



ALGORITMI (1)

Obiective

- Definirea unui algoritm
- Gandirea algoritmica. Tipuri de structuri
- Structura liniara.
- Structura alternativa.
- Structura repetitiva.

> Ce este un algoritm?

In general, rezolvarea unei probleme presupune:

- formulare corecta a problemei, incadrarea ei intr-o clasa de probleme
- reprezentarea ei in vederea rezolvarii cu ajutorul calculatorului

Asadar **Algoritm** este o metoda de rezolvare a mai multor probleme de acelasi tip, caracterizata prin **generalitate** (nu exista problema din clasa respectiva nerezolvabila), **corectitudine** (solutia furnizata este corecta) si **finitudinea** (solutia se furnizeaza dupa un numar finit de operatii).

> Gandirea algoritmica. Tipuri de structuri

STRUCTURA LINIARA

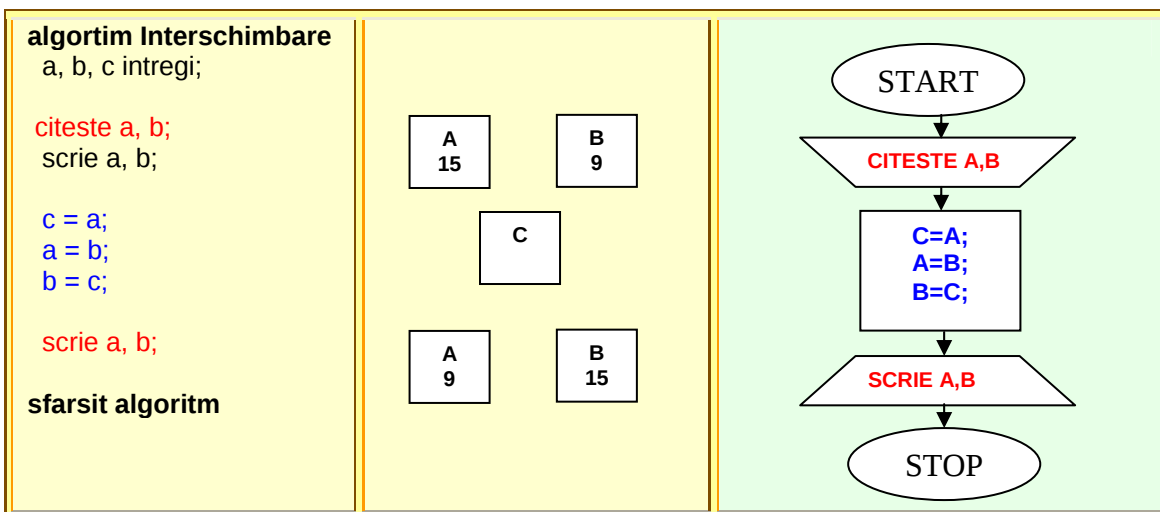
V-ati gandit cum faceti extragerea radacinii patrate dintr-un numar, sau descompunerea unui numar in factori primi? Algoritmul consta dintr-o insiruire de operatii numite instructiuni, efectuate una dupa alta. Numim aceasta insiruire **structura liniara (secventiala)**.



Enumerati operatiile efectuate, numerotandu-le, pentru adunarea a doua numere intregi a si b.

Structura liniara cuprinde numai instructiuni de citire, scriere, calcul si atribuire.

Exemplu :



Algoritm de interschimbare se mai numeste si "**Regula celor trei pahare**", deoarece este necesara o a treia variabila pentru a face interschimbarea. Acest algoritm este intalnit in algoritmi precum **sortarea numerelor**



Realizati un alt algoritm pentru interschimbarea celor doua variabile fara a folosi a treia variabila.



STRUCTURA ALTERNATIVA

Auzim in viata de zi cu zi afirmatii de genul: **DACA** am promovat la toate materiile, **ATUNCI** ma voi duce in tabara, **ALTFEL** stau sa invat.

