0. Да се најде аритметичката средина на внесените броеви

```

1  #include <iostream>
2  #include <cstdlib>
3  using namespace std;
4  int main()
5  {
6      int broj, brojac, zbir;
7      brojac = zbir = 0;
8      cout<<"Vnesi broj: ";
9      cin>>broj;
10     while (broj!=0)
11     {
12         zbir = zbir + broj;
13         brojac++;
14         cout<<"Vnesi broj: ";
15         cin>>broj;
16     }
17     if (brojac!=0)
18     {
19         cout<<"Aritmeticka sredina na";
20         cout<<" vnesenite broevi e ";
21         cout<<float(zbir)/brojac<<endl;
22     }
23     else
24         cout<<"Prviot vnesen broj e 0!"<<endl;
25     system("PAUSE");
26     return 0;
27 }

```

2. Структура за повторување do - while

Структурата do – while се користи кога се извршуваат одредени наредби се додека е исполнет условот. Наредбите се извршуваат барем еднаш.

Во структурите за повторување кај кои бројот на повторувања на исказите од циклусот зависи од некој услов, условот може да се испита и на крајот на циклусот. Во овој случај, во програмскиот јазик C++ се користи do-while структура. Оваа структура поретко се користи од структурата while.

Основна форма на do-while циклусот е:

```

do
{
    blok_od_iskazi;
}
while (uslov);

```

Со структурата do-while условот се проверува на крајот од циклусот. Ако условот е исполнет, блокот на искази во циклусот ќе се повтори, а ако условот не е исполнет програмата ќе продолжи со следната наредба по циклусот.

Структурата do-while мора да биде напишана така што во конечен број на повторувања логичкиот израз ќе добие вредност неточно. Така обезбедува да се излезе од циклусот по конечен број на повторувања. Во спротивно циклусот би се повторувал бесконечно.

Не заборавај по циклусот do-while да ставиш точка и записка по условот.

Пр. Програмата испишува броеви наназад. Корисникот внесува број од кој ќе започне

испишувањето.

```
1 #include <cstdlib>
2 #include <iostream>
3 using namespace std;
4 int main()
5 {
6     int n;
7     cout<<"Ke brojам nanazad!"<<endl;
8     cout<<"Od koj broj da zapocnam so odbrojuvanje? ";
9     cin >>n;
10    do
11    {
12        cout<<endl<<n;
13        n--;
14    }
15    while (n>0);
16    cout<<endl<<"Odbrojuvanjeto završeno!"<<endl;
17    system ("PAUSE");
18    return 0;
19 }
```

Кај do-while циклусот, блокот на искази од циклусот ќе се изврши барем еднаш независно дали условот е исполнет или не. Зошто?

Блок од искази кои го сочинуваат циклусот, исто така, се ставени во големи загради. Тоа се исти искази како во while циклусот.

На почетокот на програмата, пред циклусот, од корисникот се бара да внесе број од кој ќе започне броењето.

Ако корисникот внесе број 5 циклусот ќе се повтори точно 5 пати. Во секое повторување програмата го испишува бројот на екранот, потоа тој број го намалува за еден и повторно го проверува условот (n>0). Циклусот ќе се повторува сè додека бројот е поголем од 0. Кога бројот ќе добие вредност 0 изразот broj>0 ќе има вредност неистина (false, 0) и програмата ќе продолжи со извршување на прва наредба по телото на циклусот. Која е тоа наредба?

Што ќе се случи ако корисникот внесе 0 или негативен број? Независно од тоа кој број е внесен, блокот од искази во циклусот ќе се изврши, затоа што прво се влегува во циклусот, а условот се проверува на крајот од циклусот. Ова е основната разлика помеѓу while и do-while структурата.

Ако бројот во секое повторување не се намалува за 1 тогаш логичкиот израз даден во условот (n>0) секогаш ќе биде точен (true, 1) и циклусот ќе се повторува бесконечен број пати, односно додека програмата не ја истроши работната меморија со која располага.

Пр. Корисник внесува броеви, внесувањето се прекинува кога ќе внесе бројот 0. Да се најде аритметичката средина на внесените броеви! Ако првиот внесен број е 0 да се прикаже соодветна порака.

```
1 #include <iostream>
2 #include <cstdlib>
3 using namespace std;
4 int main()
5 {
6     int broj, brojac, zbir;
7     brojac = zbir = 0;
8     do
9     {
10        cout<<"Vnesi broj: "<<endl;
11        cin>>broj;
12        zbir+=broj;
13        brojac++;
14    }
15    while (broj!=0);
16    if (brojac>1)
17    {
18        cout<<"Aritmeticka sredina na";
19        cout<<" vnesenite broevi e ";
20        cout<<float(zbir)/(brojac-1)<<endl;
21    }
22    else
23        cout<<"Prviot vnesen broj e 0!"<<endl;
24    system("PAUSE");
25    return 0;
26 }
```

Пр. Корисникот внесува броеви се додека нивниот збир е помал од 100. Напиши програма со која ќе се изброи колку броеви се внесени!

```
# include <iostream>
using namespace std;
int main()
{
    int b,n,s;
    b=0;
    s=0;
    do
    {
        cout<<"vnesi broj"<<endl;
        cin>>n;
        s+=n;
        b++;

    }
    while (s<=100);
    cout<<"vneseni se "<<b<<" broevi "<<" nivniot zbir e "<<s-n;
    return 0;
}
```

3. Структура за повторување for

Општа форма на for циклусот е:

```
for(pocetna_vrednost; uslov; promena)
{
    blok_na_iskazi;
}
```

Секој for циклус има своја контролна променлива на која вредноста ѝ се менува со секое извршување на циклусот што е одредено со вредноста на прирастот (оваа вредност може да биде и негативна). Условот мора да биде логички израз кој може да добие вредност точно (true, 1) или неточно (false, 0). Блокот од искази кој се наоѓа во циклусот се извршува додека вредноста на условот е вистина. Кога условот ќе добие вредност неvistина, циклусот се прекинува.

