

## INSTRUCȚIUNI DE CICLARE

### 1. SCOPUL LUCRĂRII

În această lucrare se vor studia instrucțiuni de ciclare for, while și do...while.

### 2. BREVIAR TEORETIC

#### 2.1. Instrucțiunea a for

Sintaxa:

```
for(<expr1: inițializarea variabilelor de control ale ciclului>;  
    <expr2: testul de continuare a ciclului>;  
    <expr3: actualizarea variabilelor de control ale ciclului>)  
{  
    <grup de instrucțiuni>  
}
```

Observații:

- ÎNAINTE de prima iterație se evaluează expr1.
- DUPĂ fiecare iterație pe buclă, este evaluată expr3.
- Toate cele 3 expresii sunt opționale.
- Repetarea la infinit a unui grup de instrucțiuni se face astfel:  
for(;;){  
 <grup instrucțiuni> }

Exemplu:

```
for(i=0;i<=10;i++)  
    printf("\nPatratul lui %d este %d",i,i*i);
```

#### 2.2. Instrucțiunea a do-while

Sintaxa:

```
do{  
    <grup de instrucțiuni>  
}while(expresie_condiție);
```

Descriere:

Grupul de instrucțiuni este executat repetitiv, cât timp valoarea expresiei rămâne diferită de zero (rămâne adevărată). Expresia de test este evaluată DUPĂ fiecare execuție a grupului de instrucțiuni.

Exemplu:

```
do{  
    printf("\nTastati un numar intreg strict pozitiv: ");
```

---

```
scanf("%d",&nr);  
}while(nr<=0);
```

### 2.3. Instrucțiunea a while

Sintaxa:

```
while(expresie_condiție){  
    <grup de instrucțiuni>  
} //end while
```

Descriere:

Grupul de instrucțiuni se execută repetat, cât timp valoarea expresiei rămâne diferită de 0 (rămâne adevărată). Expresia de test este evaluată ÎNAINTE de fiecare execuție a grupului de instrucțiuni.

Exemplu:

```
stop=0;  
while(!stop){  
    c=getch( );  
    if(c=='q')stop=1;  
    else printf("\Codul Ascii al caracterului %c este %d",c,c);  
}
```

### 2.4. Instrucțiunea a continue

Sintaxa:

```
continue;
```

Descriere:

Determină, în iterația curentă, saltul la sfârșitul secvenței de instrucțiuni ce formează corpul unui ciclu (for, do-while, while) și reluarea ciclului cu iterația următoare (daca mai sunt iterații de făcut).

Exemplu:

```
for(i=1;i<=10;i++){  
    printf("\nTastati un numar intreg: ");  
    scanf("%d",&nr);  
    if(nr==0)continue;  
    printf("\nInversul numarului este %.5lf",1.0/nr);  
}
```

### 2.5. Instrucțiunea a break

Sintaxa:

```
break;
```

Descriere:

---

Realizează ieșirea forțată în afara ciclului curent ( for, do-while, while).

Exemplu:

```
for(;;){
    c=getch( );
    if(c=='q')break;
    else printf("\Codul Ascii al caracterului %c este %d",c,c);
}
```

### **3. DESFĂȘURAREA LUCRĂRII**

Se vor edita și apoi executa programele descrise în continuare.

#### ***Programul nr. 1***

Se citește un număr natural N. Să se afișeze N steluțe, pe verticală.

***Sursa programului:***

```
#include<stdio.h>
#include<conio.h>
void main(void)
{
    clrscr();
    int i,N;
    printf("N=");
    scanf("%d",&N);
    for(i=1;i<=N;i++)
        printf("*\n");
    getch();
}
```

#### ***Programul nr. 2***

Să se calculeze și să se afișeze suma a N numere citite de la tastatură. (N este cunoscut).

***Sursa programului:***

```
#include <stdio.h>
#include <conio.h>
#define N 20
void main(void)
{
    int nr; //numarul curent citit
    int i;
    int suma;
```

---

```
clrscr();
suma = 0;
for(i=1;i<=N;i++) {
    printf("Tastati un numar intreg: ");
    scanf("%d",&nr);
    suma = suma + nr;
}
printf("Suma calculata este %d",suma);
getch();
}
```

**Programul nr. 3**

Să se calculeze și afișeze maximul a N numere întregi citite de la tastatură.

