

PROGRAMARE STRUCTURATĂ IN BASIC

IOANA STAN

MARINEL ȘERBAN

OVIDIU ȘANDOR

HARALD SINGER



Ioana Stan
Marinel Șerban
Ovidiu Șandor
Harald Singer

PROGRAMARE STRUCTURATĂ IN BASIC

*Culegere de probleme
vol I*

Ediția III



CLUJ 1992

COORDONATOR: CLARA IONESCU

**Control științific: lector dr. Florian Boian (cap.1)
Vajda András (cap.2,3)**

Redactor: Clara Ionescu
Tehnoredactare computerizată: Vajda András
Grăfica computerizată: Eugen Ionescu și Claudia Bretan
Coperta: Sipos László

Bun de tipar: 18.10.1991.
Coli de tipar 16
ISBN 973-95118-2-1

PREFAȚA

O *culegere de probleme de programare* este în mod cert printre cele mai utile materiale didactice care se pot pune la dispoziția celor ce doresc să pătrundă în lumea calculatoarelor. Iar pătrunderea în această lume este o necesitate în momentul de față și va fi cu atât mai mult o necesitate în viitor. Fără calculatoare nu mai poate fi imaginată viața societății contemporane și nu am reuși să ne conectăm în mecanismele tot mai complexe și mai eficiente ale vieții mondiale.

Calculatoarele sînt deci importante, deosebit de utile, conduc la rezultate spectaculoase, dar stăpînirea lor nu este simplă. Ea presupune o muncă organizată, o gîndire logică, o bună cunoaștere în primul rînd a matematicii, dar și a altor domenii, precum și multă, multă pasiune.

Culegerea de față a luat naștere din pasiunea autorilor ei pentru calculatoare și din dăruirea lor pentru munca de a insufla și altora această pasiune. Sper ca rezultatele să fie pe măsură.

Se remarcă în mod deosebit structura acestei culegeri. Apare mai întîi o scurtă prezentare teoretică a unor noțiuni de bază utile cititorului, urmată de enunțurile problemelor, invitîndu-i astfel pe utilizatorii cărții la o gîndire independentă asupra soluției. Enunțurile sînt grupate conform programei școlare în vigoare, dînd astfel cititorului posibilitatea de a pătrunde *treptat* în tainele programării. În scopul sprijinirii activității de rezolvare a problemelor fiecare grup de enunțuri este precedat de o problemă rezolvată. Partea a 3-a conține sugestii pentru a porni la rezolvarea problemelor enunțate precum și algoritmi sub formă de scheme logice sau pseudocod, respectiv programele BASIC.

Gama problemelor conținute în culegere corespunde cunoștințelor dobîndite de elevi în clasele de informatică. Ea poate fi oricînd îmbunătățită, extinsă, completată. Problemele alese trebuie considerate ca un foarte bun și necesar "bagaj" de start.

Conf.dr.ing. Ioan Jurca,
Decanul Facultății de Calculatoare
și Automatică,

Institutul Politehnic "Traian Vuia"
Timișoara

NOTA

Pentru fiecare program, culegerea cuprinde listingul programului precum și una sau mai multe seturi de rezultate care furnizează imaginea rulării pe monitor. Pentru a obține rezultatele la imprimantă, este suficientă schimbarea tuturor instrucțiunilor PRINT cu instrucțiuni LPRINT (nu și instrucțiunile PRINT #0 și PRINT AT), eventual folosind următoarea rutină:

```
9000 FOR U=23755 TO PEEK 23627+256*PEEK 23628
9010 LET T=PEEK U : LET T1=PEEK (U+1)
9020 IF T=245 AND T1<>35 AND T1<>172 THEN POKE U,224
9030 NEXT U
```

unde:

23755 este, de obicei, adresa de început a programului BASIC;
PEEK 23627+256*PEEK 23628 este adresa zonei variabilelor;
245 este codul instrucțiunii PRINT;
35 este codul caracterului '#';
172 este codul instrucțiunii AT;
224 este codul instrucțiunii LPRINT.

Există mai multe posibilități de a scrie cu FORMAT rezultatele numerice, dintre care în culegerea de față sînt folosite următoarele trei:

- a) scrierea cu FORMAT a unui număr întreg folosind o rutină de aliniere la dreapta;
- b) scrierea cu FORMAT a unui număr real folosind și o rutină de aliniere pe marca zecimală;
- c) scrierea cu FORMAT folosind instrucțiunea PRINT AT.

Deosebirea dintre primele două metode și cea de-a treia constă în faptul că în ultima metodă nu poate fi schimbată instrucțiunea PRINT cu LPRINT, obținerea rezultatelor la imprimantă necesitînd modificarea programului.

I. PREZENTARE TEORETICA

1.1. NOȚIUNI GENERALE

Rezolvarea problemelor cu ajutorul calculatorului presupune o fază premergătoare în care se concepe algoritmul.

Un algoritm este o succesiune de calcule care, pentru o anumită clasă de probleme, din condițiile inițiale (cu ajutorul unor operații efectuate mecanic, fără aportul creator al omului) permite să se obțină soluția.

Un algoritm trebuie să satisfacă în general următoarele cerințe:

a) **claritate** - descrierea algoritmului se face precis, fără nimic arbitrar, fără ambiguități, conținând toate etapele de calcul, toate situațiile ce se pot ivi pentru a ajunge la soluție;

b) **generalitate** - un algoritm este util dacă rezolvă o clasă întreagă de probleme;

c) **finitudine** - algoritmul trebuie să furnizeze rezultate pentru orice set de date într-un număr finit (cât mai mic) de pași.

Operațiile care apar într-un algoritm sînt:

- operații de **intrare-ieșire** - datele de intrare se citesc iar datele de ieșire se afișează (se tipăresc);

- operații de **atribuire** - unei variabile i se atribuie valoarea unei expresii;

- operații de **decizie** - se determină valoarea de adevăr a unei expresii logice și, în funcție de rezultatul obținut, se ramifică execuția algoritmului.

Odată cu dezvoltarea informaticii a apărut un nou concept, acela de **programare structurată**. Ideea de bază constă în elaborarea algoritmilor folosind câteva structuri elementare avînd o **singură intrare și o singură ieșire**. Problema de rezolvat se descompune în subprobleme, a căror rezolvare duce la obținerea soluției problemei inițiale. Astfel orice algoritm apare ca o secvență liniară de structuri elementare.