По изразот for нема точка и запирка (;), како ниту по третиот израз внатре во заградите.

C++ дозволува контролната променлива да се декларира во самиот for исказ. На пример,

```
for(int i = 1; i <= n; i++)
    cout << i;
```

е исто што и:

```
int i;
for(i = 1; i <= n; i++)
    cout << i;
```

Пр.1. Да се напише алгоритам со кој се пресметува збир на природните броеви од 1 до n.

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

```

int main()
{
    int n,s,i;
    cin>>n;
    s=0;
    for (i=1; i<=n; i++)
        s=s+i;
    cout<<s;
    return 0;
}

```

Пр. Програма со која се пресметува производ на првите n природни броеви. Бројот n го внесува корисникот

```

1  #include <cstdlib>
2  #include <iostream>
3  using namespace std;
4  int main()
5  {
6      int i, n;
7      int proizvod = 1;
8      cout<<"Do koj broj da mnozam? "<<endl;
9      cout<<"Vnesi prirodan broj: ";
10     cin>>n;
11     for (i = 1; i<=n; i++)
12         proizvod = proizvod * i;
13     cout<<"Proizvodot na broevite od "<<1;
14     cout<<" do "<<n<<" e "<<proizvod<<endl;
15     system ("PAUSE");
16     return 0;
17 }

```

Пр. Да се напише програма со која се пресметува збир на непарните природни броеви од 1 до n.

```

# include <iostream>
using namespace std;
int main()
{
    int n,i,s;
    cin>>n;
    s=0;
    for (i=1;i<=n; i+=2)
        s+=i;
    cout<<s;
    return 0;
}

```

Пр. Да се напише програма со која се пресметува збир на n природни броеви.

```

# include <iostream>
using namespace std;
int main()
{
    int n,s,i,x;
    cin>>n;
    s=0;
    for (i=1; i<=n; i++)
    {

```

```

    cin>>x;
    s+=x;
}

cout<<s;
return 0;
}

```

Пр. Да се напише алгоритам со кој се пресметува $s=1+4+7+10+\dots n$.

```

#include <iostream>
using namespace std;
int main()
{
    int n,i,x,s;
    cin>>n;
    s=0;
    i=1;
    while (i<=n)
    {
        s+=i;
        i+=3;
    }
    cout<<s<<endl;
    return 0;
}

```

Пр. Број и збир на цифрите на природен број n.

```

#include <iostream>
using namespace std;
int main()
{
    long int n;
    int b,s,os;
    cout<<"vnesi broj";
    cin>>n;
    b=0;
    s=0;
    while (n > 0)
    {
        os=n%10;
        n=n/10;
        s+=os;
        //cout<<n<<"a";

        // cout<<endl;
        //cout<<b<<"B"<<endl;
        b++;
    }
    cout<<"brojot na cifri e "<<b<<endl;
    cout<<"zbirot na cifri e "<<s<<endl;
    return 0;
}

```

Пр. Корисникот внесува природен број. Напиши програма со која ќе се отпечатат цифрите на внесениот број почнувајќи од цифрата на единици!

```

#include <iostream>
using namespace std;
int main()
{
    long int n,n1;

```

```

cin>>n;
if (n == 0 ) cout<<0;
else
{

while (n > 0)
{
    n1=n%10;
    cout<<n1<<endl;
    n=n/10;

}

}
return 0;
}

```

Пр. Корисникот внесува броеви, внесувањето се прекинува кога ќе внесе 0. Напиши програма со која ќе се прикаже најголемиот број од внесените броеви!

```

# include <iostream>
using namespace std;
int main ()
{
    int n, max;
    cout<<"vnesi broj"<<endl;
    cin>>n;
    max=n;
    do
    {
        cout<<"vnesi broj"<<endl;
        cin>>n;
        if (n>max) max=n;
    }
    while (n != 0);
    cout<<"najgolem e "<<max<<endl;
    return 0;
}

```

Пр. Да се напише програма со која се пресметува збир и број на позитивните вредности од n внесени цели броеви.

```

# include <iostream>
using namespace std;
int main()
{

    int n,s,i,x,b;
    cin>>n;
    s=0;
    b=0;
    for (i=1; i<=n; i++)
    {

```



```

cin>>x;
if (x>0)
{
    b++;
    s+=x;
}
}

cout<<"zbirot na pozitivni vrednosti e "<<s<<endl;
cout<<"brojot na pozitivni vrednosti e "<<b<<endl;

return 0;
}

```

Пр. Напиши програма со која ќе се прикажат сите трицифрени броеви на кои последната цифра им е 0!

```

#include <iostream>
using namespace std;
int main()
{
    int i;
    for (i=100; i<=990; i+=10)
    cout<<i<<endl;
    return 0;
}

```