**Sursa programului:**

```
#include <stdio.h>
#include <conio.h>
void main(void)
{
    int i;
    int nrCrt;//numarul curent citit
    int maxim;
    int N; //numarul de numere
    clrscr();
    printf("Introduceti numarul total de numere: ");
    scanf("%d",&N);
    //Se initializeaza maximul, cu primul numar citit:
    printf("Numarul 1: ");
    scanf("%d",&maxim);
    //Se citesc restul numerelor. Daca numarul curent citit,
    //este mai mare decat maxim, se schimba variabila maxim:
    for(i=2;i<=N;i++){
        printf("Numarul %d: ",i);
        scanf("%d",&nrCrt);
        if(nrCrt > maxim)maxim=nrCrt;
    }
    printf("Maximul dintre numerele tastate este %d.",maxim);
    getch();
}
```

---

**Programul nr. 4**

Se citește un număr natural de la tastatură. Să se stabilească dacă este număr prim sau nu.

**Sursa programului:**

```
#include <stdio.h>
#include <conio.h>
#include <math.h>
void main(void)
{
    int a,i;
    int estePrim;// Variabila semafor. Ia valoarea 1, daca a este prim.
                    //Ia valoarea 0, daca a nu este prim.

    clrscr();
    printf("a="); scanf("%d",&a);
    //Un numar este prim, daca are ca divizori,
    // doar pe 1 si pe el insusi.
    estePrim=1;// Initializare.
    for(i=2;i<=sqrt(a);i++)
        if(a%i==0){
            estePrim=0;
            break;}
    if(estePrim==1)printf("Este prim.");
    else printf("Nu este prim.");
    getch();
}
```

**Programul nr. 5**

Să se realizeze un program care scoate ca rezultat numărul de numere naturale de trei cifre care au suma cifrelor mai mare decât 10.

**Sursa programului:**

```
#include <stdio.h>
#include <conio.h>
void main(void)
{
    int N;
    int s;//cifra sutelor numarului N
    int z;//cifra zecilor numarului N
    int u;//cifra unitatilor numarului N
    int nr=0;//nr. de numere naturale care indeplineste conditia
    clrscr();
```

---

```
for(N=100;N<=999;N++){
    s=N/100;
    z=(N-s*100)/10;
    u=N%10;
    if((s+z+u)>10){
        nr++;
        printf("%d ",N); //afisez si numerele care indeplinesc
//conditia
    } // if
} //for
printf("\n%d",nr); //afisez numarul de numere care indeplinesc
//conditia
getch();
} // main
```

**Programul nr. 6**

Se citesc doua numere naturale a si b. Sa se afișeze dacă sunt prime între ele sau nu.

**Sursa programului:**

```
#include<stdio.h>
#include<conio.h>
void main(void)
{
    clrscr();
    int a,b;
    printf("a=");
    scanf("%d",&a);
    printf("b=");
    scanf("%d",&b);
    //Daca nu sunt in ordine crescatoare le interschimbam:
    int aux;
    if(a>b){
        aux=a;
        a=b;
        b=aux;}

    int i;
    int sunt=1;//presupunem ca sunt
    for(i=2;i<=a;i++)
        if((a%i==0)&&(b%i==0)){//au un divizor comun
```

---

```
    sunt=0;
    break;}
if(sunt==1)printf("Sunt prime intre ele");
else printf("Nu sunt prime intre ele");

getch();
}
```

**Programul nr. 7**

Să se rezolve în numere naturale ecuația:  $2x + 3y + 4z = 71$  .

**Sursa programului:**

```
#include <stdio.h>
#include <conio.h>
void main(void)
{
    int x,y,z;
    clrscr();
    //Se genereaza toate tripletele x, y, z , posibile a fi solutie:
    for(x=0;x<=71/2;x++)
        for(y=0;y<=71/3;y++)

            for(z=0;z<=71/4;z++)
                if(2*x+3*y+4*z==71)
                    printf("x=%d, y=%d, z=%d\n",x,y,z);
    getch();
}
```

**Programul nr. 8**

Se repetă citirea unui număr întreg de la tastatură, până când numărul introdus se găsește în intervalul [ 7, 20 ].

**Sursa programului:**

```
#include<stdio.h>
#include<conio.h>
void main(void)
{
    int nr;
    clrscr();
    for(;;){
        printf("\nTastati un numar: ");
        scanf("%d",&nr);
        if((nr>=7)&&(nr<=20))break;
    }
```

---

```
}  
printf("\n S-a tastat %d",nr);  
getch();  
} //main
```

**Programul nr. 9**

Să se calculeze numărul de cifre al unui număr întreg citit de la tastatură.

