

Laborator 9

1. Scopul lucrării

Aplicarea și aprofundarea instrucțiunilor repetitive **WHILE-DO** și **DO-WHILE**

2. Considerații teoretice

1. Instrucțiunea WHILE-DO

Această instrucțiune repetă o structură de program cât timp este îndeplinită o condiție.

Forma generală este:

```
while (condiție)
{
    set de instrucțiuni;
}
```

La întâlnirea acestei instrucțiuni, se evaluează condiția. Dacă aceasta este îndeplinită, se execută setul de instrucțiuni. Se revine apoi în punctul în care se evaluează din nou valoarea condiției. Dacă ea este îndeplinită, se repetă din ou setul de instrucțiuni, ș.a.m.d. Astfel, setul de instrucțiuni (corpul ciclului) se repetă atât timp cât condiția este îndeplinită. În momentul în care condiția este neîndeplinită, se iese din ciclu și se trece la următoarea instrucțiune din corpul programului.

Observații:

1. În cazul în care la prima evaluare a expresiei, aceasta nu este îndeplinită, setul de instrucțiuni din ciclul **while** nu va fi executat niciodată.
2. Setul de instrucțiuni din corpul ciclului **while** poate fi compusă (un bloc), sau poate conține o altă instrucțiune ciclică.
3. Este de dorit ca instrucțiunea din corpul ciclului **while** să modifice valoarea condiției. Dacă valoarea condiției este îndeplinită totdeauna, corpul instrucțiunii **while** se repetă de un număr infinit de ori.

Exemplu de ciclul **while** INFINIT:

```
#include<iostream.h>
void main()
{
    int a=0;
    while (a<7)
    {
        cout<<"\nBuna ziua!de "<<a<<" ori";
    }
}
```

Exemplu de ciclul **while** CORECT:

```
#include<iostream.h>
void main()
{
    int a=0;
    while (a<7)
    {
        a++;
        cout<<"\nBuna          ziua!de
        "<<a<<" ori";
    }
}
```

Exemplul 1. Se citește n , număr natural. Să se calculeze suma cifre sale (pentru $n=213$, se va tipări 6).

```
#include <iostream.h>
main ()
{
    int n, s=0;
    cout<<"n=";
    cin>>n;
    while (n)
    {
        s=s+n%10;
        n=n/10;
    }
    cout<<s;
}
```

Exemplul 2. Se citește n , număr natural. Să se afișeze numai obținut prin inversarea cifrelor sale (pentru $n=412$, se va tipări 214).

```
#include <iostream.h>
main()
{
    int n, ninv=0;
    cout<<"n=";
    cin>>n;
    while (n)
    {
        ninv=ninv*10+n%10;
        n=n/10;
    }
    cout<<"numarul inversat "<<ninv;
}
```

2. Instrucțiunea DO-WHILE

Această instrucțiune este asemănătoare cu structura Execută.. .cât timp.
Forma generală a acestei instrucțiuni este următoarea:

```
do
{
    set instrucțiuni;
}
while(condiție);
```

Principiul de executare este următorul:

P1. Se execută *set instrucțiuni*;

P2. Se evaluează *condiția*. atâta timp cât **condiția este îndeplinită** se execută *set instrucțiuni* iar în cazul în care **condiția nu este îndeplinită**, execuția setului de instrucțiuni **do** se termină.

Observație: secvența se execută **cel puțin o dată**, după care se pune condiția dacă să se repete sau nu (prin evaluarea expresiei logice).

Exemplul 1. Se citește un număr natural n , mai mare sau egal cu 1. Să se calculeze suma

primelor n numere naturale.

```
#include <iostream.h>
main ( )
{
    int n, s=0 ,i=1;
    cout<<"n" ;cin>>n;
    do
    {
        s=s+1;
        i=i+1;
    }
    while(i<=s);
    cout<<s;
}
```

Exemplul 2. Se citește n , număr natural. Să se descompună în factori primi

```
#include <iostream.h>
main()
{
    int n,i=2,fm;
    cout<<"n=";
    cin>>n;
    do
    {
        fm=0;
        while(n%i==0);
    {
        fm++; // sau fm=fm+1;
        n/=i; // sau n=n/i;
    }
    if (fm) cout<<i<<" la puterea " <<fm<<endl;
    i++;
    }
    while (n!=1);
}
```