# Etapele rezolvării unei probleme cu ajutorul calculatorului

Analiza problemei

Construirea programului problemei

Verificarea şi punerea la punct a programului

Determinarea algoritmului de rezolvare

Rezolvarea unei probleme necesită parcurgerea următoarelor etape prezentate în figura :

## ***1) Analiza problemei***

Reformularea problemei în termeni specifici rezolvării cu ajutorul calculatorului.

În această etapă trebuie să fie stabilite :

* 1. **Datele de intrare** (datele iniţiale ale problemei). Pentru acestea se precizează :
     1. denumirea
     2. natura
     3. domeniul de valori posibile
     4. modalitatea prin care vor fi introduse în calculator
  2. **Datele de ieşire** (rezultatele problemei ). Pentru acestea se precizează :
     1. denumirea
     2. natura
     3. forma sub care le va furniza calculatorul
  3. **Funcţia programului** (ceea ce trebuie să realizeze programul)

## ***2) Determinarea algoritmului de rezolvare***

Algoritmul trebuie să conţină o descriere foarte precisă a operaţiilor elementare şi a ordinii în care acestea vor fi executate de calculator pentru a rezolva problema.. Pentru aceasta se porneşte de la funcţia programului şi aceasta se detaliază progresiv până se ajunge la determinarea tuturor operaţiilor elementare.

Această etapă este cea mai dificilă pentru că presupune inteligenţă, inventivitate, fantezie, experienţă.

## 

## ***3) Construirea programului***

Transcrierea algoritmului într-un limbaj de programare. Această activitate se numeşte ***programare***. Limbajul trebuie ales în funcţie de cerinţele problemei.

## ***4) Verificarea şi punerea la punct a programului***

Testarea programului pe mai multe seturi de date de intrare reprezentative şi corectarea eventualelor erori apărute.

Noţiunea de algoritm

Algoritmii[[1]](#footnote-1) constituie baza programării calculatoarelor. Pornind de la un algoritm bine scris putem implementa cu uşurinţă probleme diverse în orice limbaj de programare. Pentru a elabora un algoritm trebuie să dispunem de câteva cunoştinţe de bază care pot fi grupate în mai multe categorii şi anume :

* ***Algoritmi fundamentali***
  1. căutare
  2. sortare
  3. interclasare
  4. algoritmi de aritmetica numerelor
* ***Tehnici de programare***

|  |  |
| --- | --- |
| Nr.  Crt. | Denumire  tehnică de programare |
|  | Backtracking |
|  | Divide et Impera |
|  | Greedy |
|  | Programare dinamică |
|  | Branch and Bound |

* ***Scheme de descompunere a unei probleme în probleme mai simple***

***Definiţie[[2]](#footnote-2)***

Un algoritm este o succesiune finită şi ordonată de operaţii bine determinate care acţionează asupra unor date iniţiale, obţinându-se în final rezultate.

Un algoritm are următoarele proprietăţi :

## ***1) Claritatea***

Operaţiile şi ordinea efectuării lor trebuie să aibă o descriere clară şi precisă. Nu sunt admise ambiguităţile. Din această cauză acţiunile presupuse de algoritm se vor executa la un moment dat chiar mecanic, automat.

## ***2) Generalitatea***

Un algoritm nu este destinat rezolvării unei probleme particulare ci rezolvării unei întregi clase de probleme.

*Exemplu:*

* rezolvarea ecuaţiei de gradul II cu o necunoscută
* Algoritmul lui Euclid pentru aflarea c.m.m.d.c a două numere

## ***3) Finitudinea***

Un algoritm conţine un număr finit de operaţii chiar dacă acestea sunt în număr foarte mare.

*Exemplu:*

Procedeul de extragere a radicalului de ordinul doi dintr-un număr este o metodă de calcul , dar nu este un algoritm pentru că rezultatul poate avea un număr infinit de zecimale. Această metodă poate fi transformată în algoritm prin limitarea numărului de zecimale al rezultatului.