Se remarca trei cuvinte ce au un rol deosebit: **DACA**, **ATUNCI**, **ALTFEL**. Propozitia are trei componente si anume:

1. o conditie, transcrisa prin "am promovat la toate materiile", conditie pe care o notam cu **c**;
2. o actiune transcrisa prin "ma voi duce in tabara", notata cu **p**, actiune asociata cu **ATUNCI**, adica se executa doar daca "am promovat la toate materiile";
3. o actiune transcrisa prin "stau sa invat", notata cu **q**, actiune asociata cu **ALTFEL**, adica se executa daca **NU** "am promovat la toate materiile";

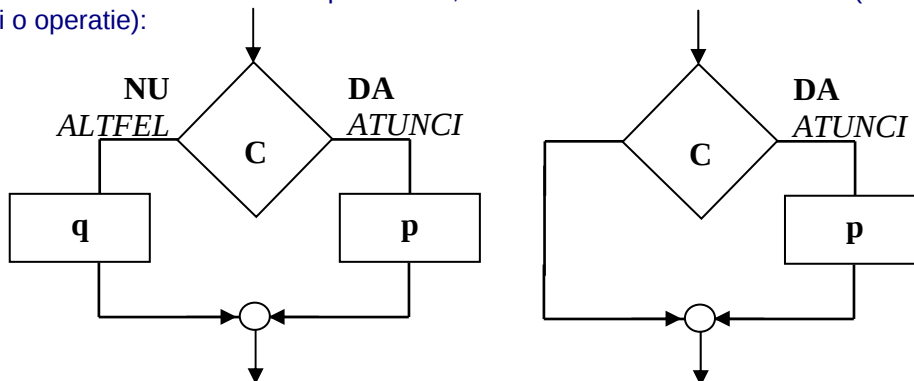
Folosind notatiile facute, afirmatia se poate scrie astfel:

DACA c, ATUNCI p, ALTFEL q.

Pentru a fi foarte clar, afirmatia se poate scrie sub forma:

**DACA c,
ATUNCI p,
ALTFEL q,
SFARSIT**

Secventa de instructiuni se numeste **structura alternativa** si se poate reprezenta si grafic. Structura alternativa admite si o forma particulara, caz in care avem o ramura vida (adica nu se executa nici o operatie):



Exista cazuri in care conditia poate fi mai complexa, de genul:

DACA am promovat la toate materiile **SI** toate mediile sunt peste 7, **ATUNCI** imi voi putea inlocui computerul cu unul mai performant, **ALTFEL** raman cu cel vechi.



Enumerati cazurile in care nu puteti inlocui computerul.



Exemplu : Calculati maximul intre 2 numere intregi.

```

algoritm maxim
a, b, max intregi;
citeste a, b;
daca (a>b)
    max=a
altfel
    max=b;
scrie max;
sfarsit algoritm
  
```

45 89
max = 89

Atentie!
Programul nu trateaza si cazul in care cele doua numere sunt egale! Modificati algoritmul astfel incat daca numerele sunt egale, sa afiseze "Nr. Egale".



STRUCTURA REPETITIVA

Exista trei tipuri de structuri repetitive:

- 1) Structura cu numar cunoscut de repetitii (**FOR**)
- 2) Structura cu numar necunoscut de repetitii si cu test initial (**WHILE**)
- 3) Structura cu numar necunoscut de repetitii si cu test final (**DO-WHILE**)

STRUCTURA REPETITIVA CU TEST INITIAL WHILE

Sa consideram urmatoarea problema:

Se cere sa se cantareasca un sac cu grau (avem suficiente greutate de 1 Kg).

Rezolvarea se reduce la a cantari sacul. Solutia se poate exprima astfel:

CAT TIMP balanta este in dezechilibru, **EXECUTA** adaugarea unei noi greutate de 1 Kg in talerul cu greutate.

