

# Algoritmi reprezentați prin scheme logice și pseudocod

---

# Pseudocodul

---

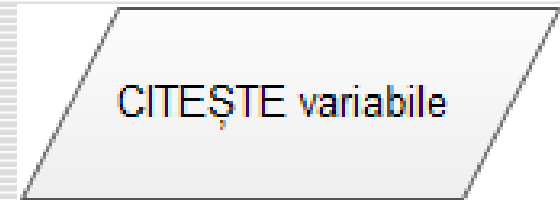
- Este un limbaj de nivel înalt cu ajutorul căruia pot fi descriși algoritmi;
  - Deși folosește convenții structurale asemănătoare cu cele ale unui limbaj de programare, pseudocodul a fost creat mai degrabă pentru a fi înțeles și interpretat de către oameni decât de către calculatoare;
  - Un program pseudocod nu poate fi interpretat sau compilat de către un calculator, dar poate fi cu ușurință “tradus” în limbaj de programare de către un informatician.
-

# Structuri de bază în pseudocod și echivalentul lor în scheme logice

---

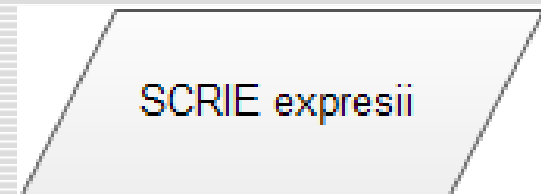
- Pentru citirea datelor, în pseudocod se folosește instrucțiunea:

**citește variabile**



- Pentru afișarea expresiilor, se folosește:

**scrie expresii**



# Structuri de bază în pseudocod și echivalentul lor în scheme logice

---

- Pentru atribuire, în pseudocod se folosește:  
variabilă ← expresie
- Ca și în cazul schemelor logice, mai întâi se evaluează expresia din partea dreaptă a săgeții, apoi valoarea acesteia este reținută de variabila specificată în partea stângă.



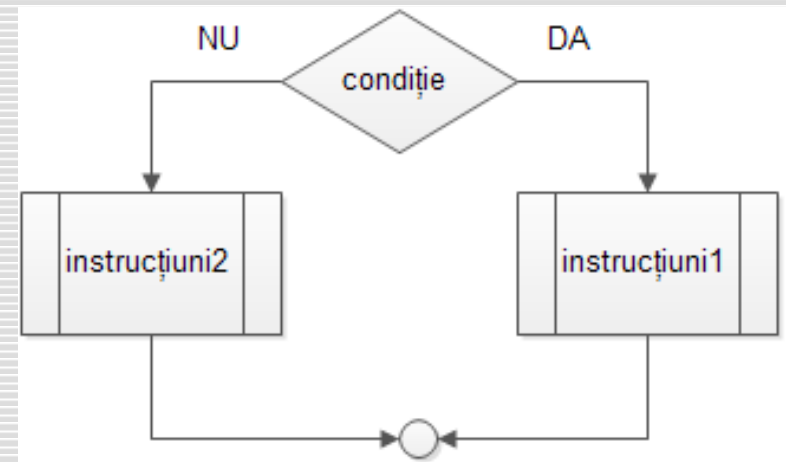
variabilă ← expresie

The diagram shows a rectangular box with a black border containing the text 'variabilă ← expresie'. This represents the logical equivalent of the assignment statement shown in the text to the left.

# Structuri de bază în pseudocod și echivalentul lor în scheme logice

- Echivalentul din pseudocod al blocurilor de decizie specifice schemelor logice este instrucțiunea “dacă”, având forma:

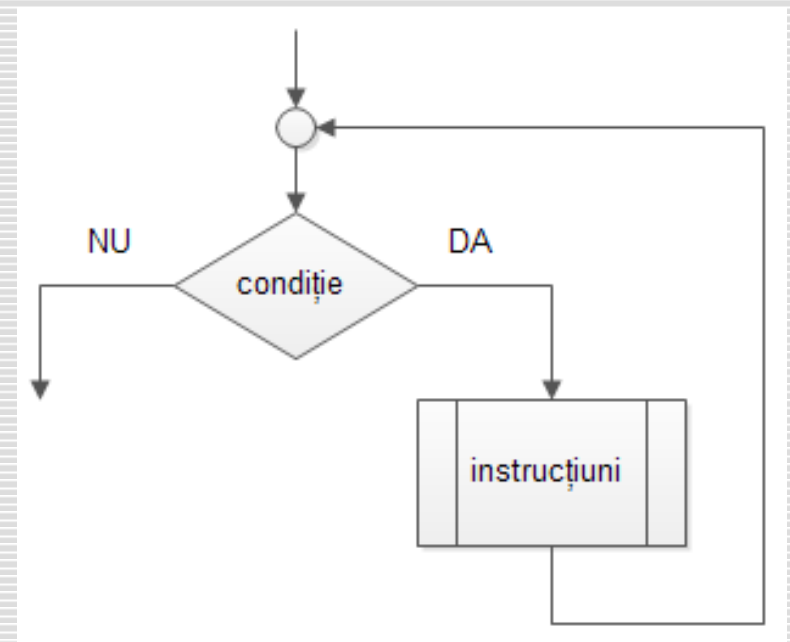
```
dacă condiție  
{  
    instrucțiuni1  
}  
altfel  
{  
    instrucțiuni2  
}
```



# Structuri de bază în pseudocod și echivalentul lor în scheme logice

- Pentru structura repetitivă, în pseudocod există mai multe variante pe care urmează să le studiem;
- Structura repetitivă cu test inițial corespunde schemei logice alăturate și are forma:  

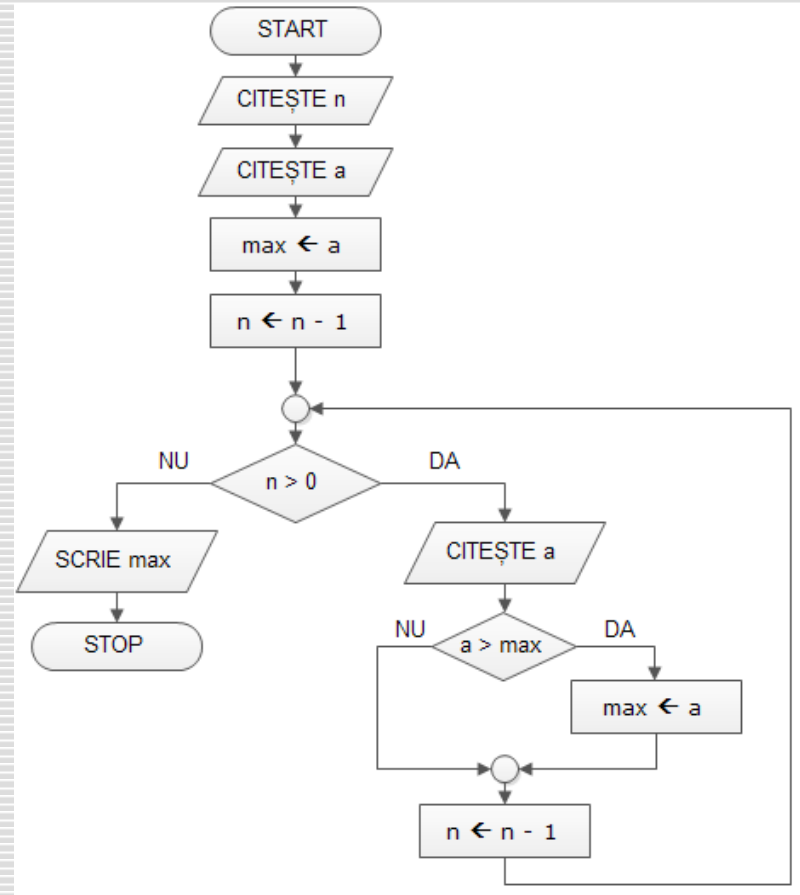
```
cât timp condiție  
{  
    instrucțiuni  
}
```
- Instrucțiunile subordonate vor fi executate atâta timp cât condiția e îndeplinită (expresia logică are valoarea "adevărat")