Structurile elementare sînt:

- **structura liniară** - constă în execuția necondiționată a unei secvențe de instrucțiuni;

- **structura alternativă** - ramifică execuția algoritmului în funcție de valoarea de adevăr a condiției evaluate;

- **structura repetitivă** - constă în execuția repetată, de un număr finit de ori, a unei secvențe de instrucțiuni.

Subproblemelor din descompunerea unei probleme complexe le corespunde conceptul de **subprogram**. Subprogramele sînt de două tipuri: **proceduri și funcții**, referirea lor numindu-se **apel**. Diferența dintre ele constă în numărul valorilor calculate și returnate programului apelant: procedura transmite oricîte astfel de valori, în timp ce funcția transmite o singură valoare, acest lucru permițînd ca apelul ei să se facă chiar din expresia care conține valoarea calculată.

O noțiune de bază în programare este cea de **variabilă**. Printr-o variabilă se înțelege un ansamblu de patru elemente: **numele variabilei, tipul ei, valoarea ei și adresa din memorie la care este păstrată**.

Numele unei variabile este format din unul sau mai multe caractere (litere și cifre, primul fiind literă). Referirea unei variabile se realizează prin intermediul numelui său.

Exemple:

A, a, min, x1

Tipul variabilei indică mulțimea de valori posibile (întreg, real, caracter, boolean etc.), operațiile ce pot fi aplicate acesteia precum și modul de reprezentare în memoria calculatorului. Un tip de dată poate fi elementar sau structurat.

Tipurile elementare de date sînt tipurile numerice (întreg, real) și tipurile nenumerice (boolean, caracter etc.). Fiecare limbaj de programare permite folosirea unor tipuri elementare de date, specifice acestuia. În funcție de limbajul de programare, pentru fiecare variabilă folosită trebuie (sau nu) declarat explicit tipul acesteia.

Tipurile structurate de date sînt cele obținute prin gruparea unor tipuri elementare de date. Dintre cele mai utilizate tipuri structurate amintim **tabloul și înregistrarea**.

O variabilă de tip **tablou** este formată din mai multe elemente de același tip care se structurează pe una sau mai multe dimensiuni prin intermediul **indicilor**.

La declararea unui tablou se precizează **numele, tipul elementelor sale precum și mărimile dimensiunilor lui** (adică numărul de elemente).

Un tablou cu o singură dimensiune se numește **șir** iar un tablou cu două dimensiuni se numește **matrice**.

Un alt tip structurat de date este tipul **înregistrare** în care se pot grupa, după dorința programatorului, date de tipuri diferite.

Adresa din memorie este adresa fizică la care se află valoarea variabilei în memoria calculatorului. În marea majoritate a cazurilor, adresa este invizibilă pentru programator.

Valoarea variabilei este valoarea efectivă pe care o are aceasta la un moment dat. O variabilă are în orice moment o singură valoare care rămîne neschimbată pînă cînd aceasta se modifică printr-o **instrucțiune de atribuire sau de citire**.

1.2 REPREZENTAREA ALGORITMILOR

Pentru descrierea algoritmilor se pot utiliza:

- schema logică;
- pseudocodul;
- limbajul de programare.

Primele două forme de reprezentare sînt independente de tehnica de calcul, nu respectă o sintaxă rigidă, fiind utile programatorului doar în faza de proiectare a algoritmilor. Algoritmul descris într-un limbaj de programare constituie singura formă direct utilizabilă pe calculator, fiind un text codificat pe baza unor reguli sintactice bine determinate.

1.2.1. Schema logică

Schema logică este una din cele mai simple forme de reprezentare a algoritmilor. Este o reprezentare grafică, ce permite vizualizarea înălțurii și subordonării secvențelor de operații. Folosește simboluri grafice numite *blocuri* care prin forma lor indică tipul operației, iar prin textul conținut specifică semantica acesteia.

Elementele de bază sînt:



-*blocul terminal* -definește începutul sau sfîrșitul unui algoritm, respectiv al unei proceduri. Astfel, orice algoritm începe cu un bloc terminal care conține cuvîntul START și are unul care conține cuvîntul STOP.

Un algoritm structurat conține un singur bloc START la început și un singur bloc STOP, la sfîrșitul algoritmului;



-*blocul de citire-scriere* -reprezintă operațiile de citire sau scriere, definite prin cuvîntul citește respectiv scrie;



-*blocul de calcul* -reprezintă operația de atribuire;

Deosebim partea stîngă a atribuirii care este numele unei variabile, și partea dreaptă care este o expresie ce poate să conțină constante, nume de variabile și apeluri de funcții legate între ele prin operatori.

Exemplu:

$$A=B+C-3$$

Efectul instrucțiunii este evaluarea expresiei din partea dreaptă și atribuirea rezultatului obținut variabilei din partea stîngă. Pentru exemplul nostru, valoarea variabilei *B* (preluată de la adresa ei din memorie) se adună cu valoarea variabilei *C*; din valoarea rezultată se scade constanta 3, obținîndu-se rezultatul expresiei; această valoare se atribuie variabilei *A* (prin memorare la adresa acesteia).

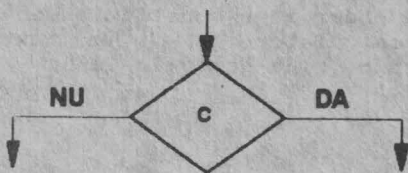
In programare apar des instrucțiuni de atribuire de forma:

$I=I+1$

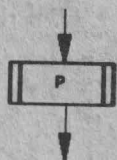
Diferența esențială față de exemplul anterior este aceea că variabila I apare atât în partea stângă cât și în partea dreaptă. Avem de-a face cu două valori ale lui I , cea dinaintea efectuării atribuirii (numită valoarea inițială) și cea de după atribuire (numită valoarea finală). Conform procedurii descris la exemplul anterior, valoarea finală va fi valoarea inițială plus 1.

