

## Algoritmi.

Algoritmul este o noțiune matematică foarte veche. Cuvântul *algoritm* este de origine arabă. El derivă din numele matematicianului "Abu Ja'far Mohammed İbn Mûsâ al Horezmi" care a scris o carte celebră intitulată "Kitab al jabr w'al - muquabala". Din titlul acestei cărți provine cuvântul *algebră*.

În Evul Mediu, noțiunea de algoritm era utilizată de matematicieni pentru a descrie o regulă pe baza căreia se efectuau calcule matematice.

Odată cu dezvoltarea calculatoarelor cuvântul algoritm a dobândit o semnificație mult mai largă iar gândirea algoritmică este fundamentală pentru rezolvarea problemelor din domenii diverse.

Definiție: Un **algoritm** reprezintă o metodă de rezolvare a unei probleme.

Mărimile cu care lucrează un algoritm, sunt:

- **date de intrare** (sunt datele sau valorile cunoscute din textul problemei)
- **date de ieșire** (sunt valorile care trebuie obținute la sfârșitul rezolvării problemei)

A rezolva o problemă înseamnă a obține, din date de intrare, datele de ieșire, adică rezultatul problemei.

Algoritmul descrie pas cu pas și în ordinea corectă, succesiunea de operații care arată modul prin care datele de ieșire se obțin din datele de intrare.

**Ex. 1:** Să facem un ceai fierbinte având la dispoziție un aragaz, un ibric, o cană, linguriță, apa, un pliculet de ceai și zahăr

**Date de intrare:** aragaz, ibric, cană, apă, pliculet de ceai, zahăr

**Date de ieșire:** cana cu ceai fierbinte

Pas 1. Aprinde aragazul

Pas 2. Pune apa în ibric

Pas 3. Pune ibricul pe foc

Pas 4. Încălzește apa

Pas 5. Pune pliculețul de ceai în cană

Pas 6. Toarnă apa fierbinte peste pliculeț

Pas 7. Pune zahăr în cană

Pas 8. Amestecă cu lingurița până se topește zahărul

Pas 9. Servește ceaiul

**Ex. 2:** Andrei are **A** lei, suficienți pentru a-și cumpăra măcar o înghețată care costă **P** lei. Cu câți bani rămâne Andrei după ce-și cumpără înghețata?

**Date de intrare:** A și P

**Date de ieșire:** restA (suma rămasă)

pas1. citește A, P

pas2. restA ← A-P

pas4. scrie 'Suma ramasa lui Andrei este', restA

???

1. Se poate aplica acest algoritm și pentru Bianca?
2. Ce s-ar întâmpla dacă nu i-ar ajunge banii pentru înghețata dorită?

### Proprietățile fundamentale ale algoritmilor

- **claritate** - fiecare operație care se execută la un moment dat trebuie să fie unic determinată, definită și realizabilă
- **generalitate** - un algoritm trebuie să poată furniza date de ieșire pentru o anumită problemă, pentru mai multe seturi de date de intrare
- **finitudine** - rezultatul problemei se poate obține după un număr finit de pași
- **eficiență** - obținerea rezultatelor corecte după un număr cât mai mic de pași (adică într-un timp cât mai scurt) și cu consum rezonabil de memorie

### Etapele rezolvării unei probleme sunt:

1. **Analiza problemei**, se referă la:
  - identificarea datelor de intrare și a datelor de ieșire
  - stabilirea operațiilor care trebuie efectuate asupra datelor de intrare, pentru a obține datele de ieșire
2. **Conceperea algoritmului de rezolvare a problemei**, se referă la descrierea operațiilor pe pași (ca în ex.1 și 2).
3. **Implementarea algoritmului într-un limbaj de programare**, se referă la scrierea algoritmului într-un limbaj de programare (de ex. C++), capabil de a fi executat automat de către calculator.
4. **Verificarea corectitudinii algoritmului** - acest lucru înseamnă executarea programului pentru diferite seturi de date de intrare.
5. **Analiza complexității algoritmului** - eficiența unui algoritm are în vedere atât spațiul de memorare necesar a fi ocupat pe durata execuției algoritmului, cât și numărul de operații executate de acesta.

### Obiectele cu care lucrează algoritmii sunt: date și expresii.

**Datele** pot fi:

#### a) variabile

O variabilă are alocată o zonă de memorie, are un nume, un tip și o valoare atașată.