# Exemple de algoritmi pseudocod și schemele logice echivalente

- Algoritmul de determinare a maximului a  $n$  numere are în pseudocod forma de mai jos:

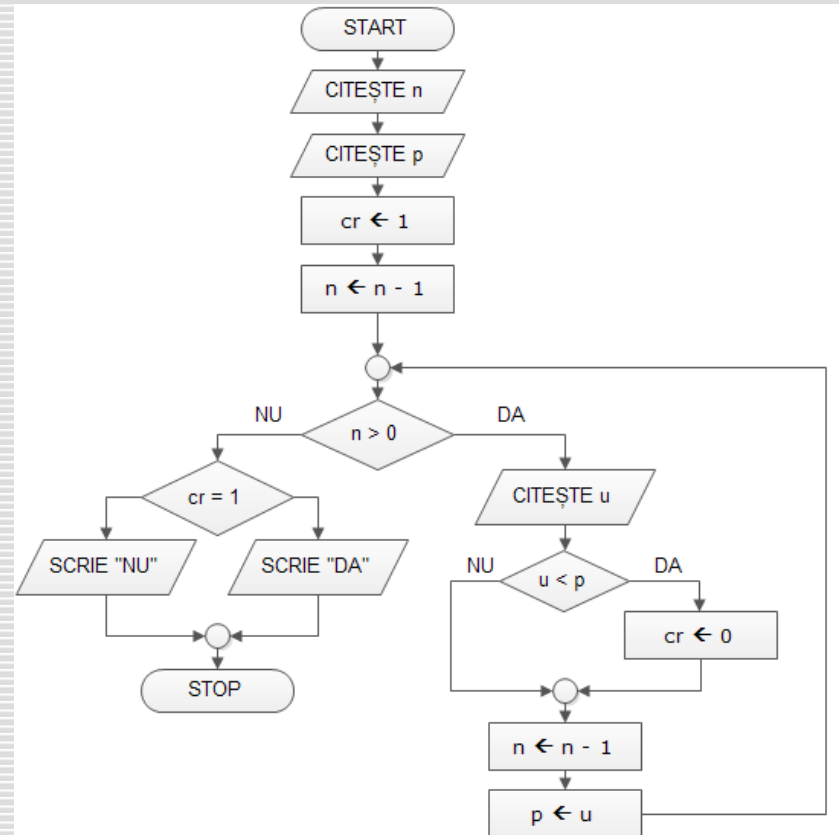
```
citește n
citește a
max ← a
n ← n - 1
cât timp n > 0
{
    citește a
    dacă a > max
    {
        max ← a
    }
    n ← n - 1
}
scrie max
```



# Exemple de algoritmi pseudocod și schemele logice echivalente

- Algoritmul care verifică dacă un șir de  $n$  numere este ordonat crescător are în pseudocod forma de mai jos:

```
citește n
citește p
 $cr \leftarrow 1$ 
 $n \leftarrow n - 1$ 
cât timp  $n > 0$ 
{
    citește u
    dacă  $u < p$ 
    {
         $cr \leftarrow 0$ 
    }
     $n \leftarrow n - 1$ 
     $p \leftarrow u$ 
}
dacă  $cr = 1$ 
{
    scrie "DA"
}
altfel
{
    scrie "NU"
}
```





# Exemple de algoritmi pseudocod și schemele logice echivalente

---

- În algoritmul anterior se folosesc două variabile:  $p$ , reprezentând penultimul număr citit și  $u$ , corespunzător ultimului număr citit;
  - De la al doilea până la cel de-al  $n$ -lea număr citit se verifică dacă  $p \leq u$ ;
  - Dacă la un moment dat această condiție nu mai e îndeplinită, atunci șirul nu e crescător (variabila  $cr$ , inițializată cu 1, devine 0);
  - Pentru a avea mereu în memorie ultimele două valori citite, după fiecare comparare  $p$  îi ia locul lui  $u$ , urmând ca  $u$  să devină următorul număr citit.
-