Observație:

-semnul de atribuire poate fi \leftarrow , $:=$ sau simplu $=$. In această culeger , pentru simplificare vom folosi $=$



-blocul de decizie -reprezintă operația de decizie, are o intrare și două ieșiri, cite una pentru fiecare valoare de adevăr posibilă;



-blocul de procedură -reprezintă apelul unei proceduri. O procedură este o secvență de instrucțiuni descrisă printr-o schemă logică separată care are înscris în blocul terminal de la început numele procedurii și eventualii parametri iar în blocul terminal de la sfârșit cuvântul EXIT;

O

-blocul conector -reprezintă punctele de unire și revenire ale schemei;

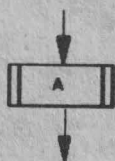


-blocul de continuare -reprezintă continuarea schemei logice pe altă pagină;



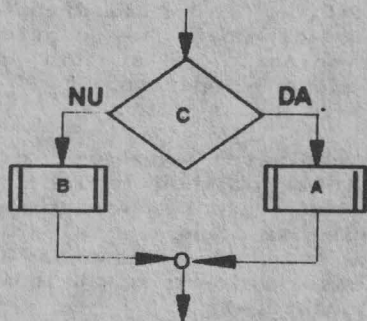
-săgețile -pun în evidență înlanțuirea logică a elementelor schemei logice.

Folosind elementele prezentate, structurile elementare care apar în schemele logice pot fi reprezentate grafic:



a) structura liniară

-unde A poate fi un bloc de calcul, un bloc de intrare-ieșire, un bloc de procedură sau o combinație liniară a acestora. Un astfel de bloc se execută necondiționat;



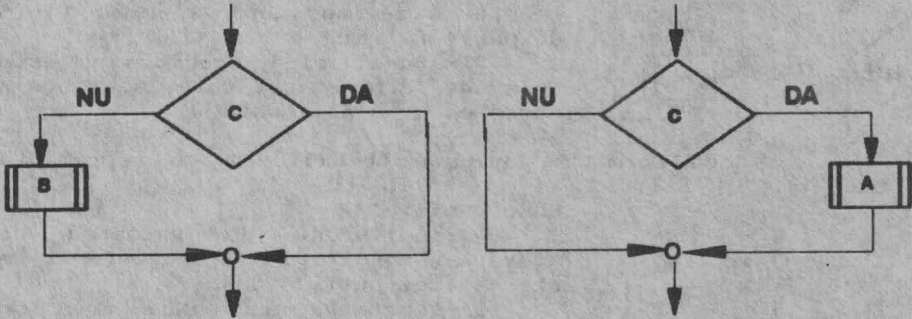
b) structura alternativă

-în cazul în care condiția C este îndeplinită, execuția programului continuă pe ramura DA , adică se execută secvența A de instrucțiuni, iar în cazul când condiția C nu este îndeplinită, execuția programului continuă pe ramura NU , adică se execută secvența B de instrucțiuni;

Observații:

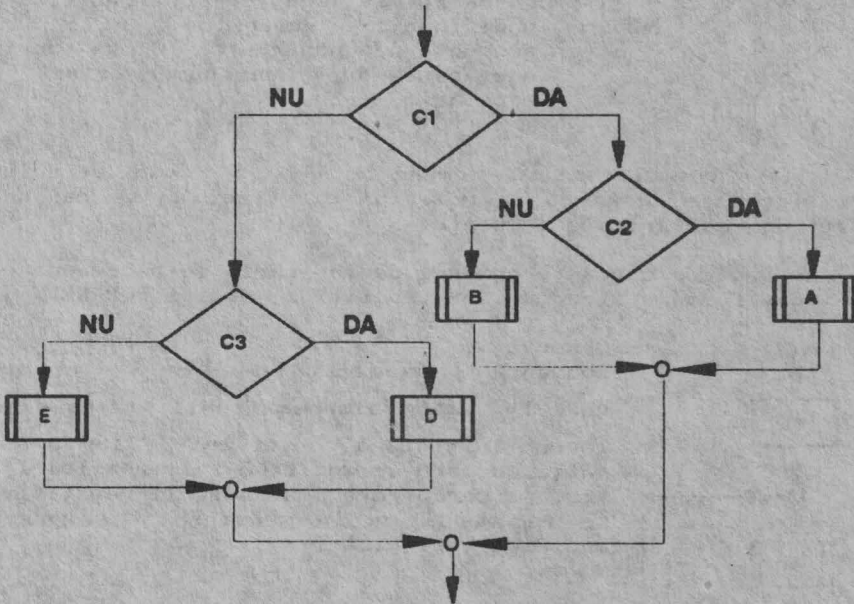
1)-intr-o astfel de structură, numită IF-THEN-ELSE, se execută fie o ramură fie cealaltă.

2)-această structură mai poate să apară în una din următoarele forme, în funcție de prezența sau absența secvențelor A sau B, caz în care se numesc *pseudoalternative*.



Observație:

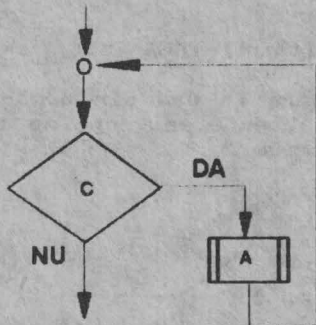
-blocurile A și B pot conține la rândul lor o altă structură alternativă, ceea ce duce la o schemă de geplu:



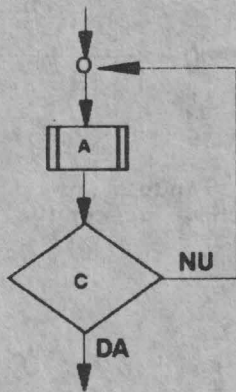
c) structura repetitivă

Această structură are mai multe forme, toate bazându-se pe executarea repetată a unei secvențe de blocuri. Întrucât un algoritm trebuie să aibă un număr finit de pași, pentru a controla repetarea parcurgerii unei secvențe, trebuie să existe o condiție (expresie logică) a cărei valoare la un moment dat să impună terminarea acestor repetări și ieșirea din structura repetitivă.

