Programarea si utilizarea calculatoarelor

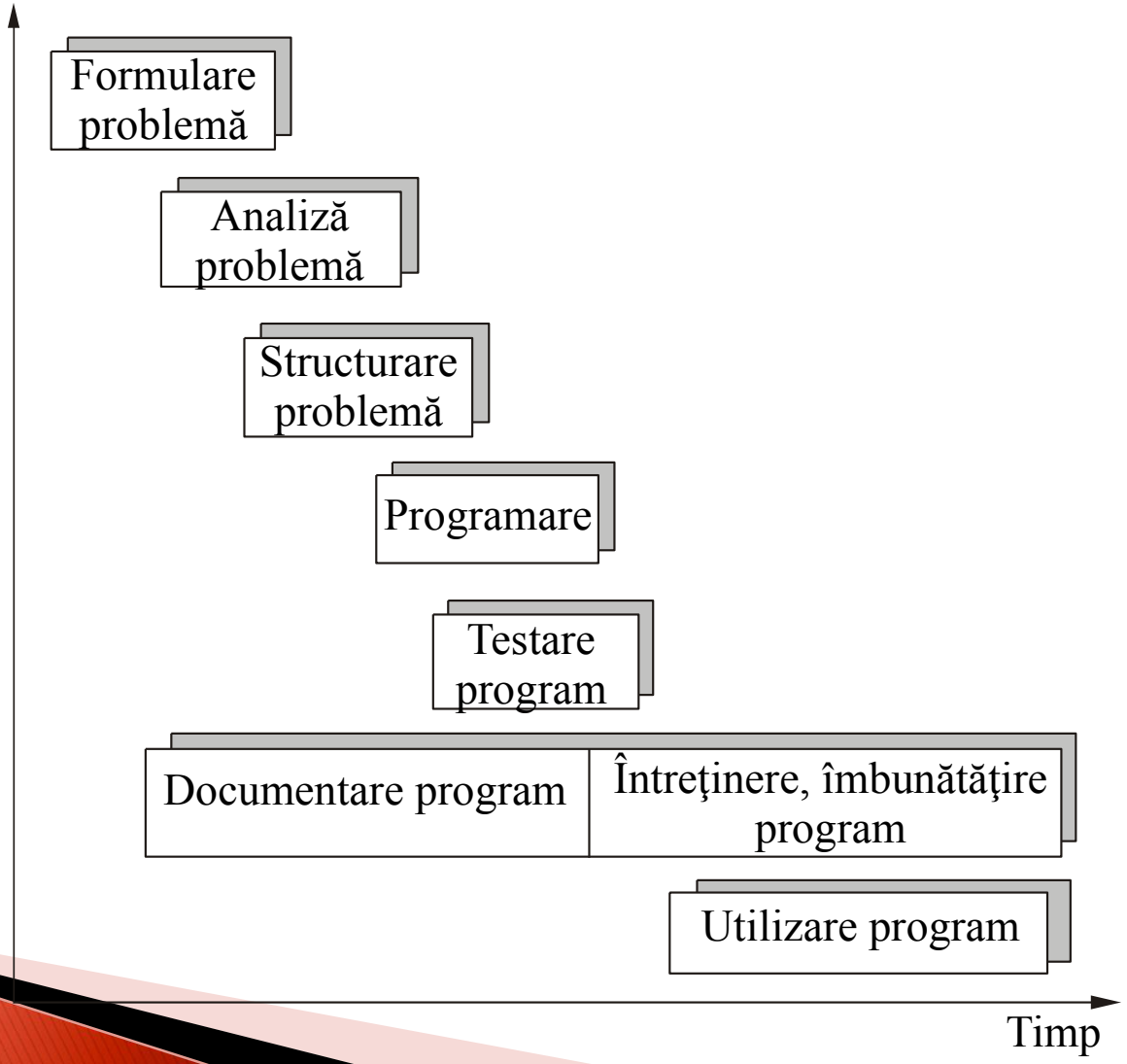
Algoritmi si scheme logice

Autori: Bogdan Gherman, Calin Vaida, Doina Pisla

Introducere

Între noţiunea de **program** şi cea de **algoritm** din matematică există o strânsă legătură. **Exprimând în program** ce trebuie să facă un calculator, **programatorul descrie de fapt un algoritm**, a cărui execuţie este încredinţată calculatorului. Orice program descrie o **transformare** a unei mulţimi de **valori cunoscute** (**datele de intrare**) într-o altă mulţime de **valori** (**datele de ieşire**). Execuţia lui de către sistemul de calcul presupune transmiterea datelor iniţiale din fişierul de intrare în memorie, efectuarea unor calcule cu datele în memorie şi transmiterea rezultatelor din memorie în fişierul de ieşire.

Rezolvarea unui proiect tehnico-ştiinţific pe calculator



Rezolvarea unui proiect tehnico-ştiinţific pe calculator – etape

Formulare problemă:

- studiul problemei date spre rezolvare de către utilizator;
- stabilirea sarcinilor ce trebuie îndeplinite;
- formularea modelului matematic.

Analiză problemă:

- descompunerea proiectului în părţi componente;
- căutare metode optime de rezolvare;
- determinare algoritmi de rezolvare.

Structurare problemă:

- transpunerea algoritmilor într-o formă standardizată cum ar fi schema logică sau schema bloc.

Programare:

- transcrierea schemei logice într-un limbaj de programare.

Rezolvarea unui proiect tehnico-ştiinţific pe calculator – etape

Testare program:

- implementarea programului pe calculator;
- crearea programului executabil;
- identificarea eventualelor erori;
- corectarea erorilor.

Documentare program:

- scrierea liniilor de comentarii explicative în program;
- crearea documentaţiei de utilizare a programului.

Întreţinere, îmbunătăţire program:

- efectuarea unor îmbunătăţiri în program;
- optimizarea programului.

Utilizare program:

- folosirea rezultatelor în practică.

Algoritmi – prezentare

Algoritmul reprezintă ansamblul regulilor care duc la soluţionarea modelului matematic al unei probleme date într-un număr cât mai mic de paşi şi cât mai exact. Cuvântul algoritm este de origine arabă. El derivă din numele matematicianului "*Abu Ja'far Mohammed ibn Musa al-Kahowârizmî*" care a scris o carte "*Kitab al jabr w'al-muqabala*".

Algoritmi – prezentare

Algoritmi		
Caracteristici	Tipuri	
claritate	numerici	nenumerici
generalitate	algebră lineară	sortare
eficienta	integrare de ecuatii diferentiale	căutare
corectitudine	sisteme de ecuatii neliniare	complecsi
	metode de optimizare	prelucrare simbolică

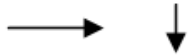
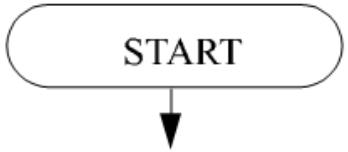
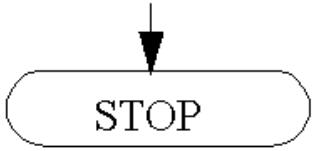
Algoritmi – prezentare

În vederea descrierii algoritmilor de calcul s-au pus la punct două forme grafice de reprezentare:

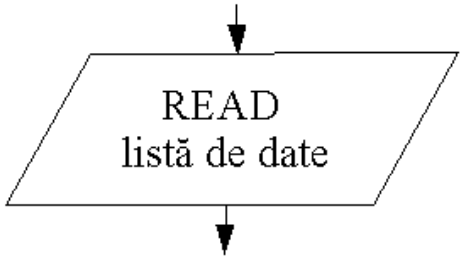
a. **Schema logică** reprezintă un limbaj grafic cu un înţeles foarte clar, cu ajutorul căruia se reprezintă algoritmii într-o formă mult mai uşor de înţeles pentru programator.

b. **Schema bloc** reprezintă totalitatea blocurilor structurale cu ajutorul cărora se descrie un program în succesiunea etapelor prevăzute în algoritm. Este important de menţionat că în cadrul schemelor bloc două blocuri structurale se pot găsi unul după celălalt, unul lângă altul, unul în altul, dar este exclusă suprapunerea acestora.

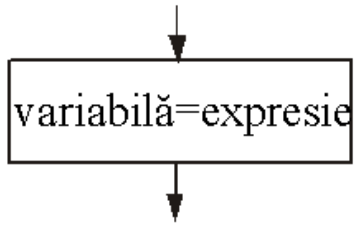
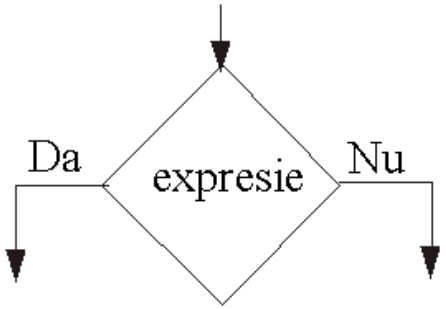
Scheme logice – reprezentare (1)

Simbol		Semnificație
Linii de flux		Indică sensul de parcurgere al etapelor succesive.
Simboluri terminale		Marchează începutul schemei logice.
		Marchează sfârşitul schemei logice.