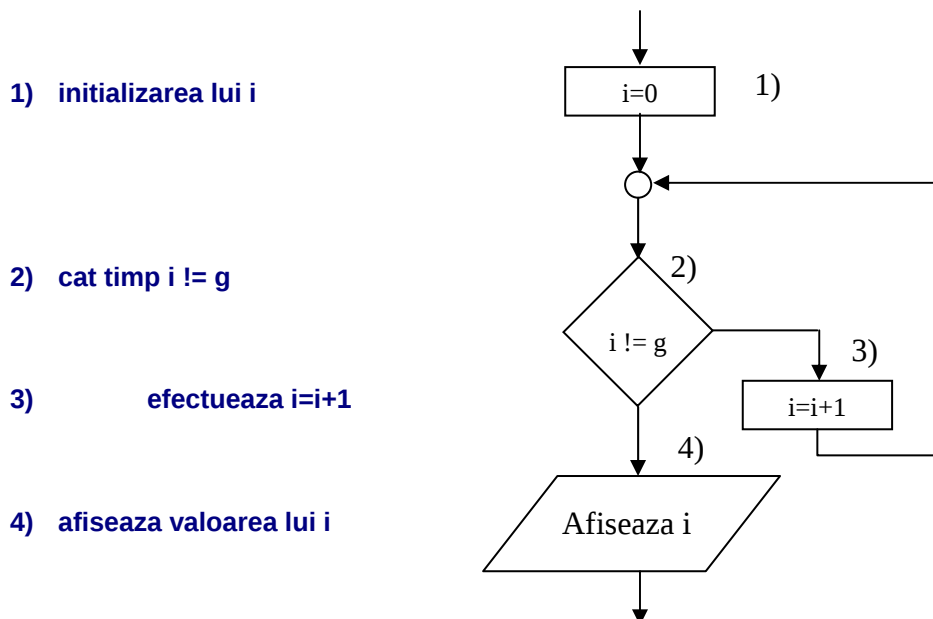
Solutia are 2 componente:

1. o conditie, transcrisa prin "*balanta este in dezechilibru*", conditie pe care o notam cu **c**;
2. o actiune transcrisa prin "*adaugarea unei noi greutate de 1 Kg in talerul cu greutate.*", notata cu **a**, actiune asociata cu **EXECUTA**;

Folosind notatiile facute, solutia se poate scrie astfel:

CAT TIMP c,
EXECUTA a

Notand greutatea sacului cu **g** si cu **i** numarul de greutate puse in balanta, algoritmul complet va fi:



Enumerati instructiunile efectuate, folosind numarul lor de ordine. Faceti o comparatie intre acest algoritm si cel cu structura alternativa.



STRUCTURA REPETITIVA CU NUMAR CUNOSCUIT DE PASI - FOR

Se foloseste in cazurile cand stim numarul de repetitii.
Forma generala a structurii FOR este urmatoarea:

FOR i = val.initial, val.final, pas
.....

De exemplu, daca vrem sa calculam suma primelor n numere naturale:
 $S=1 + 2 + 3 + \dots + n$

vom repeta instructiunea **$S=S+i$** , pentru fiecare i, intre 1 si n.

FOR i = 1, n, 1
 $S = S + i$

Practic, scrierea instructiunii FOR
inlocuieste **3 instructiuni** (1, 2 si 4):