Prima formă se numește WHILE-DO și se reprezintă astfel:



A doua formă se numește **DO-UNTIL** și se reprezintă astfel:



Observații:

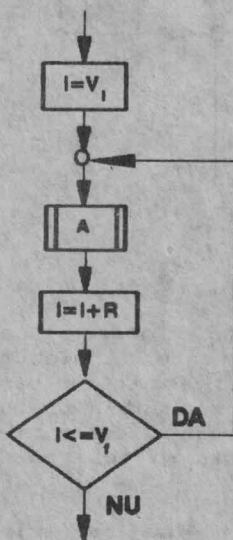
1) -indiferent de valoarea de adevăr a condiției C , secvența A se execută cel puțin o dată;

2) -repetarea secvenței are loc cît timp condiția C este falsă deci se iese din structura repetitivă atunci cînd condiția C devine adevărată;

3) -și în cazul structurii **DO-UNTIL** se menține obligativitatea modificării termenilor lui C în secvența A pentru ca C să devină adevărată după un număr finit de repetări.

O caracteristică comună a acestor două structuri repetitive este aceea că în momentul execuției nu se cunoaște numărul de repetări ale secvenței A .

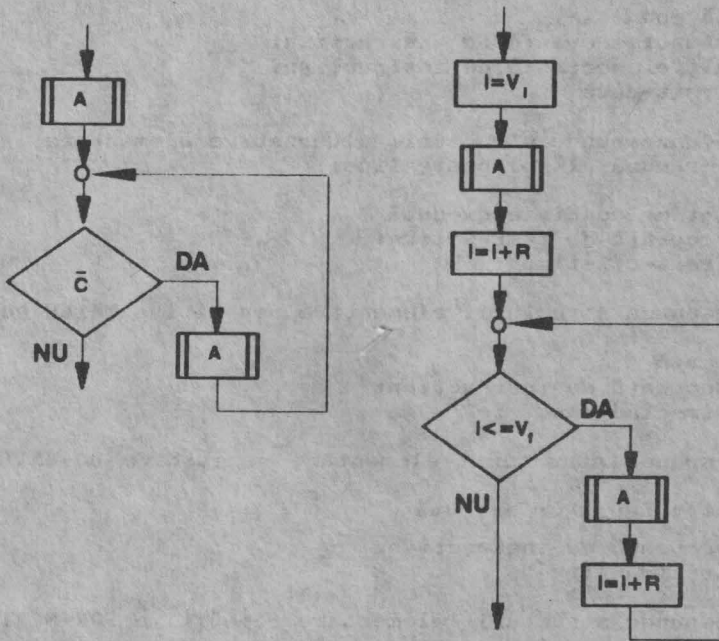
Dacă se cunoaște numărul de repetări, se recomandă cea de-a treia formă a structurilor repetitive, numită **FOR-NEXT**:



Observație:

-secvența A se repetă de $\left[\frac{|V_f - V_i|}{r} \right] + 1$ ori, unde V_f este valoarea finală, V_i este valoarea inițială a contorului I , iar r este rația cu care se modifică valoarea lui I la fiecare parcurgere a structurii repetitive; se recomandă ca în secvența A să nu se modifice variabila I .

Formele **DO-UNTIL** și **FOR-NEXT** se pot scrie cu ajutorul formei **WHILE-DO**, obținându-se schemele echivalente următoare:



Observație:

-diferențele dintre cele două tipuri de structuri au fost tratate detaliat în paragraful anterior.

1.2.2. Pseudocodul

O altă metodă de reprezentare a algoritmilor este *pseudocodul*. Inspirat din structurile elementare, această tinde spre un limbaj de programare structurat, neavând, totuși, o sintaxă rigidă ca un limbaj de programare. Setul de instrucțiuni propus de autorii prezentei lucrări este următorul:

algoritm *nume* este:

-este prima instrucțiune și are rolul de a denumi un program;

citește *listă*

-citește valorile variabilelor din *listă*; prin *listă* vom înțelege o succesiune de variabile separate prin virgulă;
-corespunde blocului de citire din schema logică;

scrie *listă*

-scrie valorile expresiilor din *listă*;
-corespunde blocului de afișare din schema logică;

variabilă ← *expresie*

-valoarea calculată pentru *expresie* se atribuie *variabilei*;

[*dacă condiție*
 atunci secvență de instrucțiuni
 altfel secvență de instrucțiuni
 sfârșit-dacă

-corespunde structurii alternative elementare;
-ramura altfel poate lipsi;

[*cît-timp condiție execută*
 secvență de instrucțiuni
 sfârșit-cît-timp

-corespunde structurii elementare repetitive **WHILE-DO**;

[*execută*
 secvență de instrucțiuni
 ieșim-cînd condiție

-corespunde structurii elementare repetitive **DO-UNTIL**;

[*pentru $i=v_i, v_f, r$ execută*
 secvență de instrucțiuni
 sfârșit-pentru

-corespunde structurii elementare repetitive **FOR-NEXT**;
-specificația *r* (rația sau pasul) este opțională, implicit *r=1*;

[*procedura nume(listă-parametri-formali) este:*
 secvență de instrucțiuni
 sfârșit-procedură

-corespunde declarării unei proceduri;
-*listă-parametri-formali* este opțională; conține numele acelor variabile care primesc valoare în momentul apelului de la *parametrii efectivi* și/sau returnează rezultat;

cheamă nume(listă-parametri-efectivi)

-corespunde apelului unei proceduri.

1.2.3. Limbajul de programare

Pentru a putea rezolva efectiv o problemă cu ajutorul calculatorului, trebuie să scriem algoritmul de rezolvare într-un limbaj înțeles de calculator. Un astfel de limbaj se numește limbaj de programare.

Din multitudinea de limbaje de programare existente, în cele ce urmează vom prezenta succint elementele limbajului **BASIC** (varianta pentru calculatoarele **ZX-SPECTRUM**), limbaj în care sînt scrise programele din această culegere.

1.3. Limbajul BASIC

1.3.1. Elementele limbajului

Alfabetul limbajului BASIC este format din:

- a)-literele mari și mici ale alfabetului englez;
- b)-cifrele sistemului zecimal;
- c)-caracterele speciale . : ; + - * / = < > " () \$ etc.

Cu ajutorul acestui alfabet se definesc cuvintele limbajului BASIC care formează vocabularul limbajului.

Cuvintele rezervate sînt cuvinte care denumesc instrucțiuni sau funcții predefinite. Numele lor nu poate fi folosit de utilizator în alt scop.

