

**STRUCTURA REPETITIVĂ CU NUMĂR  
NECUNOSCUT DE PAȘI ȘI TEST FINAL**

# STRUCTURA REPETITIVĂ

Structurile repetitive execută o instrucțiune de un anumit număr de ori, sau cât timp o condiție este adevărată.

## **Structurile repetitive pot fi:**

- **cu număr cunoscut de pași** (iterații) – se cunoaște de la început de câte ori se va executa instrucțiunea
- **cu număr necunoscut de pași** (iterații). Instrucțiunea se execută cât timp o condiție este adevărată. La fiecare pas se va evalua condiția, iar dacă aceasta este adevărată se va executa instrucțiunea.

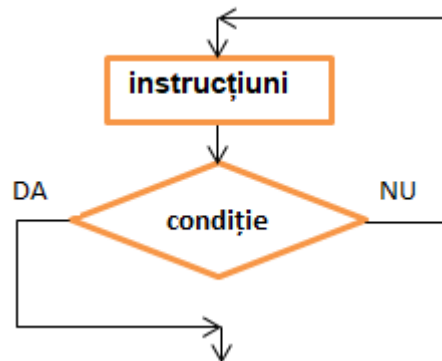
## **Structurile repetitive cu număr necunoscut de pași pot fi:**

- **cu test inițial**: mai întâi se evaluează condiția; dacă este adevărată se execută instrucțiunea și procesul se reia.
- **cu test final**: mai întâi se execută instrucțiunea, apoi se evaluează condiția; Dacă este adevărată, procesul se reia.

# STRUCTURA REPETITIVĂ CU NUMĂR NECUNOSCUT DE PAȘI ȘI TEST FINAL

În cadrul acestei structuri, instrucțiunile se execută cât timp o condiție este adevărată. La fiecare pas se va evalua condiția, iar dacă aceasta este adevărată se va executa instrucțiunile.

Sintaxa:

Schemă logică	Pseudocod	C++
	<pre>repetă     instrucțiuni până când (condiție)</pre>	<pre>do     instrucțiuni while (condiție);</pre>

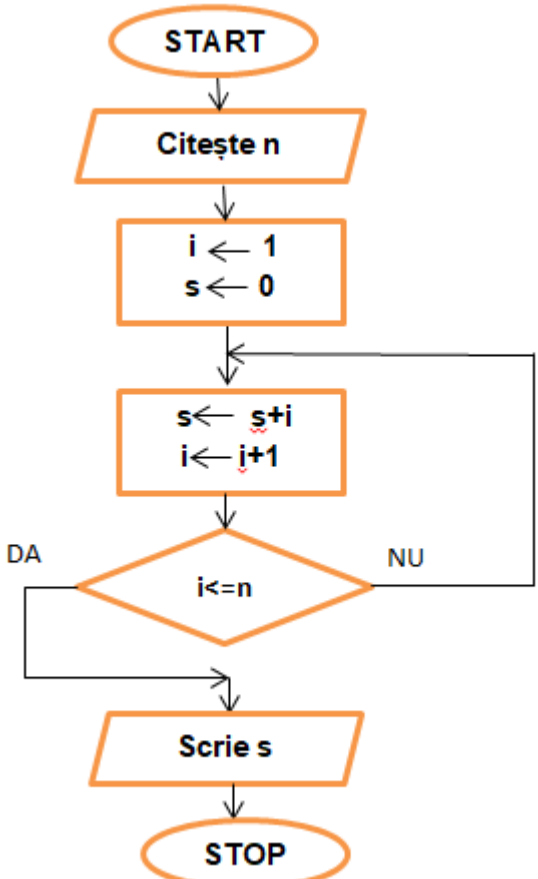
Mod de execuție:

1. Se execută **instrucțiuni**;
2. Se evaluează **condiție**
3. Dacă **condiție** este nenulă, se reia pasul 1.
4. Dacă **condiție** este nulă, se trece la instrucțiunea de după **repetă ... până când**.

# STRUCTURA REPETITIVĂ CU NUMĂR NECUNOSCUȚ DE PAȘI ȘI TEST INIȚIAL

## Aplicație

Se citește de la tastatură un număr natural  $n$ . Calculați, apoi afișați suma primelor  $n$  numere. Scrieți algoritmul de rezolvare.

Schemă logică	Pseudocod	C++
 <pre>graph TD; START([START]) --&gt; ReadN[/Citește n/]; ReadN --&gt; Init["i ← 1 s ← 0"]; Init --&gt; Loop["s ← s+i i ← i+1"]; Loop --&gt; Decision{"i ≤ n"}; Decision -- DA --&gt; Loop; Decision -- NU --&gt; WriteS[/Scrie s/]; WriteS --&gt; STOP([STOP]);</pre>	<pre>întreg n,s,i citește n s&lt;-0 i&lt;-1 repetă     s&lt;-s+i     i&lt;-i+1 până când (i&lt;=n) scrie s</pre>	<pre>int n, s, i; cin&gt;&gt;n; s=0; i=1; do {     s=s+i;     i=i+1; } while (i&lt;=n); cout&lt;&lt;s;</pre>