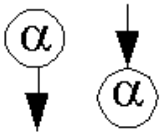
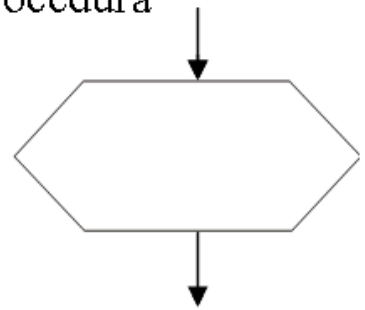
Scheme logice – reprezentare (2)

Simbol		Semnificație
Blocuri intrare- ieşire		Citeşte datele conținute în "lista de date".
		Scrie datele conținute în "lista de date".

Scheme logice – reprezentare (3)

Simbol	Semnificație
Bloc de calcul sau de atribuire 	Efectuează calcule. Se evaluează expresia din membrul drept iar valoarea ei este atribuită elementului din membrul stâng.
Decizia logică 	Pune condiții. Se evaluează expresia logică. Dacă aceasta este adevărată, secvențele continuă pe ramura Da iar în caz contrar continuă pe ramura Nu.

Scheme logice – reprezentare (4)

Simbol	Semnificație
Conectorul 	Leagă două linii de flux între ele.
Blocurile de procedură 	Grupează o serie de secvențe de calcul ca un algoritm independent (procedură sau funcție). Acest grup de calcule nu se detaliază în cadrul schemei logice care îl conține.

Algoritmi – prezentare

Atât schemele logice cât şi schemele bloc se bazează pe structuri elementare care permit abordarea oricărei probleme, de la simplu la complex, printr-o detaliere cu paşi succesivi. Avantajele unor astfel de structuri rezultă pe de o parte prin lizibilitatea lor, iar pe de altă parte prin simplificarea testării şi eliminarea greşelilor, permiţând, în acelaşi timp, munca în echipă a programatorilor.

În vederea programării în limbaje structurate cum ar limbajul C, C++ sau Pascal este necesară elaborarea unor scheme structurate. O schemă structurată se bazează pe teorema:

Orice algoritm cu o intrare şi o ieşire poate fi descris cu ajutorul a trei structuri de bază:

secvenţa ***decizia*** ***iteraţia.***

!!! Nu uitati !!!

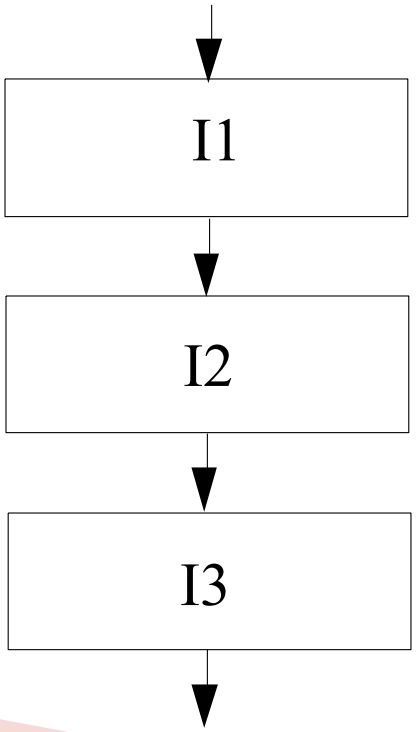
Orice algoritm (!!Oricat de complicat ar fi!!) cu o intrare şi o ieşire poate fi descris cu ajutorul a trei structuri de bază:

- *secvenţa;*
- *decizia;*
- *iteraţia.*

Scheme logice – Secventa

Secvenţa

Secvenţa reprezintă o succesiune de instrucţiuni.



BLOC STRUCT

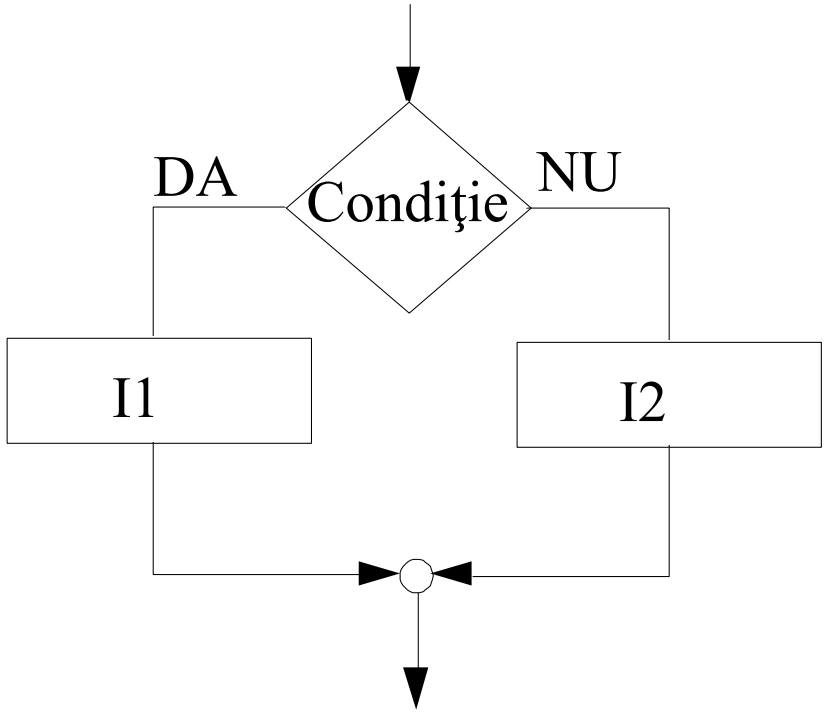
BLOC STRUCT

BLOC STRUCT

Scheme logice – Decizia

Decizia

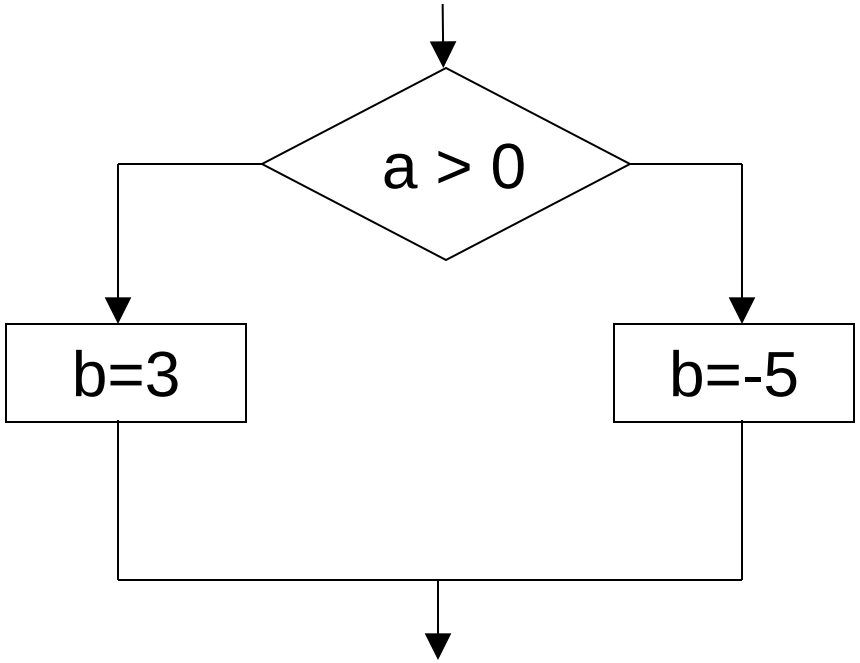
Decizia arată că secvenţele se continuă pe una sau alta dintre ramuri după cum condiţia impusă materializată printr-o expresie logică este îndeplinită sau nu.



a. Decizia în schema logică

Scheme logice – Decizia

Decizia – exemple



Exemplu numeric

a = -3 $\Rightarrow$ b=-5

a = 8 $\Rightarrow$ b=3

Scheme logice – Iteratia

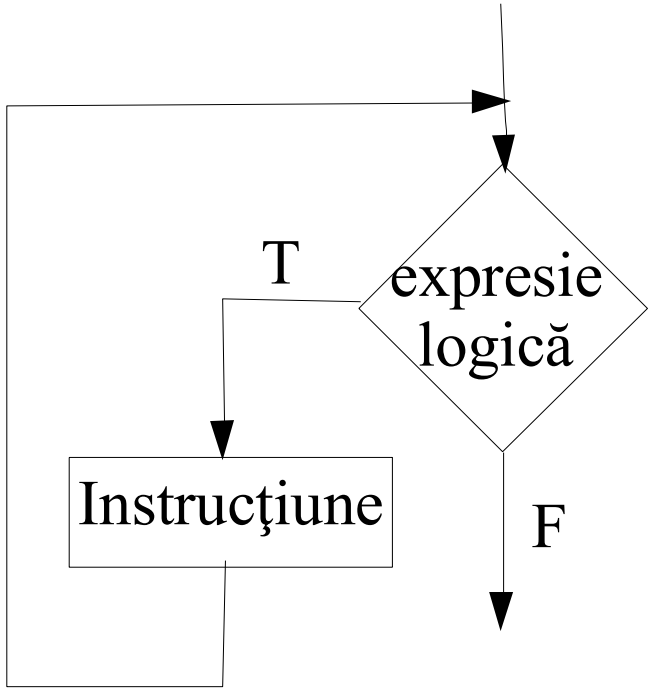
Iteraţia reprezintă structura repetitivă care permite execuţia unui grup de instrucţiuni atât timp cât condiţia impusă este îndeplinită.