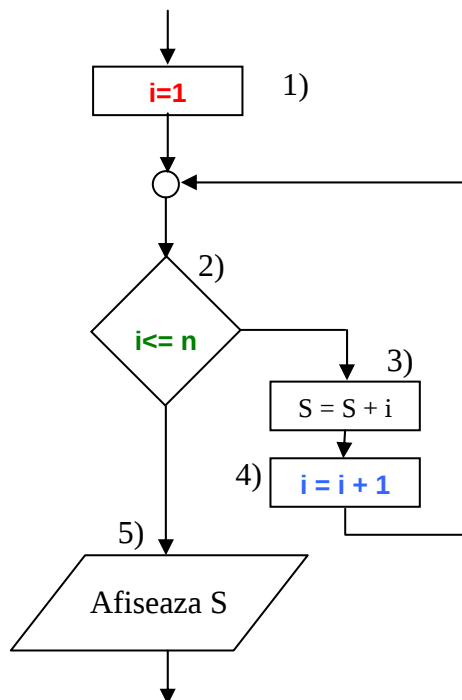
1) initializarea lui i cu 1

2) cat timp $i \leq n$

3) efectueaza $S=S+i$

4) efectueaza $i=i+1$

5) afiseaza valoarea lui i



Cand scriem instructiunea FOR $i=1, n, 1$ inseamna ca i ia valoarea de la 1 la n, din 1 in 1. Am putea alcatui o structura repetitiva FOR cu alt pas decat 1 (de exemplu 2, 3, etc.) Daca pasul este un numar negativ, inseamna ca valoarea lui i va scadea de la valoarea initiala la valoarea finala.



Dati exemple de situatii in care putem folosi structura FOR



Enumerati instructiunile efectuate, folosind numarul lor de ordine. Faceti o comparatie intre algoritmul scris cu **FOR** si cel cu **WHILE**



STRUCTURA REPETITIVA CU TEST FINAL DO - WHILE

Se mai numeste structura repetitiva **REPEAT – UNTIL** (in Pseudocod si in Pascal)

Se foloseste in cazurile in care nu este necesara testarea initiala pentru a realiza repetitia. Principala diferenta intre structura DO-WHILE si WHILE ar fi ca DO-WHILE repeta cel putin odata instructiunile, pe cand in structura WHILE se poate intampla sa nu se execute niciodata. Din acest motiv, este important sa alegem cu grija ce tip de structura repetitiva folosim.



➤ EXERCITII

1. Sa se determine suma primelor 10 numere naturale
2. Sa se determine suma primelor 100 numere naturale
3. Sa se determine suma primelor n numere naturale
4. Sa se determine suma primelor n numere pare
5. Sa se determine suma primelor n numere impare
6. Sa se determine suma patratelor primelor n numere
7. Sa se determine suma cuburilor primelor n numere
8. Sa se determine produsul primelor n numere naturale
9. Sa se calculeze media aritmetica a n numere
10. Sa se calculeze suma inverselor primelor n numere naturale
11. Sa se calculeze suma inverselor patratelor primelor n numere naturale
12. Dandu-se un numar n sa se afiseze daca este par sau nu
13. Dandu-se a si b, sa se determine suma, produsul si media lor aritmetica
14. Se da un numar a. Sa se determine primii 10 multiplii ai lui. (prin adunare, apoi prin inmultire)
15. Se considera trei numere a, b, c. Sa se afiseze cel mai mare dintre ele.
16. Sa se calculeze c.m.m.d.c. al numerelor a si b.
17. Sa se calculeze c.m.m.m.c. al numerelor a si b.
18. Dandu-se un numar n sa se afiseze daca este impar sau nu
19. Dandu-se un numar n sa se afiseze daca este divizibil cu 3
20. Care este cel mai mic numar prim mai mare ca 1000?
21. Sa se afiseze toti divizorii numarului n, dat
22. Sa se afiseze toti divizorii primi ai numarului n, dat
23. Sa se afiseze toate numerele prime mai mici ca n, dat
24. Folosind impartirea repetata, sa se descompuna in factori primi un numar n dat.
25. Sa se extraga radicalul din numarul n dat
26. sa se gaseasca perechile de numere a caror suma este 1000, primul sa fie divizibil cu 17 iar al doilea cu 19.
27. sa se gaseasca perechile de numere a caror suma este 1000, primul sa fie divizibil cu 17 sau cu 13 iar al doilea cu 19 sau cu 7.
28. Sa se gaseasca numarul abc pentru care $a^2+b^2+c^2=a+b+c$
29. sa se genereze toate numerele de 4 cifre de forma 3a2b care se divid cu 9
30. Sa se gaseasca perechile de cifre a si b pentru care numarul 7ab3 sa fie divizibil cu 7 si cu 3
31. Se da un numar x. Sa se afle daca apartine intervalului [a,b] (2 variante: cu AND si fara AND)
32. Se da un numar x. Sa se afle daca NU apartine intervalului [a,b] (2 variante: cu OR si fara OR)
33. (vectori:) Se citeste un sir de numere. Sa se spun ape ce pozitie se afla primul element nul.
34. Se citeste o succesiune de numere pana la zero. Sa se adune cele pozitive, sa se numere cate negative.
35. Sa se determine trei numere x,y,z direct proportionale cu a,b,c si a caror suma este S. (indicatie: (x,y,z) direct prop cu (a,b,c) $\Rightarrow x/a=y/b=z/c=S/(a+b+c) \Rightarrow x=S*a/(a+b+c); y=S*b/(a+b+c); z=S*c/(a+b+c)$)
36. Se da un sir de n numere intregi. Sa se calculeze urmatoarele sume: a celor care se afla inaintea primului element =0; a celor care se afla intre 2 elemente nule, consecutive.
37. Ghiceste numarul
38. Permutarea a doua variabile
39. Se dau n numere. Sa se treaca cele nule la coada
40. Se dau n numere. Sa se faca produsul P al celor diferite de zero. In caz ca toate sunt nule sa se specifice acest lucru. Numerele se vor citi unul cate unul.(se poate folosi un K=0 initial, semafor pt. cazul cand toate sunt nule. K=1 daca nr<>0.)
41. Se da o succesiune de n numere. Sa se calculeze raportul dintre suma algebrica a celor de rang impar si suma algebrica a celor de rang par. Citirea se face element cu element.
42. Suma $S=1^2+2^2+3^2+...+n^2$
43. Suma $S=1^2+3^2+5^2+...+(2n+1)^2$
44. Suma $S=1+1*2+1*2*3+...+1*2*3*...*n$
45. Suma $S=1+2/1*2+3/1*2*3+...+n/1*2*3*...*n$