De exemplu, **A, P, restA** sunt variabilele ce intervin în ex 2.

b) **constante** - sunt date care nu-și schimbă valoarea pe parcursul execuției programului.

După **tipul** lor, datele pot fi: date întregi, date alfabetice și date logice.

#### Expresiile

O expresie este formată dintr-o succesiune de operanzi intercalați de operatori.

**ex: 25 - 7 + 32**

**operanzi: 25, 7 și 32**

**operatori: - și +**

### Operații efectuate în cadrul algoritmului

În cadrul algoritmului, cu datele putem efectua:

- **operația de citire** - realizează introducerea de date (după implementare, la execuția programului, introducerea datelor va fi în în memoria calculatorului);
- **operația de scriere** - realizează extragerea rezultatelor (în program, datele se extrag din memoria calculatorului și se afișează pe ecran sau se scriu pe hard-disk);
- **operația de atribuire** - permite efectuarea de calcule; este operația prin care o variabilă primește o valoare constantă sau rezultatul evaluării unei expresii.  
ATRIBUIREA **NU** ESTE EGALITATEA MATEMATICĂ!!!
- **operația de decizie** - în funcție de îndeplinirea sau neîndeplinirea unei condiții, algoritmul / programul se ramifică.

## Reprezentarea algoritmilor în pseudocod

**Limbajul pseudocod** reprezintă o altă modalitate foarte utilizată de reprezentare a algoritmilor.

Acesta folosește o serie de **cuvinte-cheie** în descrierea algoritmilor. **Cuvintele-cheie** sunt cuvinte cu o semnificații prestabilite; ele participă în descrierea operațiilor ce formează algoritmul.

În ex.1 cuvintele-cheie utilizate au fost: **citește, scrie**.

**Orice algoritm poate fi reprezentat folosind următoarele tipuri de structuri de control.**

- **Structura secvențială**

*operație 1*

*operație 2*

.....

*operație n*

- **Structura alternativă**

**dacă** expresie

**atunci execută** instructiune\_1

**altfel execută** instructiune\_2

- **Structura repetitivă** – va fi detaliată ulterior

## Aplicații

**Scrieți algoritmi și programe C++ pentru rezolvarea următoarelor probleme:**

1. Cunoscând  $a$  și  $b$  reprezentând lungimile laturilor unui dreptunghi, determinați perimetrul și aria dreptunghiului.

### Algoritm

```
date de intrare: a, b
date de ieșire: p, aria
pas1. citește a,b
pas2.  $p \leftarrow 2*a + 2*b$ 
pas3.  $aria \leftarrow a*b$ 
pas4. scrie p, ' ',aria
```

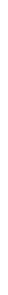
### Programul C++

```
#include<fstream.h>
ifstream fin("date.in");
ofstream fout("date.out");
int main()
{
    int a,b,p, aria;
    fin>>a>>b;
    p = 2*a+2*b;
    aria = a*b;
    fout<<"Perimetrul="<<p;
    fout<<"  aria="<<aria;
    fout.close();
return 0;
}
```

2. Cunoscând cele trei note din oral și nota din teză obținute de un elev la o anumită disciplină , calculați media semestrială a elevului.



3. Fișierul "**date.in**" conține pe prima linie un număr natural reprezentând lungimea laturii unui pătrat. Calculați și afișați perimetrul și aria pătratului.



4. Cunoscând două numere naturale  $a$  și  $b$ , calculați și afișați numărul de numere pare dintre ele (inclusiv  $a$  și  $b$ , dacă este cazul).



5. Cunoscând numărul natural  $a$  mai mare ca 100, scrieți secvența de operații care afișează:

- a) ultima cifră a numărului
- b) ultimele două cifre a numărului
- c) prima cifră a numărului
- d) primele două cifre din număr

6. Pentru un număr natural dat  $N$ ,  $N > 1$ , să se afișeze cele mai apropiate două numere pare de valoarea dată  $N$ , diferite de ea însăși.

**TEMA: Greutatea ideală**



Se știe că greutatea ideală se calculează după formula de mai jos

$$\text{Greutatea\_ideala} = (\text{inaltimea} - 100) + (\text{varsta} / 10) * 90/100$$

unde înălțimea este dată în centimetri iar vârsta în ani împliniți.  
Scrieți un algoritm / program C++ care să calculeze și să afișeze greutatea ideală pentru o persoană pentru care se dă înălțimea  $H$  și vârsta  $V$ .