Există două tipuri de iteraţii:

- cu test iniţial;
- cu test final.

Scheme logice – Iteratia

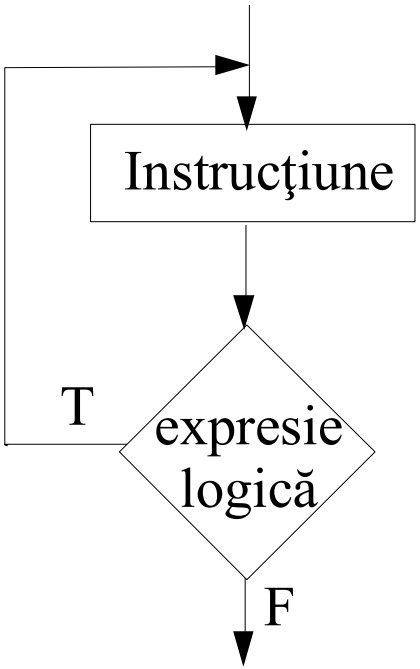
Iteraţia cu test iniţial este numită şi structura **WHILE-DO**. În cadrul acestei structuri se evaluează mai întâi valoarea expresiei logice. **Instrucţiune** se execută în mod repetat atât timp cât valoarea expresiei logice este adevărată (True). În caz contrar (False) se trece la secvenţa următoare din schema logică sau schema bloc.



a. Iteraţia cu test iniţial în schema logică b

Scheme logice – Iteratia

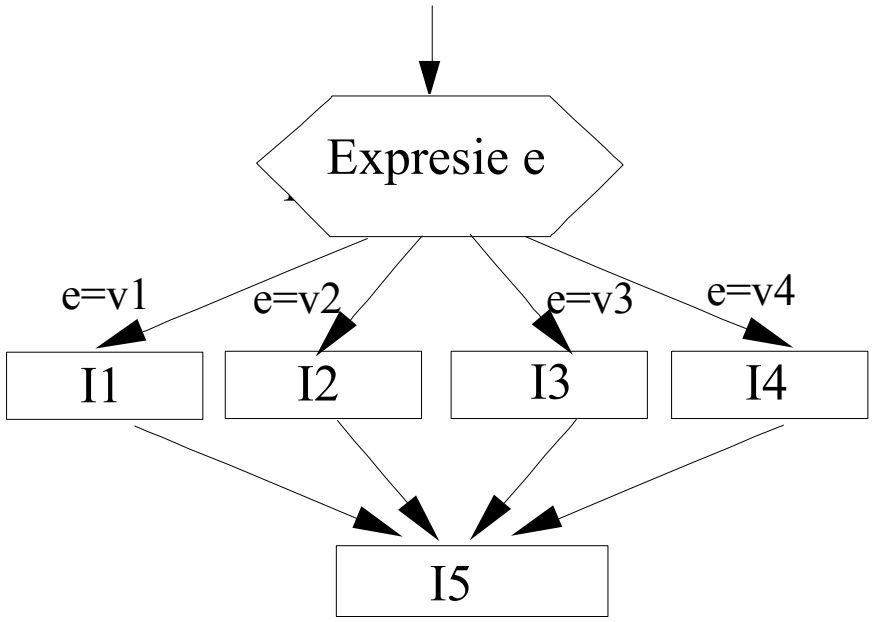
Iteraţia cu test final este numită şi structura **DO-WHILE**. În cazul acestei instrucţiuni se execută mai întâi **Instrucţiune** după care se evaluează expresia logică. Până când expresia logică este adevărată se revine şi se execută din nou **Instrucţiune**. În caz contrar se trece la secvenţa următoare.



a. Iteraţia cu test final în schema logică b.

Scheme logice – Structura selectiva

În cazul acestei structuri se evaluează mai întâi valoarea expresiei e. Dacă ea coincide cu valoarea vi, i=1,4 atunci se execută Ii corespunzătoare cazului nr. i. După execuţia instrucţiunii nr. i se trece la instrucţiunea următoare structurii selective.



a. Structura selectivă în schema logică b. St

Exemple (1)

1. Sa se citeasca doua numere, sa se calculeze suma acestora si sa se afiseze rezultatul.
2. Sa se citeasca doua numere, **a** si **b**, si daca primul este mai mare ca zero (**a>0**) sa se calculeze suma lor iar daca nu, sa se calculeze diferenta intre primul si al doilea.
3. Sa se citeasca doua numere, **a** si **b**, si sa se calculeze produsul lor daca ambele sunt diferite de zero.
4. Sa se citeasca un numar **a**, pana cand se introduce o valoare mai mare ca zero, si apoi sa se calculeze radacina lui patrata. (solutie cu iteratie cu test initial si test final)

Exemple (2)

1. Se citesc doua numere, **c** si **d**. Daca primul numar este divizibil cu doi, sa se calculeze catul si restul impartirii numarului mai mare la numarul mai mic.

!!!Notatii

a|b inseamna a divide pe b (3|6, 2|8, 11|121)

a div b inseamna catul impartirii intregi a lui a la b

(5 div 2 = 2, 7 div 4 = 1)

a mod b inseamna restul impartirii intregi a lui a la b

(5 mod 2 = 1, 7 mod 4 = 3)