46. Sa se transforme un numar n din baza 10 in 2.
47. sa se afiseze daca un numar n e divizibil cu : 5, 7
48. sa se afiseze daca un numar a e divizibil cu : b; a+3 cu b; a-b cu 5; a*b cu c
49. sa se afiseze dif dintre x si y daca $x > y$ si suma lor daca $x < y$
50. sa se afle val functiei: $f = \{ \max(x,y), \text{pt. } x < y; 0 \text{ pt. } x = y; \min(x,y) \text{ pt. } x > y$
51. Sa se afle val functiei: $f = \{x-y \text{ pt. } x > y; x+y \text{ pt. } x < y$
52. Sa se calc. m.a. a elementelor sirului $a_1..a_n$, cuprinse intre a si b, $a < b$.
53. Se da un vector $a_1..a_n$. Sa se determine nr. numerelor pozitive si suma lor, apartinand intervalului $[a,b]$.
54. sa se afle m.a intre $\min(x,y^2)$ si $\min(x^2,y)$
55. sa se afle m.a intre $\max(2x,y)$ si $\max(x,2y)$
56. sa se afle m.a intre $a+b$, $\min(a,b)$ si $\min(a-b, (a+b)/2)$
57. sa se afiseze sirul puterilor lui 2 (primii 15 termeni)
58. Se dau n numere. Sa se numere cate sunt mai mici ca 5, egale cu 5 si mai mari ca 5
59. Sa se faca m.a intre suma si produsul elementelor unui vector

Exemplu 1:

```
// folosirea functie strchr()
#include <iostream.h>
#include <string.h>
int main ()
{
    char s[50];
    cin.get(s,20); // aici citim "Hello world"
    cout << strchr(s,'o');
    return 0;
}
```

```
Hello world
o world
```



Atentie! Functia **strchr()** se apleaza ca o functie cu tip. Vezi exemplele.

Exemplu 2:

```
// folosirea functie strchr()
#include <iostream.h>
#include <string.h>
int main ()
{
    char s[50];
    cin.get(s,20); // aici citim "Hello world"
    cout << strchr(s,'o')-s;
    return 0;
}
```

```
Hello world
4
```



Atentie! Se intoarce valoarea 4 deoarece caracterele se numereaza de la 0!