Identificatorii sînt cuvinte definite de utilizator pentru a denumi variabile sau funcții utilizator. Un identificator este format dintr-o succesiune de maximum 16 litere sau cifre, primul caracter fiind obligatoriu literă.

Exemple:

A, X1, x1n, var

O instrucțiune BASIC este formată dintr-o succesiune de cuvinte rezervate, identificatori și constante separate prin caractere speciale, prin ea cerindu-se executarea unei acțiuni.

O linie de instrucțiuni este compusă dintr-un număr de linie (între 0 și 9999) și din una sau mai multe instrucțiuni separate prin caracterul ':':

O secvență de linii formează un program BASIC.

Variabilele în BASIC pot fi de două tipuri: cele declarate implicit și cele declarate explicit. Cele declarate implicit pot fi la rîndul lor numerice sau șiruri de caractere. Variabilele șir de caractere nu se declară explicit, dar numele lor este format dintr-o singură literă urmată obligatoriu de caracterul '\$'. Variabilele declarate explicit se vor trata la instrucțiunea DIM.

Constantele în BASIC sînt fie valori numerice, fie șiruri de caractere între ghilimele ("").

1.3.2. Instrucțiunile limbajului ZX-BASIC

Instrucțiunea LET - este instrucțiunea de atribuire. Ea corespunde blocului de calcul și atribuire din schema logică respectiv instrucțiunii de atribuire din pseudocod. Forma generală a instrucțiunii este:

nr_linie LET variabilă=expresie

Efect:

-este același ca al instrucțiunii de atribuire din pseudocod.

Expresie conține identificatori, constante și apeluri de funcții separate prin operatori aritmetici: + - * / ^ . In expresie nu pot apare decit termeni de același tip cu variabila din partea stîngă a atribuirii. Intra expresie ordinea implicită de executare a operațiilor poate fi schimbată prin folosirea parantezelor.

Funcțiile aritmetice și trigonometrice predefinite in BASIC sint:

EXP(x)=e ^x	SQR(x)=√x
LN(x)=lnx	ABS(x)= x
INT(x)=[x]	SIN(x)=sinx
COS(x)=cosx	TAN(x)=tgx
ASN(x)=arcsinx	ACS(x)=arccosx
ATN(x)=arctgx	SGN(x)=sgnx

De asemenea, există predefinită constanta $PI=\pi$.

Instrucțiunea PRINT - este instrucțiunea de afișare și corespunde blocului de ieșire din schema logică, respectiv instrucțiunii scrie din pseudocod. Are următorul format:

nr_linie PRINT listă

Efect:

-constă în afișarea valorilor elementelor listei.

Elementele listei pot fi constante, variabile sau expresii. Acestea pot fi separate prin caracterele ',' (scrierea se face in continuare), ',' (scrierea se face la următorul TAB) sau '' (scrierea se face pe rîndul următor). Dacă valorile nu încap într-o linie, afișarea lor continuă în linia următoare. Poziția de afișare poate fi controlată cu ajutorul caracterelor de control AT sau TAB inserate în lista de elemente de afișat. Analog se pot controla culorile și modurile de scriere a caracterelor prin instrucțiunile INK, PAPER, FLASH, BRIGHT etc.

Instrucțiunea INPUT - este instrucțiunea de introducere a datelor și corespunde blocului de intrare din schema logică și instrucțiunii citește din pseudocod. Ea are următoarea formă:

nr_linie INPUT listă_de_variabile

Efect:

-programul va solicita valori pentru variabilele din listă_de_variabile, valori care pot fi numerice sau șiruri de caractere.

O facilitate deosebită a acestei instrucțiuni este aceea că în lista de variabile pot apare șiruri de caractere care vor fi afișate. Astfel, următoarea instrucțiune:

INPUT "Introduceți valoarea lui x: ",x

are ca efect afișarea mesajului respectiv, apoi solicitarea valorii variabilei x.

De asemenea, în lista de variabile poate să apară o variabilă între paranteze. Efectul constă în faptul că, în loc

să se aștepte introducerea valorii numerice, se va afișa valoarea variabilei respective. De exemplu, dacă variabila *i* are valoarea 4, instrucțiunea:

```
INPUT "Introduceti al ";(i);"-lea element al sirului a";a(i)
```

are ca efect afișarea mesajului:

```
"Introduceti al 4-lea element al sirului a"
```

după care sistemul va aștepta introducerea valorii respective.

Instrucțiunile **READ**, **DATA** și **RESTORE** - sînt instrucțiuni de introducere de date, dar nu de la tastatură. Instrucțiunea **DATA** are următoarea formă:

```
nr_linie DATA listă_valori
```

Efect:

-valorile specificate în *listă_valori* se introduc într-un bloc de date din care pot fi apoi citite cu instrucțiunea **READ**.

Valorile pot fi specificate în oricite instrucțiuni **DATA** deoarece aceste valori sînt trecute într-un singur bloc de date. Valorile dintr-o listă se despart prin caracterul ','. Pentru a atribui unor variabile valorile din instrucțiunile **DATA**, se folosește instrucțiunea **READ** cu următoarea formă:

```
nr_linie READ listă_variabile
```

Efect:

-variabilele din *listă_variabile* iau valorile din **DATA** în ordinea în care apar.

Pentru a da programatorului posibilitatea de a-și alege singur grupa de valori din blocul de date (adică instrucțiunea **DATA** care conține valorile respective) s-a prevăzut instrucțiunea **RESTORE** cu forma generală:

```
nr_linie RESTORE nr_linie_instrucțiune_DATA
```

Instrucțiunea **GOTO** -este instrucțiunea de salt necondiționat și are următoarea formă generală:

```
nr_linie GOTO alt_nr_linie
```

Efect:

-produce un salt necondiționat la linia *alt_nr_linie* din program.

Instrucțiunea **IF-THEN** - este instrucțiunea de test corespunzătoare blocului de decizie din schema logică. Forma ei este următoarea:

```
nr_linie IF cond THEN instrucțiune
```

Efect:

-în cazul în care condiția din instrucțiunea **IF-THEN** este îndeplinită, execuția continuă cu instrucțiunea specificată după **THEN** (această instrucțiune poate fi și o instrucțiune **GOTO** care impune un salt la o altă linie de program decît cea următoare); în cazul în care condiția nu este îndeplinită, execuția continuă cu linia următoare instrucțiunii **IF-THEN**.