1. Se citeste un numar care reprezinta un an. Sa se verifice daca anul este bisect sau nu.

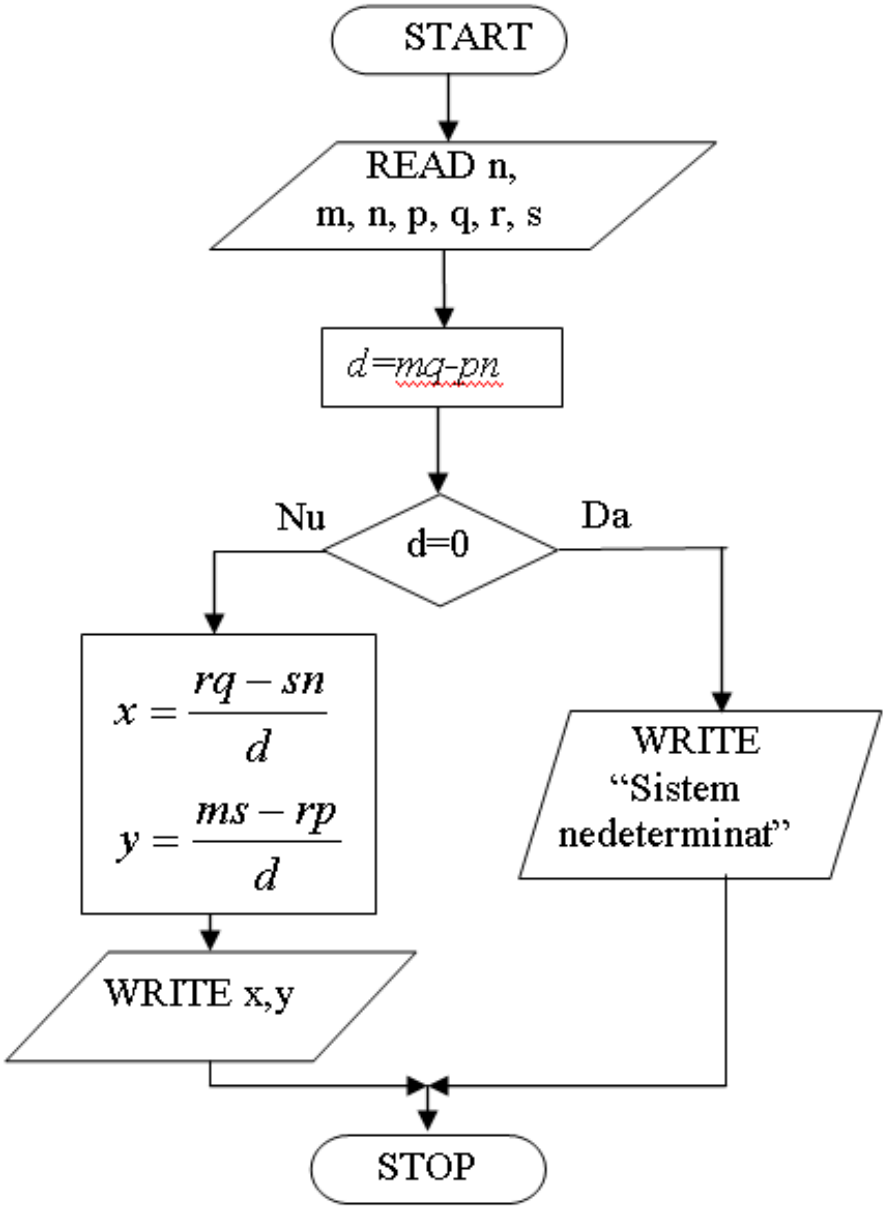
Aplicatii

Aplicatia 1. Să se elaboreze schema logică pentru rezolvarea unui sistem de ecuatii de gradul I de forma:

$$\begin{cases} mx + ny = r \\ px + qy = s \end{cases}$$

1. Se citesc coeficienţii m, n, p, q, r şi s.
2. Se calculează determinantul sistemului:
   $d=mq-pn$
3. Dacă d=0 se trece la pasul 6, iar în caz contrar se trece la pasul 4.
4. Se calculează
   $$x = \frac{rq - sn}{d}$$
   $$y = \frac{ms - rp}{d}$$
5. Se afişează x şi y şi STOP.
6. Se afişează un mesaj: “Sistem nedeterminat”.

Aplicatii


START
READ n,
m, n, p, q, r, s
d=mq-pn
Nu
d=0
Da
$x = \frac{rq - sn}{d}$
$y = \frac{ms - rp}{d}$
WRITE x,y
WRITE
"Sistem
nedeterminat"
STOP

Exemplu functii

Sa se determine valoarea functiei:

$$f(x)=\begin{cases} e1, & x \le a \\ e2, & x > a \end{cases}$$

Pentru un **x** citit dat.

Sa se determine valoarea functiei:

$$f(x)=\begin{cases} e1, & x \le a \\ e2, & a < x \le b, \quad a < b \\ e3, & x > b \end{cases}$$

Pentru un **x** citit dat.

Aplicatii

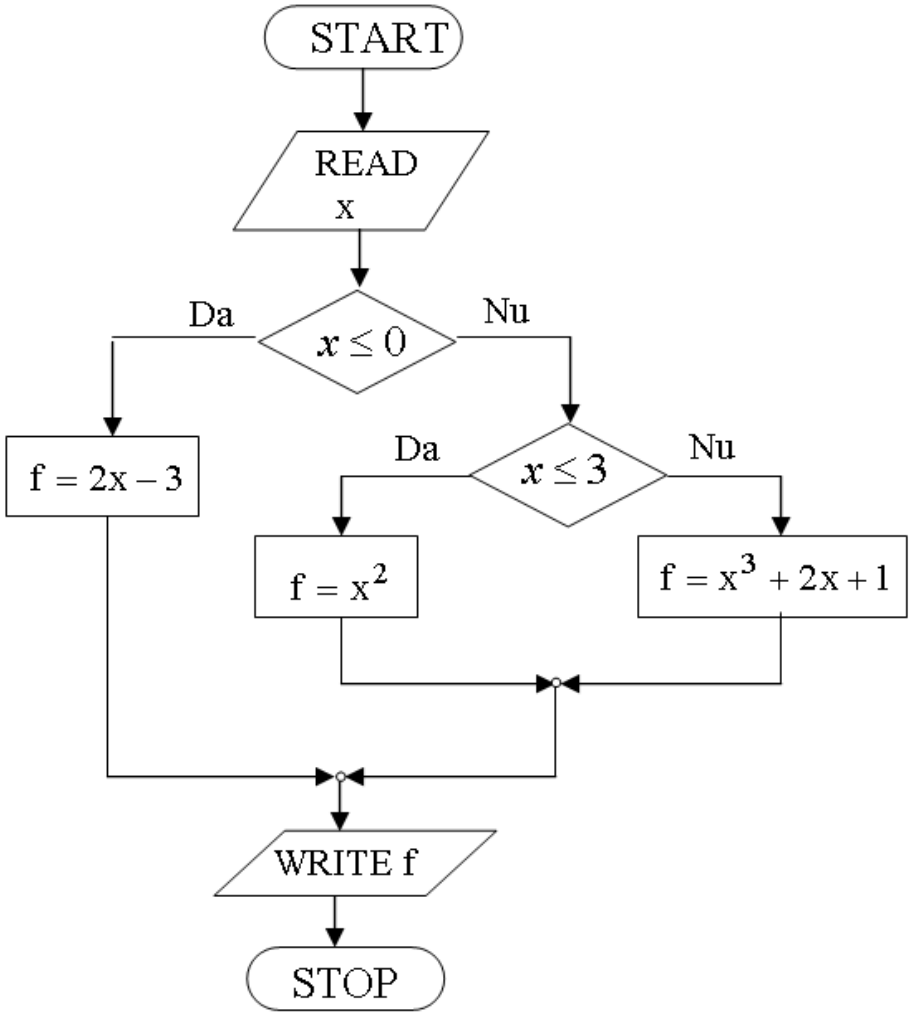
Aplicatia 2. Se consideră functia:

$$f(x)=\begin{cases}2x-3, & x\le 0\\ x^2, & 0<x\le 3\\ x^3+2x+1, & x>3\end{cases}$$

Se cere să se elaboreze schema logică pentru calculul valorii funcţiei f într-un punct x dat. Se va tipări rezultatul obţinut.

Aplicatii

1. Se citeşte valoarea argumentului x.
2. Dacă $x \leq 0$ se calculează valoarea lui $f = 2x - 3$ şi se trece la pasul 5, iar în caz contrar se trece la pasul 3.
3. Dacă $x \leq 3$ se calculează valoarea lui $f = x^2$ şi se trece la pasul 5 iar în caz contrar se trece la pasul 4.
4. Se calculează valoarea lui $f = x^3 + 2x + 1$.
5. Se afişează valoarea lui f.



Exemplu functii

Sa se determine valoarea functiei:

$$f(x)=\begin{cases} e1, & x \le a \\ e2, & a < x \le b, \quad a < b \\ e3, & x > b \end{cases}$$

Pentru un **x** definit in intervalul [**m,n**] parcurs cu pasul **p**.

Variatie: **x** definit in intervalul [**m,n**] impartit in **k** subintervale.

!!! Fata de cazul general **ATENTIE la nedeterminari** !!!

Aplicatii

Aplicaţia 3. Se consideră funcţia:

$$f(x)=\begin{cases}\dfrac{x^2+7}{x+11}, & x\le -6\\ \dfrac{x^2}{x^2-12x+1}, & -6<x\le -1\\ \dfrac{\sqrt{x+5}}{x(x-1)}, & x>-1\end{cases}$$

Să se elaboreze schema logică pentru calculul valorii funcţiei f în intervalul [a,b] parcurs cu pasul p. Se va tipări pentru fiecare punct al intervalului numărul de ordine, abscisa şi ordonata dacă funcţia se poate calcula şi numai numărul de ordine şi abscisa dacă funcţia nu se poate calcula.

