

Laborator 8. Instrucțiuni de ciclare: *while*, *for*, *do-while*. Instrucțiunile *break*, *continue*. Exemple de programe

8.1. Instrucțiunea *while*

Sintaxă:

```
while ( <expresie> ) <instrucțiune>
```

<instrucțiune> este executată repetat, cât timp valoarea <expresie> rămâne diferită de zero. Testul se realizează **înaintea** fiecărei execuții a instrucțiunii <instrucțiune>.

Ex1. Să se calculeze valoarea $n!$, utilizând instrucțiunea *while*.

```
#include<stdio.h>
void main(void)
{
    unsigned long fact;
    unsigned short n,i;
    printf("\nn="); scanf("%u",&n);
    if(n==0||n==1) fact=1;
    else
    {
        fact=1;
        i=2;
        while(i<=n)
            fact=fact*i++;
    }
    printf("%u!=%lu\n",n,fact);
}
```

8.2. Instrucțiunea *do-while*

Sintaxă:

```
do <instrucțiune> while ( <expresie> );
```

<instrucțiune> este executată repetat, cât timp valoarea <expresie> rămâne diferită de zero. Testul se realizează **după** fiecare execuție a instrucțiunii <instrucțiune>.

Ex2. Să se calculeze valoarea $n!$, utilizând instrucțiunea *do-while*.

```
#include<stdio.h>
void main(void)
{
    unsigned long fact;
    unsigned short n,i;
    printf("\nn="); scanf("%u",&n);
    if(n==0||n==1) fact=1;
    else
    {
        fact=1;
        i=2;
        do
            fact=fact*i++;
        while(i<=n);
    }
}
```

```
printf("%u!=%lu\n", n, fact);
}
```

8.3 Instrucțiunea iterativă *for*

Sintaxă:

```
for ( [<expr1>] ; [<expr2>] ; [<expr3>] ) <instrucțiune>
```

<instrucțiune> este executată repetitiv până când valoarea <expr2> devine zero.

Înainte de prima iterație este evaluată <expr1>. De obicei aceasta se realizează pentru a iniția variabilele de ciclare.

După fiecare iterație, se evaluează <expr3>. De obicei, aceasta are rolul de a incrementa contorul de ciclare.

Toate expresiile sunt opționale. Dacă <expr2> este omisă, se presupune că este egală cu 1.

Ex3. Să se calculeze valoarea $n!$, utilizând instrucțiunea *for*.

```
#include<stdio.h>
void main(void){
    unsigned long fact;
    unsigned short n,i;
    printf("\nn="); scanf("%u",&n);
    if(n==0||n==1) fact=1;
    else
        for(i=2,fact=1;i<=n;i++)
            fact=fact*i;
    printf("%u!=%lu\n",n,fact);
}
```

8.4 Instrucțiunea *break*

Sintaxă:

```
break ;
```

Permite ieșirea din ciclul (while, do-while sau for) sau instrucțiunea switch în al cărei corp este inclusă, trecându-se la execuția primei instrucțiuni din afara ciclului (instrucțiunii switch) respectiv(e).

Ex4. Se citește de la tastatură un număr întreg cu semn cu maxim 10 cifre. Să se numere câte cifre egale cu zero, câte cifre pare diferite de zero și câte cifre impare conține numărul citit.

```
#include<stdio.h>
void main(void)
{
    long int nr;
    unsigned short cifra, par, nr_par = 0, nr_impar = 0, nr_0 = 0;
    printf("\nnr="); scanf("%lu",&nr);
    printf("Numarul %lu are ",nr);
    do
    {
        cifra = nr%10;
        nr/=10;
        par=cifra%2;
        if(cifra==0) par=2;
        switch(par)
        {
            case 0: nr_par++; break;
```

```

    case 1: nr_impar++; break;
    case 2: nr_0++; break;
}
while(nr);
printf("%u cifre pare, %u cifre impare si %u \
cifre egale cu zero\n", nr_par, nr_impar, nr_0);
}

```

8.5 Instrucțiunea *continue*

Sintaxă:

```
continue ;
```

În ciclurile *while*, *do-while* sau *for* permite abandonarea iterației curente și se trece la reevaluarea expresiei care stabilește continuarea ciclului.

Ex5. Se dă un șir A de n numere reale, cu termenul general $a_i, i = \overline{1, n}$. Se cere să se obțină șirul ce va conține inversul elementelor din șirul inițial: $a_i = \frac{1}{a_i}, i = \overline{1, n}, a_i \neq 0$. În cazul când un element oarecare este nul, acesta nu se va inversa.

```

#include<stdio.h>
void main(void)
{
    float a;
    unsigned short int n, i;
    printf("\nn="); scanf("%d", &n);
    printf("\nIntroduceti sirul:\n");
    for(i=0; i<n; i++)
    {
        scanf("%f", &a);
        if(a==0)
        {
            printf("Element neinvertibil!\n");
            continue;
        }
        a=1/a;
        printf("Element inversat: %f \n", a);
    }
}

```

8.6 Exerciții propuse spre rezolvare

1. Să se scrie un program care citește o propoziție, reprezentată printr-un șir de caractere și determină numărul de cuvinte din propoziție. Cuvintele vor fi delimitate prin spațiu sau prin virgulă.
2. Să se scrie un program care citește două variabile întregi m și n și determină numărul anilor bisecți din intervalul $[m, n]$.
3. Să se scrie un program care calculează valoarea aproximativă a funcției sinus, într-un punct din domeniul de definiție, luându-se în calcul primii n termeni din dezvoltarea în serii de puteri:

$$\sin x = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \dots$$