## ***4) Eficienţa***

# Dintre algoritmii care rezolvă o anumoită clasă de probleme prezintă interes aceia care au un număr cât mai mic de operaţii elementare. Din acest motiv studiul complexităţii algoritmilor este foarte important

# Noţiunea de pseudocod

Algoritmii[[3]](#footnote-3) constituie baza programării calculatoarelor. Pornind de la un algoritm bine scris putem implementa cu uşurinţă probleme diverse în orice limbaj de programare. Pentru a elabora un algoritm trebuie să dispunem de câteva cunoştinţe de bază care pot fi grupate în mai multe categorii şi anume :

* ***Algoritmi fundamentali***
  1. căutare
  2. sortare
  3. interclasare
  4. algoritmi de aritmetica numerelor
* ***Tehnici de programare***

|  |  |
| --- | --- |
| Nr.  Crt. | Denumire  tehnică de programare |
|  | Backtracking |
|  | Divide et Impera |
|  | Greedy |
|  | Programare dinamică |
|  | Branch and Bound |

* ***Scheme de descompunere a unei probleme în probleme mai simple***

***Definiţie[[4]](#footnote-4)***

Un algoritm este o succesiune finită şi ordonată de operaţii bine determinate care acţionează asupra unor date iniţiale, obţinându-se în final rezultate.

Un algoritm are următoarele proprietăţi :

## ***1) Claritatea***

Operaţiile şi ordinea efectuării lor trebuie să aibă o descriere clară şi precisă. Nu sunt admise ambiguităţile. Din această cauză acţiunile presupuse de algoritm se vor executa la un moment dat chiar mecanic, automat.

## ***2) Generalitatea***

Un algoritm nu este destinat rezolvării unei probleme particulare ci rezolvării unei întregi clase de probleme.

*Exemplu:*

* rezolvarea ecuaţiei de gradul II cu o necunoscută
* Algoritmul lui Euclid pentru aflarea c.m.m.d.c a două numere

## ***3) Finitudinea***

Un algoritm conţine un număr finit de operaţii chiar dacă acestea sunt în număr foarte mare.

*Exemplu:*

Procedeul de extragere a radicalului de ordinul doi dintr-un număr este o metodă de calcul , dar nu este un algoritm pentru că rezultatul poate avea un număr infinit de zecimale. Această metodă poate fi transformată în algoritm prin limitarea numărului de zecimale al rezultatului.

## ***4) Eficienţa***

Dintre algoritmii care rezolvă o anumoită clasă de probleme prezintă interes aceia care au un număr cât mai mic de operaţii elementare. Din acest motiv studiul complexităţii algoritmilor este foarte important.

[Etapele rezolvării unei probleme cu ajutorul calculatorului 1](#_Toc129172611)

[***1) Analiza problemei*** 1](#_Toc129172612)

[***2) Determinarea algoritmului de rezolvare*** 1](#_Toc129172613)

[***3) Construirea programului*** 1](#_Toc129172614)

[***4) Verificarea şi punerea la punct a programului*** 1](#_Toc129172615)

[***1) Claritatea*** 2](#_Toc129172616)

[***2) Generalitatea*** 2](#_Toc129172617)

[***3) Finitudinea*** 2](#_Toc129172618)

[***4) Eficienţa*** 2](#_Toc129172619)

[Dintre algoritmii care rezolvă o anumoită clasă de probleme prezintă interes aceia care au un număr cât mai mic de operaţii elementare. Din acest motiv studiul complexităţii algoritmilor este foarte important 2](#_Toc129172620)

[Noţiunea de pseudocod 3](#_Toc129172621)

[***1) Claritatea*** 3](#_Toc129172622)

[***2) Generalitatea*** 3](#_Toc129172623)

[***3) Finitudinea*** 3](#_Toc129172624)

[***4) Eficienţa*** 3](#_Toc129172625)

|  |  |  |
| --- | --- | --- |
| catelus | cu | parul |
| cret | fura | rata |

1. Termenul de algoritm provine de la numele matematicianului arab al-Horezmî [↑](#footnote-ref-1)
2. Orice definiţie dată noţiunii de algoritm este doar aproximativă. Aşa cum matematica nu poate fi definită, nici noţiunea de algoritm nu are o definiţie universal acceptată [↑](#footnote-ref-2)
3. Termenul de algoritm provine de la numele matematicianului arab al-Horezmî [↑](#footnote-ref-3)
4. Orice definiţie dată noţiunii de algoritm este doar aproximativă. Aşa cum matematica nu poate fi definită, nici noţiunea de algoritm nu are o definiţie universal acceptată [↑](#footnote-ref-4)