Aplicatii

Algoritmul de calcul este următorul:

1. Se citesc datele de intrare: capetele intervalului a, b şi pasul p.
2. Se calculează numărul de puncte np în care se va evalua funcţia f.
3. Pentru valori ale lui i cuprinse între 1 şi np (pasul de variaţie al lui i este 1) se parcurg repetat următorii paşi:

3.1. Se calculează $x = a+(i-1)p$.

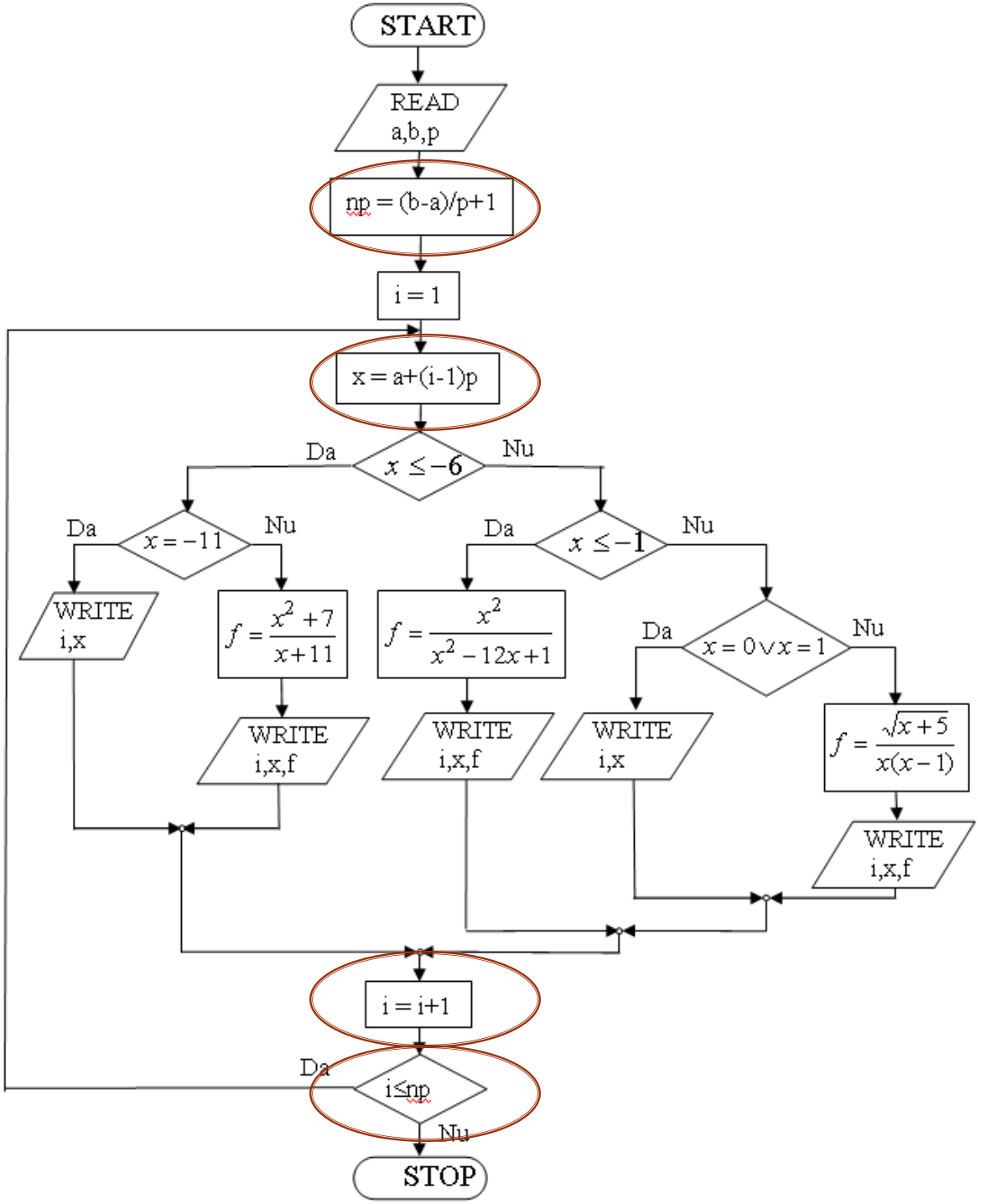
3.2. Dacă $x \leq -6$ se trece la pasul 3.3 iar în caz contrar se trece la pasul 3.4.

3.3. Dacă x=-11 se afişează i şi x, iar în caz contrar se calculează f şi se afişează şirul de valori i, x ,f. În ambele situaţii se revine la pasul 3.

3.4. Dacă $x \leq -1$ se calculează f şi se afişează şirul de valori i, x ,f. Se revine la pasul 3. În caz contrar se trece la pasul 3.5.

Dacă x=0 sau x=1 se afişează i şi x iar în caz contrar se calculează valoarea lui f şi se afişează i, x, f. În ambele situaţii se revine la pasul 3.

O posibila solutie!!!!



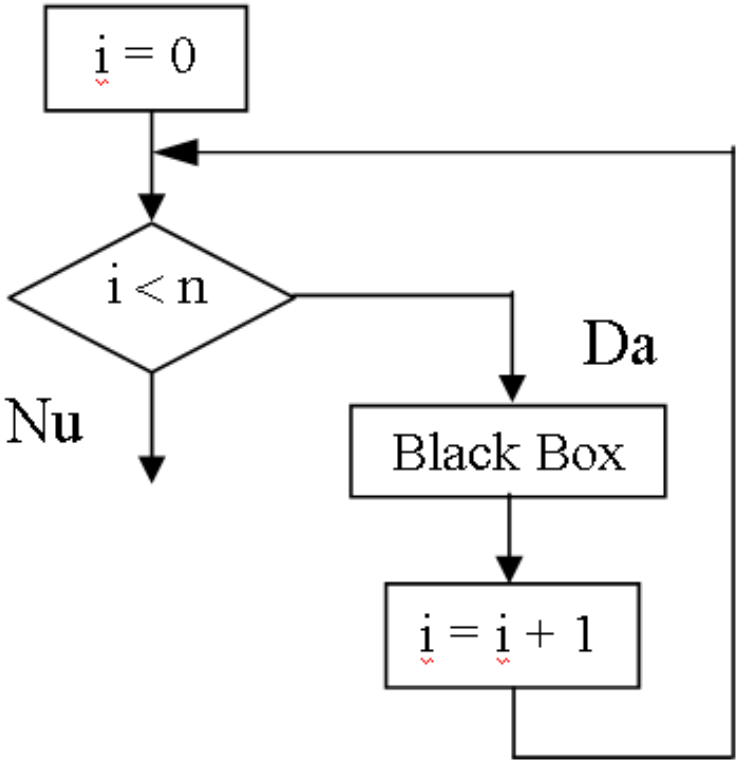
Siruri

Tabloul reprezintă o mulţime ordonată de elemente la care ne putem referi utilizând indici. Tipul comun al elementelor unui tablou este şi tipul tabloului respectiv. Şirurile sunt tablouri unidimensionale. Un element al unui şir se identifică cu e[i] unde i reprezintă poziţia acelui element în cadrul şirului:

$$e_0, e_1, \ldots e_{n-1}$$

Obs. În cazul şirurilor, **i** va lua valori între **0 şi n-1.**

Siruri – Mod uzual de parcurgere



Exempe cu siruri

1. Citirea unui sir.
2. Suma elementelor unui sir.

Aplicatii

Aplicaţia 1. Se consideră un şir format din n numere, notate cu $x_0, x_1, \ldots, x_{n-1}$.

Să se elaboreze schema logică pentru calculul mediei aritmetice a termenilor strict negativi şi media geometrică a termenilor strict pozitivi ai şirului. Se vor tipări elementele şirului, media aritmetică şi media geometrică dacă este posibil.

Aplicatii

Algoritmul de calcul este următorul:

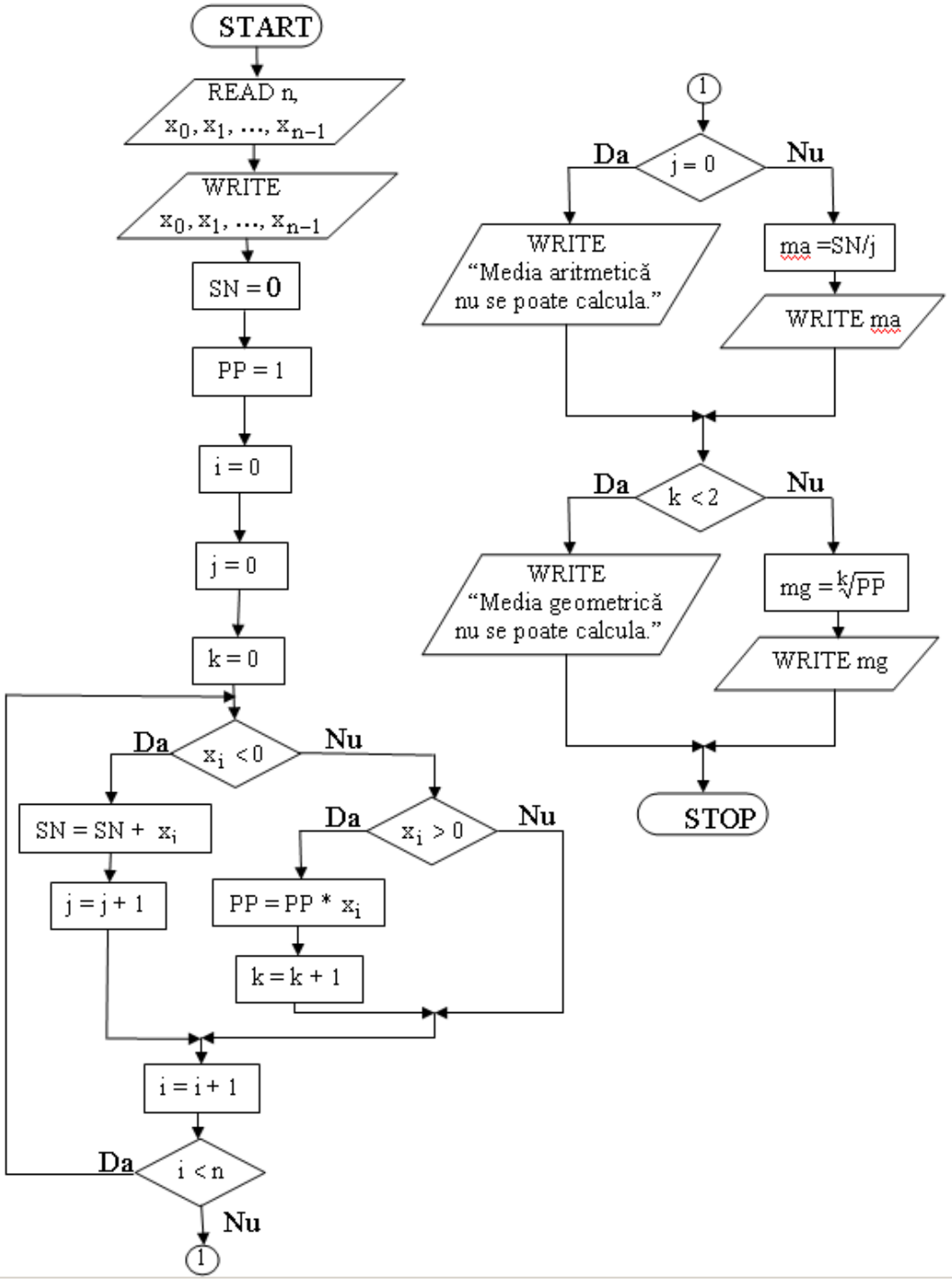
1. Se citesc numărul de termeni ai şirului şi termenii acestuia x_i, i=0,n-1.
2. Se tipăresc elementele şirului x_i, i=0,n-1.
3. Se iniţializează suma SN, produsul PP ai termenilor şirului, numărul de termeni negativi j şi numărul de termeni pozitivi k precum şi un numărător de termeni i.
4. Se parcurge şirul pentru valori ale lui i=0, n-1 în cadrul unui ciclu unde se compară fiecare element cu 0.

4.1. Dacă $x_i < 0$, se adaugă de fiecare dată valoarea elementului curent al şirului la SN şi se majorează cu 1 numărul elementelor de tipul corespunzător j.

4.2. Dacă $x_i > 0$ se înmulţeşte valoarea elementului curent cu produsul PP şi se majorează cu 1 numărul elementelor de tipul corespunzător j.

5. Dacă j=0 se tipăreşte un mesaj prin care se specifică că nu se poate calcula media aritmetică iar în caz contrar media aritmetică se calculează şi se afişează.
6. Dacă $k < 2$ se tipăreşte un mesaj prin care se specifică că nu se poate calcula media geometrică iar în caz contrar media geometrică se calculează şi se afişează.

Aplicatii


START
READ n, $x_0, x_1, \ldots, x_{n-1}$
WRITE $x_0, x_1, \ldots, x_{n-1}$
SN = 0
PP = 1
i = 0
j = 0
k = 0
$x_i < 0$
Da
Nu
SN = SN + x_i
j = j + 1
$x_i > 0$
Da
Nu
PP = PP * x_i
k = k + 1
i = i + 1
i < n
Da
Nu
1
1
j = 0
Da
Nu
WRITE "Media aritmetică nu se poate calcula."
ma =SN/j
WRITE ma
k < 2
Da
Nu
WRITE "Media geometrică nu se poate calcula."
mg = $\sqrt[k]{PP}$
WRITE mg
STOP

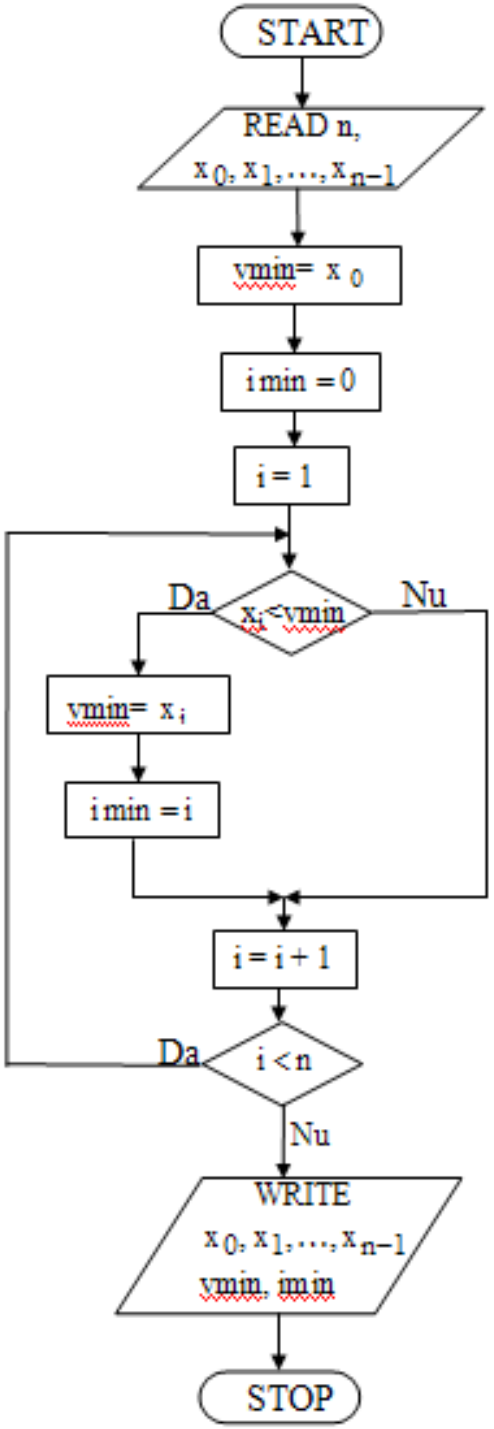
Aplicatii

Aplicaţia 2. Se consideră un şir format din n numere, notate cu $x_0, x_1, \ldots, x_{n-1}$.

Să se elaboreze schema logică pentru determinarea elementului minim din şir precum şi poziţia acestuia. Se vor tipări elementele şirului, elementul minim şi poziţia sa.

1. Se citesc numărul de termeni ai şirului şi termenii acestuia x_i, i=0,n-1.
2. Se presupune iniţial că elementul minim este primul element din şir iar indicele elementului minim este iniţial 0. Se iniţializează de asemenea un contor de termeni i.
3. Se parcurge şirul pentru valori ale lui i=0, n-1 în cadrul unui ciclu unde se compară pe rând fiecare element al şirului cu valoarea vmin înregistrată până la etapa respectivă. Dacă $x_i < vmin$, lui vmin i se atribuie valoarea acestui element iar imin primeşte valoarea indicelui i.
4. Se tipăresc rezultatele: elementele şirului x_i, i=0,n-1 valoarea vmin şi imin.

Schema logică este prezentată în figura 2.12.



Aplicatii

Aplicaţia 3. Se consideră un şir format din n numere, notate cu $x_0, x_1, \ldots, x_{n-1}$.

Să se elaboreze schema logică pentru ordonarea elementelor sirului. Se va tipari sirul nou rezultat.