**Sursa programului:**

```
#include <stdio.h>  
#include <conio.h>  
void main(void)  
{  
    long int n;  
    int nrCifre;  
    clrscr();  
    printf("n=");  
    scanf("%ld",&n);  
    nrCifre=0; //initializare  
    //Numarul de cifre se calculeaza impartind repetat numarul la  
    // valoarea 10, pana cand numarul devine 0.  
    for(;;){  
        n=n/10;  
        nrCifre++;  
        if(n==0)break;  
    }  
    printf("Numarul are %d cifre. ",nrCifre);  
    getch();  
}
```

**Programul nr. 10**

Folosind instrucțiunea do..while, să se scrie un program în care se execută următorul meniu de comenzi:

1. desen triunghi
2. desen pătrat
3. ieșire

Figurile geometrice se desenează în modul text, folosind caracterul steluță (asterisc).

---



**Sursa programului:**

```
#include <stdio.h>
#include <conio.h>
void main(void)
{
    int stop; /*variabila semafor care ne va spune cand este cazul sa
              nu se mai execute acest meniu*/
    int nrOptiune;
    stop=0;
    do{
        //afisare meniu
        clrscr();
        printf("1. Desen triunghi\n");
        printf("2. Desen patrat\n");
        printf("3. Iesire\n");
        printf("Introduceti nr. optiunii: ");
        //se citeste nr.optiunii si apoi se executa comanda:
        nrOptiune=getch();
        //executia meniului:
        switch(nrOptiune) {
            case '1': clrscr();
                    printf(" * \n");
                    printf(" * * \n");
                    printf("*****");
                    getch();
                    break;
            case '2': clrscr();
                    printf("*****\n");
                    printf(" * * \n");
                    printf(" * * \n");
                    printf("*****");
                    getch();
                    break;
            case '3': stop=1;
                    break;
        }
        //switch
    }while(stop==0);
} //end main
```

---

**Programul nr. 11**

Folosind instrucțiunea **do...while**, să se transpună în limbajul C următoarea secvență de program, scrisă în pseudocod:

```
se repetă
    se repetă
        se citește numărul întreg a
        cât timp a este în afara intervalului [1 , 10]
            se citește numărul întreg b
        cât timp b < a .
```

**Sursa programului:**

```
#include <conio.h>
#include <stdio.h>
void main(void)
{
    int a,b;
    clrscr();
    do{
        do{
            printf("a=");scanf("%d",&a);
        }while((a<1)||(a>10));
        printf("b=");scanf("%d",&b);
    }while(b<a);
    printf("a=%d, b=%d",a,b);
    getch();
}
```

**Programul nr. 12**

Să se calculeze cel mai mare divizor comun a doua numere naturale a și b, folosind următorul algoritm:

```
cât timp a ≠ b repetă
    dacă a > b atunci a ← a-b
    altfel b ← b-a
cmmdc ← a
```

**Sursa programului:**

```
#include <stdio.h>
#include conio.h>
void main()
{
    int a;
    int b;
```

---

```

clrscr();
printf("a=");scanf("%d",&a);
printf("b=");scanf("%d",&b);
while(a!=b)
    if(a>b)a=a-b;
    else b=b-a;
printf("cmmdc=%d",a);
getch();
}

```

#### 4. PROBLEME PROPUSE

1. Să se implementeze, folosind instrucțiunea for, următoarea secvență de instrucțiuni, descrisă în pseudocod:

```

    se repetă
        se repetă
            se citește a
            până când  $a > 0$ 
            se citește b
        cât timp  $b < a$ 

```

2. Se citește un număr natural a. Să se calculeze și afișeze primul număr prim mai mare decât a.

3. Să se realizeze un program care realizează c.m.m.d.c. a trei numere.

4. Se citesc două numere N1 și N2, naturale ( $N2 > N1$ ). Câte numere prime sunt în intervalul  $[N1, N2]$  ?

5. Să se implementeze jocul „află numărul”. Operatorul alege un număr între 1 și 1000. Calculatorul trebuie să-l determine din cât mai puține încercări. Calculatorul propune un număr. Operatorul îi răspunde: Este prea mare, este prea mic, sau egal.

6. Să se arate că fracția  $(65n+3)/(39n+2)$  este ireductibilă pentru orice număr natural n cuprins între 1 și 500.

7. Ce semne ( + sau -, deci adunare sau scădere) se pun în locul semnului ? pentru ca expresia următoare să fie adevărată?

$$5 ? 7 ? 2 ? 20 = 30$$


---