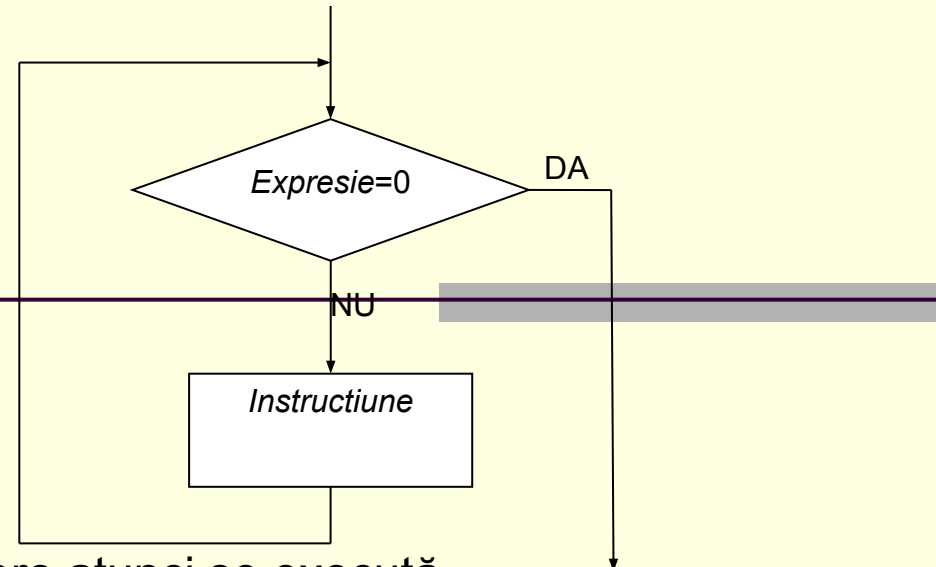


Structura repetitivă cu test inițial

while

Forma: **while**(*expresie*) *instrucțiune*;



Funcționare:

1. Se evaluează expresia din paranteză;
2. Dacă expresia are valoare diferită de zero atunci se execută *instrucțiune* și se reia de la pasul 1, altfel se trece la instrucțiunea următoare instrucțiunii **while**.

Obs:

- a. *Instrucțiune* se execută repetat cât timp *expresie* este diferită de zero.
- b. Este posibil ca *instrucțiune* să nu se execute niciodată.
- c. Este indicat ca în *instrucțiune* să se modifice variabila/variaibilele ce apar în *expresie*, altfel nu se va ieși din structura repetitivă niciodată.

Exemple:

1. Să se afișeze numerele naturale, strict pozitive, mai mici ca un **n** dat.

Următoarea secvență de program își propune să rezolve problema.

```
int n, i=1;  
cout<<"Dati n:"; cin>>n;  
while (i<n)  
    cout<<i++;
```

Cum funcționează:

Să presupunem că am citit pentru **n** valoarea 4:

i=1 și e mai mic ca 4, se tipărește și se incrementează cu 1.

i=2 și e mai mic ca 4, se tipărește și se incrementează cu 1.

i=3 și e mai mic ca 4, se tipărește și se incrementează cu 1.

i=4 și **NU** e mai mic ca 4, se iese din **while**

Pe ecran se afișează: 1 2 3

Exemple:

2. Ce afișează secvența de program de mai jos:

a=6;b=1;

while(a>b)cout<<a--<<b++;

Răspuns: 615243

Justificare:

a=6, b=1, expresia a>b este adevărată, se afișează a și se decrementează, iar apoi se afișează b și se incrementează

a=5, b=2, expresia a>b este adevărată, se afișează a și se decrementează, iar apoi se afișează b și se incrementează

a=4, b=3, expresia a>b este adevărată, se afișează a și se decrementează, iar apoi se afișează b și se incrementează

a=3, b=4, expresia a>b este falsă, se iese din **while**

Exemple:

3. Fiind dat un șir de numere terminat prin 0 să se afișeze suma numerelor.

O posibilă soluție ar fi:

```
int s=0, x;  
cin>>x;      // citim primul număr din sir  
while(x)      // cât timp numărul citit nu e 0  
{s=s+x;      // adunăm numărul la sumă  
  cin>>x;}    // citim următorul număr  
cout<<s;     //tipărim suma
```

Pentru șirul de numere 3 2 7 0 se va executa:

x=3, e diferit de 0, se adună la sumă, se citește următoarea valoare a lui x

x=2, e diferit de 0, se adună la sumă, se citește următoarea valoare a lui x

x=7, e diferit de 0, se adună la sumă, se citește următoarea valoare a lui x

x=0, e 0, se iese din **while** și se tipărește suma

Exemple:

4. Să se calculeze suma cifrelor unui număr natural **n**.

O posibilă soluție ar fi:

```
int s=0, c, n;  
cout<<"Dati numarul:"; cin>>n;  
while(n)           //cât timp mai sunt cifre în n, adică nr. e diferit de 0  
{c=n%10;           // extragem ultima cifră din n  
  s=s+c;           // adunăm cifra extrasă la sumă  
  n=n/10;}         // ștergem ultima cifră din n  
cout<<s;
```

Pentru **n=273** se vor executa următorii pași:

n=273, este > 0, se extrage ultima cifră (3) se adună la sumă, se șterge din **n**

n=27, este > 0, se extrage ultima cifră (7) se adună la sumă, se șterge din **n**

n=2, este > 0, se extrage ultima cifră (2) se adună la sumă, se șterge din **n**

n=0, se iese din **while** și se tipărește suma=3+7+2

Exemple:

5. Fiind date **n** cifre (**n**<10) să se afișeze numărul obținut prin alăturarea cifrelor.

O posibilă soluție ar putea fi:

```
int n, c, x=0;
```

```
cout<<"Dati n:"; cin>>n;
```

```
while(n--)
```

// cât timp n>0, mai avem de citit cifre, decrementăm pe n

```
{ cin>>c;
```

// citim o cifră

```
  x=x*10+c;} // o adăugăm la sfârșitul lui x
```

```
cout<<x;
```

// tipărim numărul obținut prin alăturarea cifrelor

Dacă **n**=3 și cifrele date în ordine sunt 3 7 2 atunci vom avea:

n=3, este >0, se decrementează, se citește cifra (3), se execută $x=0*10+3$

n=2, este >0, se decrementează, se citește cifra (7), se execută $x=3*10+7$

n=1, este >0, se decrementează, se citește cifra (2), se execută $x=37*10+2$

n=0, se iese din **while** și se tipărește **x** (care este 372)

Temă:

1. Fiind dat un număr n ($n < 2$ miliarde) să se determine câte cifre are.
2. Să se verifice dacă un număr este palindrom sau nu.

Indicații:

1. Problema e foarte asemănătoare cu suma cifrelor unui număr, doar că pentru fiecare cifră se adună la sumă 1, nu valoarea cifrei.
2. Se extrag cifrele din numărul citit (vezi problema cu suma cifrelor unui număr) și se adaugă la un lat număr (vezi problema anterioară). Apoi se compară cele 2 numere, dacă sunt egale atunci numărul e palindrom. Se va avea în vedere faptul că prin extragerea cifrelor din număr acesta e distrus ...