Exemplu:

```
100 IF a<b THEN GOTO 200
110 instrucțiuni ramura NU
.....
190 GOTO 300
200 instrucțiuni ramura DA
.....
300 instrucțiunea următoare instrucțiunii IF-THEN
.....
```

Observații:

1)-o condiție este o expresie relațională care poate avea, la un moment dat, o singură valoare: ori ADEVARAT ori FALS; în general ea se obține prin combinarea operatorilor relaționali: <, >, =, <=, =, <> și a unor expresii aritmetice; de asemenea se pot folosi operatorii logici: NOT (negare), AND (și), OR (sau) și paranteze rotunde pentru a schimba prioritatea operatorilor;

2)-execuția instrucțiunii IF-THEN începe cu evaluarea condiției; se obține valoarea 0 pentru FALS respectiv 1 pentru ADEVARAT; în funcție de această valoare se continuă execuția pe una din ramuri;

3)-deoarece în limbajul BASIC nu sînt implementate instrucțiuni corespunzătoare structurilor elementare WHILE-DO și DO-UNTIL, simularea acestor structuri se realizează cu instrucțiunea IF-THEN.

Exemple:

1)-structura WHILE-DO se simulează în felul următor:

```
100 IF NOT cond THEN GOTO 200
110 secvență de instrucțiuni
.....
190 GOTO 100
200 instrucțiunea următoare structurii WHILE-DO
```

2)-structura DO-UNTIL va avea următoarea formă:

```
100 secvență de instrucțiuni
.....
200 IF NOT cond THEN GOTO 100
210 instrucțiunea următoare structurii DO-UNTIL
```

Instrucțiunea FOR-NEXT -este instrucțiunea care corespunde structurii repetitive cu același nume. Forma sa generală este:

```
nr_linie_1 FOR i=exp1 TO exp2 STEP exp3
.....
nr_linie_n NEXT i
```

Efect:

-variabila *i* va lua valoarea expresiei *exp1* după care se execută instrucțiunile dintre FOR și NEXT; apoi la valoarea lui *i* se adună valoarea expresiei *exp3* și dacă noua valoare *i* nu depășește valoarea expresiei *exp2*, se reia execuția instrucțiunilor dintre FOR și NEXT; execuția se repetă pînă cînd valoarea lui *i* depășește valoarea expresiei *exp2*.

Observații:

1)-se recomandă ca în instrucțiunile dintre FOR și NEXT să nu se modifice valorile termenilor din *exp1*, *exp2* respectiv

exp3 deoarece aceste modificări pot duce la rezultate imprevizibile sau la repetarea la infinit a ciclului **FOR-NEXT**;

2)-nu se vor efectua salturi din exteriorul ciclului **FOR-NEXT** în interiorul său deoarece în acest caz nu se cunosc valorile expresiilor *exp1*, *exp2*, *exp3* respectiv a contorului *i*;

3)-saltul din interiorul ciclului spre exterior este permis, dar programarea structurată evită astfel de soluții;

4)-pot să apară mai multe instrucțiuni **FOR-NEXT** una în alta, cu observația că ultimul **FOR** deschis se închide primul (se respectă principiul parantezelor);

5)-specificația **STEP** din instrucțiunea **FOR-NEXT** este opțională; valoarea implicită a pasului este 1.

Instrucțiunile **GO SUB** și **RETURN** - sînt instrucțiunile de apelare a unui subprogram respectiv de revenire la programul apelant. Forma lor generală este următoarea:

```
nr_linie1 GO SUB nr_linie_sub
.....
nr_linie_sub .....
.....
nr_linie2 RETURN
```

Efect:

-la întîlnirea instrucțiunii **GO SUB** se intrerupe execuția programului și se continuă cu execuția instrucțiunilor din subrutină, începînd cu linia *nr_linie_sub* care este numărul de linie al primei instrucțiuni din subprogram; se execută subrutina pînă la întîlnirea instrucțiunii **RETURN** cînd se reia execuția programului apelant de la instrucțiunea care urmează după **GO SUB**.

Observații:

1)-subrutinele se pot așeza oriunde în program;

2)-se observă că în **BASIC** nu există parametri formali și efectivi pentru a transfera valori numerice între programul apelant și subrutină;

3)-este indicat ca în subrutine să avem alte variabile decît cele din programul apelant; înaintea apelării subrutinei, respectiv la revenirea din subrutină trebuie să aibă loc schimbul de valori între variabilele care se transmit respectiv se returnează ,prin instrucțiuni de atribuire.

Instrucțiunile **DEF FN** și **FN** - sînt instrucțiunile de definire respectiv de apelare a unei funcții.

Definirea unei funcții se face în modul următor:

```
nr_linie DEF FN nume(param_formali)=expr
```

unde *nume* este numele dat funcției de către utilizator, *param_formali* este o listă de parametri formali, elementele listei fiind despărțite prin virgulă, iar *expr* este o expresie care calculează valoarea funcției; numele este format dintr-un singur caracter (eventual urmat de \$, dacă funcția este de tip caracter).

Apelarea funcției se face prin instrucțiunea:

```
FN nume(param_efectivi)
```

Apelul funcției poate să apară în orice expresie a unei instrucțiuni BASIC. Parametrii efectivi pot fi constante, variabile, expresii sau apeluri de alte funcții care au în momentul apelării valori cunoscute. Cu ajutorul parametrilor efectivi se calculează valoarea expresiei din DEF FN și se returnează valoarea obținută. De obicei definirea funcțiilor se face la începutul programului.

Instrucțiunea STOP - este instrucțiunea care determină oprirea execuției programului. Forma generală a instrucțiunii este:

nr_linie STOP

Efect:

-la întâlnirea acestei instrucțiuni execuția programului se oprește.

Instrucțiunea PAUSE - permite oprirea temporară a execuției unui program. Ea are următoarea formă generală:

nr_linie PAUSE n

unde *n* este intervalul de timp (exprimat în unități de 1/50 secunde) în care se întrerupe execuția programului; de exemplu, PAUSE 50 va opri programul pentru o secundă; un caz aparte este PAUSE 0 la apariția căreia execuția programului se suspendă pînă la apăsarea unei taste.

