

Structuri repetitive

Clasificare:

- a) Cu număr cunoscut de pași
- b) Cu număr necunoscut de pași

a) Cu număr cunoscut de pași

Sintaxă:

```
[ pentru <contor> ← <expresie_inițială>, <expresie_finală>, <pas> execută  
    <bloc_instrucțiuni>  
sfârșit_pentru
```

Mod de execuție:

- <expresie_inițială> și <expresie_finală> sunt expresii cu rezultat întreg, iar <pas> este o valoare întreagă; dacă nu se specifică pasul, acesta va fi considerat, automat, având valoarea 1.
- contorul ia valori de la <expresie_inițială> la <expresie_finală>, iterând intervalul folosind valoarea pasului (contor ← contor+pas)

Exemplu:

1)

n ← 5 (n ia valoarea 5)

```
[ pentru i ← 1, n execută  
    scrie i, " "  
sf._pentru
```

Se va afișa: 1 2 3 4 5

2) Să se determine suma numerelor din intervalul [1,n]

Exemplu: pentru n=5 suma se va calcula astfel

Inițial variabila s (suma) are valoarea 0.

s ← 0+1 adică 1

s ← 1+2 adică 3

s ← 3+3 adică 6

s ← 4+6 adică 10

s ← 5+10 adică 15

15 este valoarea lui s

Pseudocodul corespunzător pentru **problema 2**:

citește n;

```
[ pentru i ← 1, n execută  
    s ← s+i  
sf._pentru  
scrie s
```

3) Se citesc de la tastatură n numere întregi. Să se afle câte numere sunt negative.

```

citește n
[ pentru i ← 1, n execută
  citește x
  [ dacă x < 0 atunci
    s ← s + x
  sf_dacă
sf_pentru
scrie s

```

- 4) Se citesc de la tastatură n numere întregi. Să se calculeze suma numerelor pare.

```

citește n
[ pentru i ← 1, n execută
  citește x
  [ dacă x % 2 == 0 atunci
    s = s + x
  sf_dacă
sf_pentru
scrie s

```

- 5) Specificați ce valori afișează secvența de operații alăturată:

```

citește n
[ pentru i ← n, 1, -1 execută
  scrie i, ' '
sf_pentru

```

b) Cu număr necunoscut de pași

b1) cu test inițial

Se verifică la început condiția de continuare a execuției instrucțiunilor – dacă este adevărată se continuă execuția în caz contrar se oprește execuția structurii repetitive.

Sintaxa structurii:

```

[ cât_timp <condiție> execută
  <bloc_instrucțiuni>
sf_cât_timp

```

Obs.: Dacă este falsă condiția instrucțiunile nu se vor efectua niciodată.

Exemple:

b2) cu test final

Se verifică la început condiția de continuare a execuției instrucțiunilor – dacă este adevărată se continuă execuția în caz contrar se oprește execuția structurii repetitive.

Sintaxa structurii:

repetă
 <bloc_instrucțiuni>
până_când <condiție>
Se revine la începutul structurii doar dacă este adevărată condiția de la sfârșit.

În acest caz verificarea condiției se face la sfârșitul structurii repetitive ceea ce înseamnă că instrucțiunile se execută **cel puțin o dată**.

Exemple:

b1) Pentru Structuri repetitive cu test inițial

1. Ce va afișa pseudocodul de mai jos pentru $n \leftarrow 5$?

```
citește n
i ← n
cât_timp i >= 1 execută
    scrie i, ' '
    i ← i - 1
sf_cât_timp
```

Răspuns: 5, 4, 3, 2, 1

2. Ce va afișa pseudocodul de mai jos pentru $n \leftarrow 5$?

```
citește n
i ← 1
cât_timp i <= n execută
    scrie i, ' '
    i ← i + 1
sf_cât_timp
```

Răspuns: 1, 2, 3, 4, 5

b2) Pentru Structuri repetitive cu test final (condiția este testată la sfârșit)

1. Ce va afișa pseudocodul de mai jos pentru $n \leftarrow 7564$?

```
citește n
execută
    scrie n, ' '
    n ← [n/10]
cât_timp n > 0
```

Se vor afișa valorile: 7564, 756, 75, 7