Aplicatii

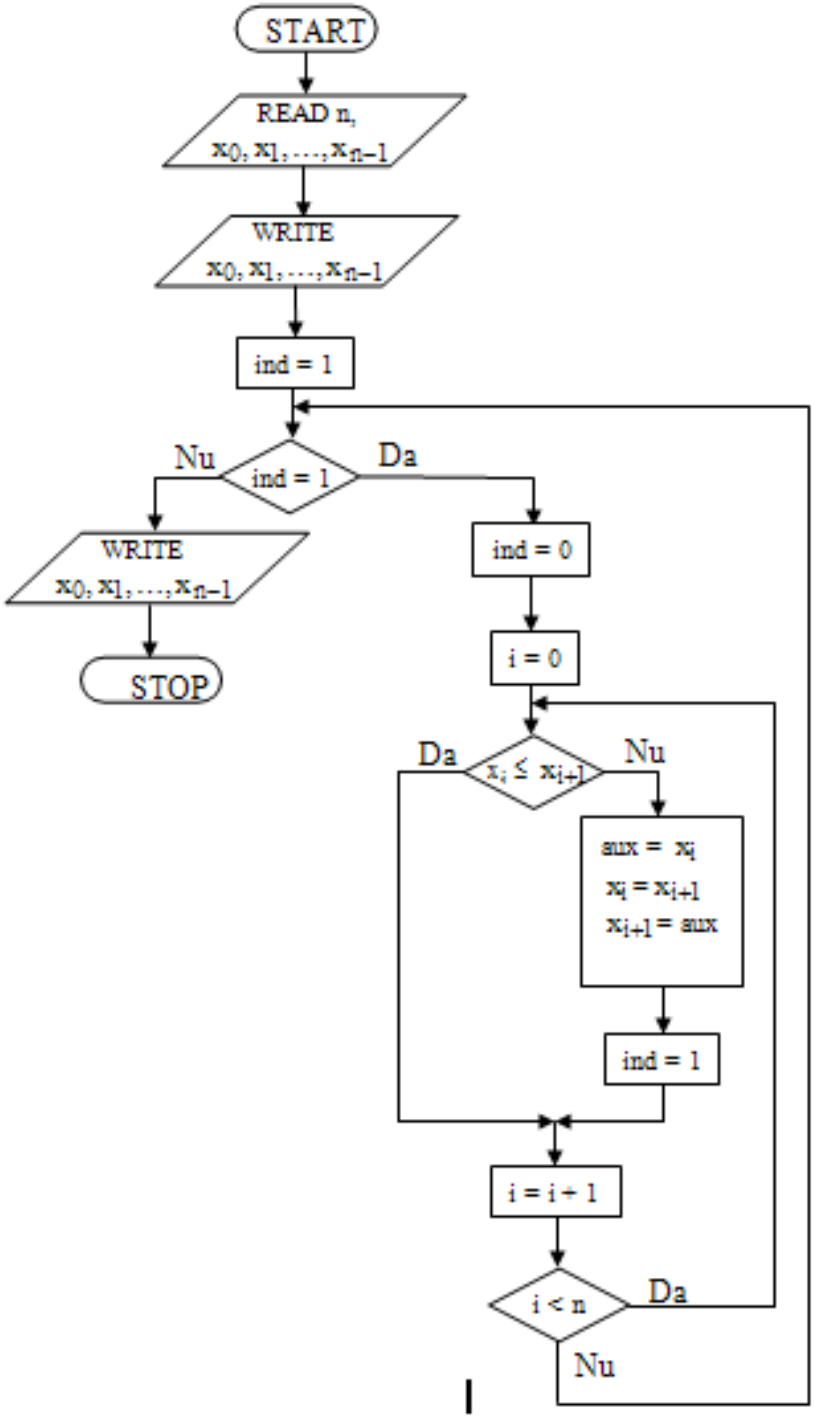
Algoritmul de calcul propus se bazează pe ordonarea perechilor de elemente vecine din şir în număr de n-1. Se parcurge şirul şi dacă două elemente vecine nu sunt ordonate crescător se interschimbă, semnalând aceasta prin modificarea variabilei ind la valoarea 1. Parcurgerea şirului se reia dacă s-a făcut cel puţin o interschimbare.

$$ind = \begin{cases} 0 \text{ daca au avut loc cele } n-1 \text{ relatii} \\ 1 \text{ daca cel putin o relatie nu a avut loc} \end{cases}$$

Etapele algoritmului sunt:

1. Se citesc numărul de termeni ai şirului şi termenii acestuia x_i , i=0,n-1.
2. Se tipăresc elementele şirului x_i , i=0,n-1.
3. Se iniţializează variabila ind la valoarea 1 (şir neordonat iniţial)
4. Se verifică dacă valoarea lui ind este egală cu 1. Dacă este adevărat se trece la pasul 5 iar în caz contrar se trece la pasul 6.
5. ind ia valoarea 0, se parcurge şirul pentru valori ale lui i=0, n-1 în cadrul unui ciclu unde se compară pe rând două elemente vecine prin relaţia $x_i \le x_{i+1}$. În caz afirmativ algoritmul de comparare se continuă iar în caz negativ se schimbă cele două valori şi ind=1. Se revine la pasul 4.
6. Se tipăresc elementele şirului ordonat crescător x_i , i=0,n-1.

Aplicatii


START
READ n,
$x_0, x_1, \ldots, x_{n-1}$
WRITE
$x_0, x_1, \ldots, x_{n-1}$
ind = 1
Nu
ind = 1
Da
WRITE
$x_0, x_1, \ldots, x_{n-1}$
STOP
ind = 0
i = 0
Da
$x_i \leq x_{i+1}$
Nu
aux = x_i
$x_i = x_{i+1}$
x_{i+1} = aux
ind = 1
i = i + 1
i < n
Da
Nu

Aplicatii

Aplicaţia 4. Să se citească un şir de la tastatură. Să se creeze, din acest şir, două şiruri noi, unul care va conţine elementele pozitive şi celălalt pe cele negative din primul şir. Să se afişeze cele două şiruri.

Aplicatii

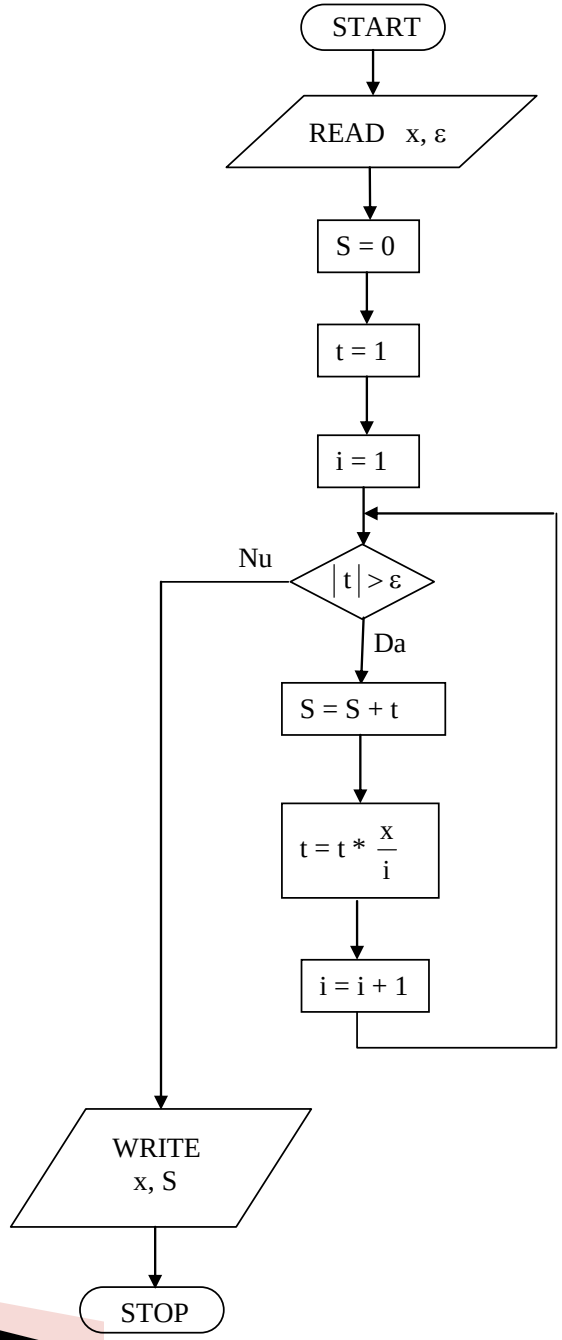
Generarea sirurilor prin dezvoltari in serie.

Să se elaboreze schema logică pentru calculul funcţiei $\mathbf{e^x}$ într-un punct x dat. Pentru calculul funcţiei se va utiliza următoarea dezvoltare în serie de puteri:

$$e^x = 1 + \frac{x}{1!} + \frac{x^2}{2!} + \frac{x^3}{3!} + \ldots + \frac{x^i}{i!} + \ldots$$

Din dezvoltarea în serie se vor considera toţi acei termeni care sunt mai mari în valoare absolută decât o eroare impusă, ε.

RASPUNS



Aplicatii

Să se calculeze valoarea funcţiei y=sin(x) folosind următoarea dezvoltare în serie de puteri, cu toţi termenii mai mari decât o valoare ε citită de la tastatură.

