

## Relații și transformări între structurile repetitive

### 1) Transformarea dintr-o structură cu test inițial într-una cu test final

```
cât_timp <condiție> execută  
    <bloc_instrucțiuni>  
sfârșit_cât_timp
```

și o vom transforma într-o structură de forma:

|                                                                                                                                                             |                                                                                                                                                        |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre>dacă &lt;condiție&gt; atunci<br/>    execută<br/>        &lt;bloc_instrucțiuni&gt; <b>sau</b><br/>    cât_timp &lt;condiție&gt;<br/>sfârșit_dacă</pre> | <pre>dacă &lt;condiție&gt; atunci<br/>    repetă<br/>        &lt;bloc_instrucțiuni&gt;<br/>    până_când not (&lt;condiție&gt;)<br/>sfârșit_dacă</pre> |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------|

Exemplu:

Să se rescrie următorul pseudocod, utilizând o structură repetitivă cu test final. Pseudocodul citește o valoare pentru variabila  $a$  și calculează suma cifrelor lui  $a$ .

```
citește a  
s ← 0  
cât_timp a > 0 execută  
    s ← s + a % 10  
    a ← [a/10]  
sfârșit_cât_timp
```

structura devine

```
citește a  
s ← 0  
dacă a > 0 atunci  
    execută  
        s ← s + a % 10  
        a ← [a/10]  
    cât_timp a > 0  
sfârșit_dacă  
scrie s
```

### 2) Transformarea dintr-o structură cu test final într-una cu test inițial

```
repetă  
    <bloc_instrucțiuni>  
până_când <condiție>
```

O transformăm într-o structură de forma:

```
<bloc_instrucțiuni>  
cât_timp not(<condiție>) execute
```

<bloc\_instrucțiuni>  
sfârșit\_cât\_timp

Exemplu:

Să se rescrie următorul pseudocod utilizând o structură repetitivă cu test inițial.

citește a

$c \leftarrow a$

[ repetă

$c \leftarrow [c/10]$

] până\_când  $c \leq 9$

scrie c

O transformăm într-o structură de forma:

citește a

$c \leftarrow a$

$c \leftarrow [c/10]$

[ cât\_timp  $c > 9$  execută

$c \leftarrow [c/10]$

] sfârșit\_cât\_timp

scrie c

### 3) Transformarea dintr-o structură cu număr cunoscut de pași într-una cu număr necunoscut de pași cu test inițial.

[ pentru contor  $\leftarrow$  inițial, final, pas execută

<bloc\_instrucțiuni>

] sfârșit\_pentru

O transformăm într-o structură de forma:

contor  $\leftarrow$  inițial

[ **cât\_timp** contor  $\leq$  final execută

<bloc\_instrucțiuni>

contor  $\leftarrow$  contor + pas

] sfârșit\_cât\_timp

Exemplu:

Se citește o variabilă întreagă n. Să se determine suma primelor n numere naturale.

Pentru  $n \leftarrow 3$  rezultatul va fi  $S \leftarrow 6$

$S \leftarrow 0$

[ pentru i  $\leftarrow 1, n, 1$  execută

$S \leftarrow S + i$

] sfârșit\_pentru

scrie S

Se va transforma în:

```

S ← 0
i ← 1
[ cât_timp i ≤ n execută
  S ← S + i
  i ← i + 1
sfârșit_cât_timp
scrie S

```

**4) Transformarea dintr-o structură cu număr cunoscut de pași într-una cu număr necunoscut de pași cu test final.**

```

[ pentru contor ← inițial, final, pas execută
  <bloc_instrucțiuni>
sfârșit_pentru

```

O transformăm într-o structură de forma:

```

contor ← inițial
dacă contor ≤ final atunci
[ execută
  <bloc_instrucțiuni>
  contor ← contor + pas
cât_timp contor ≤ final
sfârșit_dacă

```

Exemplu:

Se citește o variabilă întreagă  $n$ . Să se determine suma primelor  $n$  numere naturale.

Pentru  $n \leftarrow 3$  rezultatul va fi  $S \leftarrow 6$

```

S ← 0
[ pentru i ← 1, n, 1 execută
  S ← S + i
sfârșit_pentru
scrie S

```

Se va transforma în:

```

S ← 0
i ← 1
[ execută
  S ← S + i
  i ← i + 1
cât_timp i ≤ n
scrie S

```

**5) Transformarea dintr-o structură cu număr necunoscut de pași și test final într-o altă structură cu număr necunoscut de pași și test final.**

Este vorba de a transforma structura *repetă ... până\_când* în structura *execută ... cât\_timp* sau invers.

Exemplu:

Să se rescrie următorul pseudocod utilizând o altă structură repetitivă cu număr necunoscut de pași și test final.

citește  $n$

execută

scrie  $n$ , ' '

$n \leftarrow [n/10]$

cât\_timp  $n > 0$

scrie  $c$

citește  $n$

repetă

scrie  $n$ , ' '

$n \leftarrow [n/10]$

până\_când  $n \leq 0$

scrie  $c$

**Observație:** La transformarea din structura *repetă ... până\_când* în structura *execută ... cât\_timp* trebuie să negăm condiția.

Aplicații:

Folosind exemplele de mai sus rezolvați următoarele exerciții:

1) citește  $a, b$

pentru  $i \leftarrow a, b$  execută

dacă  $i \% 2 == 0$  atunci

scrie  $i$ , ' '

sfârșit\_dacă

sfârșit\_pentru

a) Scrieți valorile afișate în urma executării algoritmului dacă pentru  $a$  se citește valoarea 5, iar pentru  $b$  valoarea 12.

b) Scrieți în pseudocod un algoritm echivalent cu cel dat, care să conțină o altă structură repetitivă.

```
2) citește n
   s ← 0
   pentru d ← 1, n/2 execută
   [   dacă n%d == 0 atunci
     [   s ← s + d
     sfârșit_dacă
   sfârșit_pentru
   dacă s == n atunci
     scrie "Număr perfect"
   altfel
     scrie "Număr imperfect"
```

- a) Scrieți mesajul afișat în urma executării algoritmului dacă pentru n se citește valoarea 7.
- b) Scrieți în pseudocod un algoritm echivalent cu cel dat, care să conțină o altă structură repetitivă.