Instrucțiunea DIM - este instrucțiunea de declarare a unui tablou de date. Forma generală a instrucțiunii este:

nr_linie DIM nume(dimens)

unde *nume* este numele tabloului, format dintr-o literă, urmată eventual de caracterul '\$' în cazul unui tablou cu elemente de tip caracter, iar *dimens* este lista care conține dimensiunile tabloului, despărțite prin virgulă.

Exemplu:

DIM *m*(3,4) definește matricea *m* avînd 3 linii și 4 coloane; adresarea unui element al acestei matrice se face prin intermediul indicilor care pot fi constante, variabile sau expresii; o componentă a tabloului definit poate fi referită cu *m*(*expr1*,*expr2*) unde valorile indicelui *expr1* trebuie să fie cuprinse între 1 și 3, iar valorile expresiei *expr2* trebuie să fie cuprinse între 1 și 4.

Observație:

-instrucțiunea DIM trebuie să apară înaintea primei instrucțiuni care folosește un element al tabloului.

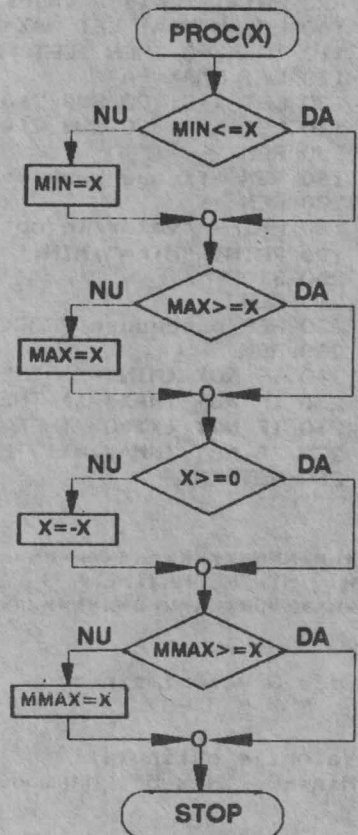
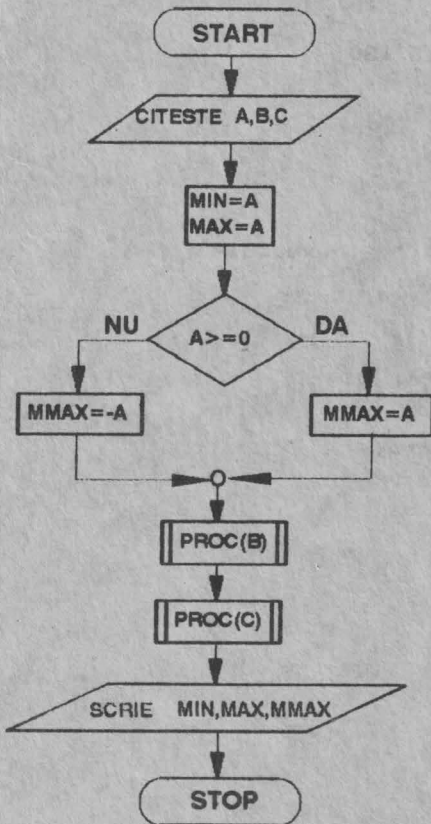
II. ENUNȚURI

1. STRUCTURI LINIARE ȘI ALTERNATIVE

Să se calculeze minimul, maximul și maximul modulelor a trei numere reale date.

INDICAȚIE:

După citirea celor trei numere a, b, c , se inițializează o variabilă pentru minim și una pentru maxim, ambele cu valoarea lui a , iar o a treia variabilă cu modulul lui a . Apoi se apelează de două ori o procedură (de parametru x), o dată cu parametrul efectiv b și a doua oară cu parametrul efectiv c , procedură în care x și modulul său se compară cu variabilele ce trebuie determinate, iar în cazul în care este necesar, se modifică valoarea acestor variabile.



```

algoritmul MINMAX este
citește a,b,c
min ← a
max ← a
dacă a≥0 atunci mmax ← a
    altfel mmax ← -a
sfârșit-dacă
cheamă PROC(b)
cheamă PROC(c)
scrie min,max,mmax
stop

```

```

procedura PROC(x) este:
dacă not (min≤x)
    atunci min ← x
sfârșit-dacă
dacă not (max≥x)
    atunci max ← x
sfârșit-dacă
dacă not (x≥0)
    atunci x ← -x
sfârșit-dacă
dacă not (mmax≥x)
    atunci mmax ← x
sfârșit-dacă
sfârșit-procedură

```

```

10 REM
20 REM *** MINIME SI MAXIME ***
30 REM
40 CLS
50 PRINT "*****"
60 PRINT "*   MINIME/MAXIME A 3 VALORI   *"
70 PRINT "*****"
80 INPUT "Introdu cele 3 valori a,b,c: " a=";A;"   b=";B;"
c=";C
90 PRINT "Cele 3 valori sint: " "   ";A;"   ";B;"   ";C"
100 LET MIN=A: LET MAX=A
110 IF A>=0 THEN LET MMAX=A: GO TO 130
120 LET MMAX=-A
130 LET X=B: GO SUB 210
140 LET X=C: GO SUB 210
150 REM
160 REM afisare rezultat
170 REM
180 PRINT "Valorile obtinute:"
190 PRINT "Min=";MIN;"   Max=";MAX;"   Maxmodul=";MMAX
200 STOP
210 REM
220 REM procedura PROC(X)
230 REM
240 IF NOT (MIN<=X) THEN LET MIN=X
250 IF NOT (MAX>=X) THEN LET MAX=X
260 IF NOT (X>=0) THEN LET X=-X
270 IF NOT (MMAX>=X) THEN LET MMAX=X
280 RETURN

```

```

*****
*   MINIME/MAXIME A 3 VALORI   *
*****

```

```

Cele 3 valori sint:
3   5   -8

```

```

Valorile obtinute:
Min=-8   Max=5   Maxmodul=8

```


1.1 Se dau înălțimea și raza unui cilindru circular drept. Să se calculeze aria laterală, aria totală și volumul cilindrului și ale conului circular drept de aceeași rază și înălțime cu cilindrul. (R: pag. 53)