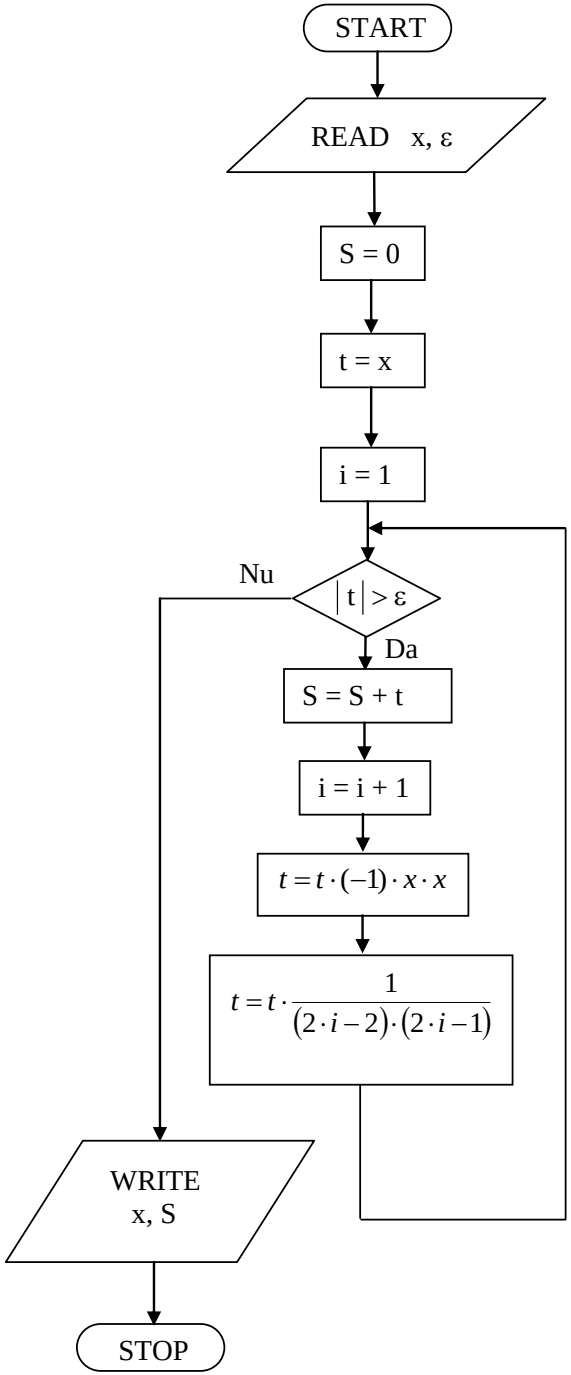
$$\sin(x) = \frac{x}{1!} - \frac{x^3}{3!} + \frac{x^5}{5!} - \cdots + (-1)^n \frac{x^{2n+1}}{(2n+1)!} + \cdots$$

!!! Stabilirea relatiei de recurenta

Variatie: in loc sa se calculeze toti termenii pana la o valoare epsilon, sa se calculeze valoarea functiei luand in considerare primii **m** termeni.

RASPUNS

Ce se schimba pentru a rezolva **variatia** problemei?



Exercitii

1. Sa se determine numarul de elemente pozitive, negative si nule dintr-un sir citit.
2. Se dă funcţia

$$f(x)=\begin{cases}\dfrac{x+7}{x-3}, & daca\ x\le 4\\ \dfrac{2x+3}{x-8}, & daca\ 4<x\le 10\\ x+1, & daca\ x>10\end{cases}$$

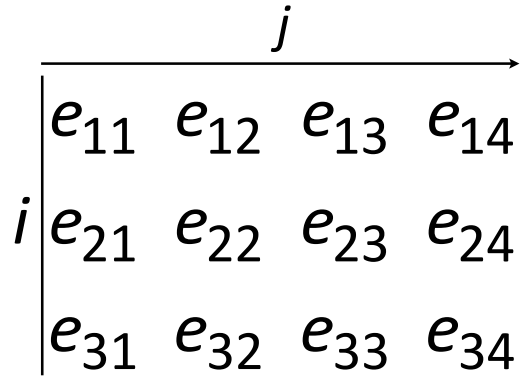
Să se calculeze valoarea funcţiei y=f(x) pe un interval [a,b] cu un pas p. Se vor considera a,b,x si p numere intregi.

Exercitii

3. Sirul lui Fibbonaci. Sa se genereze sirul lui Fibbobaci cu n termeni.
4. Se citeste un sir. Sa se formeze doua siruri noi, primul care sa contina elementele cu indice par a primului sir, iar al doilea pe cele cu indice impar.
5. Se citesc doua siruri de aceeasi dimensiune. Sa se formeze un sir nou care sa contina elementele pozitive din cele doua siruri, concatenate dupa indici.

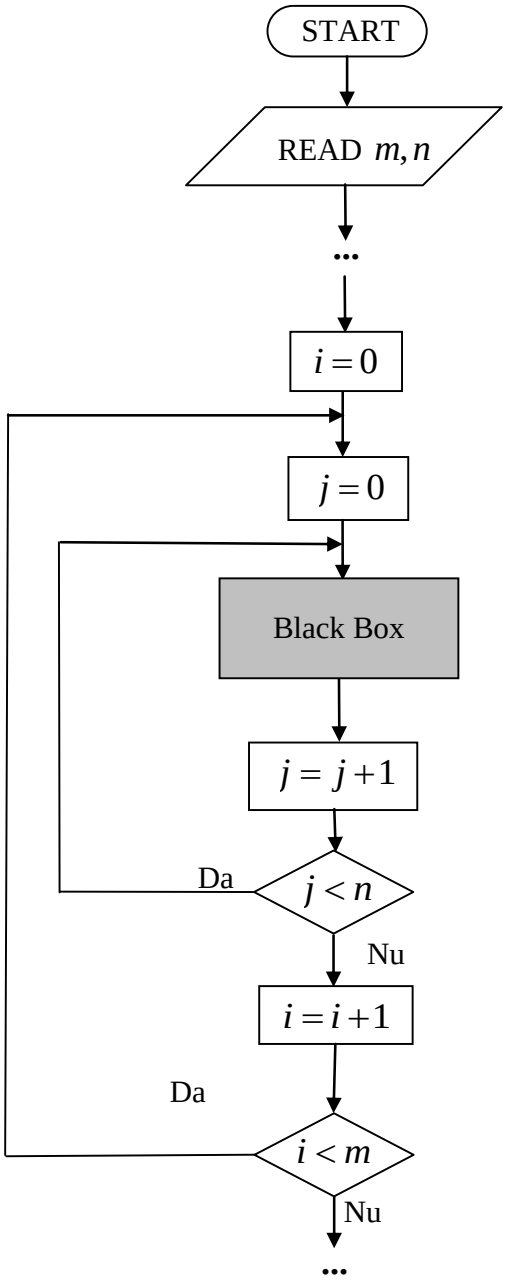
Matrice

Matricele sunt tablouri bidimensionale şi pot fi asemuite cu o secvenţa de mai multe şiruri. Dacă un element al unui şir se identifică cu **e[i]** unde i reprezintă poziţia acelui element în cadrul şirului, la matrice un element se identifica prin **e[i][j]** unde **i** reprezintă poziţia pe coloană (verticală) a acelui element, iar **j** reprezintă poziţia pe linie (orizontală) a acelui element. Dacă facem analogia cu şirurile, **i** reprezintă numărul şirului în care se află elementul, iar **j** reprezintă poziţia elementului în cadrul şirului **i**.



Matrice

Parcurgerea unei matrice



Varianta 1. Cu test final plecand de la zero

Matrice

Parcurgerea unei matrice

Varianta 2. Cu test initial

TABLA

Discutie **zero** versus **unu**!!!

Matrice

Aplicatii

- Calculati suma elementelor unei matrice;
- Calculati produsul elementelor pozitive;
- Calculati suma si produsul elementelor de pe o linie k citita
- Determinati elementul maxim si pozitia lui in matrice

Matrice

Matrice patratice

$$A_{[n][n]} = \begin{bmatrix} A_{[1][1]} & A_{[1][2]} & \cdots & \cdots & A_{[1][n-1]} & A_{[1][n]} \\ A_{[2][1]} & A_{[2][2]} & \cdots & \cdots & A_{[2][n-1]} & A_{[2][n]} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ A_{[n-1][1]} & A_{[n-1][2]} & \cdots & \cdots & A_{[n-1][n-1]} & A_{[n-1][n]} \\ A_{[n][1]} & A_{[n][2]} & \cdots & \cdots & A_{[n][n-1]} & A_{[n][n]} \end{bmatrix}$$

Determinati conditiile speciale

Matrice

- Determinati suma elementelor de pe cele doua diagonale
- Determinati produsul elementelor de sub cele doua diagonale
- Determinati elementul maxim de pe diagonala secundara
- Determinati transpusa unei matrice
  - In aceeasi variabila
  - Intr-o variabila noua