1.2 Se dau lungimile laturilor unui triunghi. Să se calculeze aria triunghiului și lungimile înălțimilor sale. (R: pag. 54)

1.3 Se dau trei numere reale u, v, w . Să se calculeze valoarea expresiei:

$$E = \begin{cases} u \cdot v & \text{dacă } w < 0 \\ u + v & \text{dacă } w = 0 \\ 0 & \text{dacă } w > 0 \end{cases}$$

(R: pag. 55)

1.4 Să se calculeze valoarea funcției reale de variabilă reală definită astfel:

$$f(x) = \begin{cases} 3x - 5 & \text{dacă } x < 5 \\ 10 & \text{dacă } 5 \leq x \leq 10 \\ 9x + 1 & \text{dacă } x > 10 \end{cases}$$

pentru o valoare dată x a variabilei. (R: pag. 57)

1.5 Se dau patru numere reale a, b, c, d . Să se afișeze numerele în ordinea dată și apoi, dacă $d \geq 0$, în ordinea a, c, b, d iar, în caz contrar, în ordinea a, b, d, c . (R: pag. 58)

1.6 Să se calculeze valoarea expresiei E definită astfel:

$$E = \begin{cases} d - 3b & \text{dacă } a + c > 2d \text{ și } b > 0 \\ d + 3b & \text{dacă } a + c > 2d \text{ și } b < 0 \\ 4 & \text{in rest} \end{cases}$$

a, b, c, d fiind patru numere reale date. (R: pag. 59)

1.7 Aceeași problemă pentru expresia:

$$E = \begin{cases} \max(|a| - 3, |b| + 6) & \text{dacă } c + 2d < 5 \\ \min(|a| - 3, |b| + 6) & \text{dacă } 5 \leq c + 2d \leq 7 \\ \min(|a| + |b|, 18) & \text{dacă } c + 2d > 7 \end{cases}$$

(R: pag. 60)

1.8 Aceeași problemă pentru expresia:

$$E = \begin{cases} \max(\min(|a| - 3b, c - 2|d|), \min(|a| + 3b, c + 2|d|)) & \text{dacă } a + b > c - d \\ \min(\max(|a| - 3b, c - 2|d|), \max(|a| + 3b, c + 2|d|)) & \text{dacă } a + b \leq c - d \end{cases}$$

(R: pag. 61)

1.9 Se dă un pătrat de latură l , căruia i se circumscrie un cerc, acestuia i se circumscrie un pătrat etc. Să se calculeze aria cercului și pătratului obținut după n pași, considerând un pas ca fiind circumscrierea unei figuri.

(R: pag. 63)

1.10 Se dau trei numere reale strict pozitive a, b, c . Să se decidă dacă ele pot reprezenta lungimile laturilor unui triunghi și, în caz afirmativ, să se calculeze raza cercului înscris, respectiv circumscriștriunghiului. (R: pag. 64)

1.11 Se dau trei numere reale strict pozitive a, b, c reprezentând lungimile laturilor unui triunghi. Să se decidă dacă triunghiul este ascuțitunghic, dreptunghic sau obtuzunghic. (R: pag. 66)

2. SUME, PRODUSE, CONTOARE

Dacă se cere calcularea unei sume (respectiv a unui produs), înainte de a efectua calculul cerut, se inițializează suma cu valoarea 0 (respectiv produsul cu valoarea 1), care este elementul neutru față de adunare (respectiv față de înmulțire).

Un *switch* este o variabilă care ia două sau mai multe valori distincte, pentru a marca două sau mai multe situații posibile.

2.1 Se dau trei numere reale. Să se calculeze valoarea absolută a celor trei numere și media aritmetică a modulelor lor. Numerele se citesc și se prelucrează pe rând.

(R: pag.68)

2.2. Se citesc pe rând patru numere naturale. Să se numere câte dintre ele împărțite la 13 dau restul 7. Să se afișeze aceste numere și să se calculeze produsul celorlalte.

(R: pag.69)

2.3 Să se numere câte din patru numere naturale date se divid cu 3, câte cu 5 și câte cu 15.

(R: pag.69)

2.4 Se citesc pe rând trei numere reale. Să se calculeze media geometrică a numerelor strict pozitive și media aritmetică a celor negative, luând în considerare și posibilitatea ca toate numerele citite să fie de același semn.

(R: pag.71)

3. STRUCTURI REPETITIVE DE TIP WHILE-DO ȘI DO-UNTIL

Să se calculeze *cmmdc* (cel mai mare divizor comun) și *cmmmc* (cel mai mic multiplu comun) a două numere naturale date, folosind algoritmul lui Euclid.

INDICAȚIE:

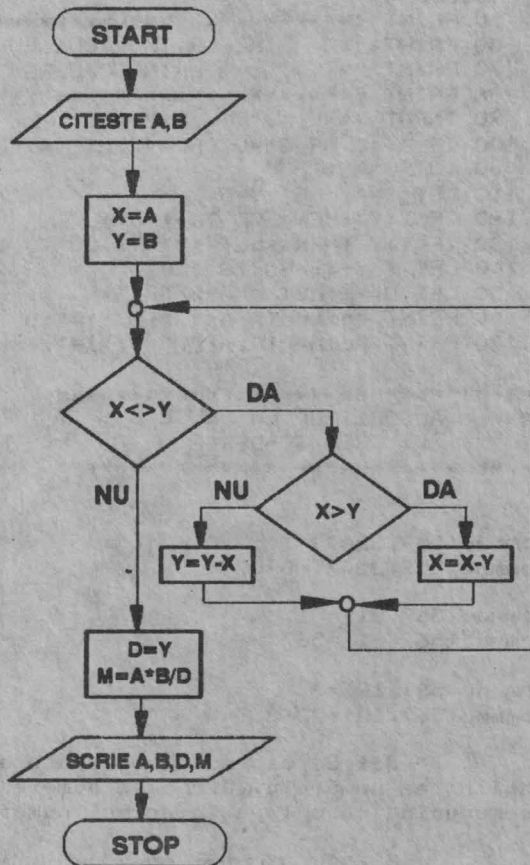
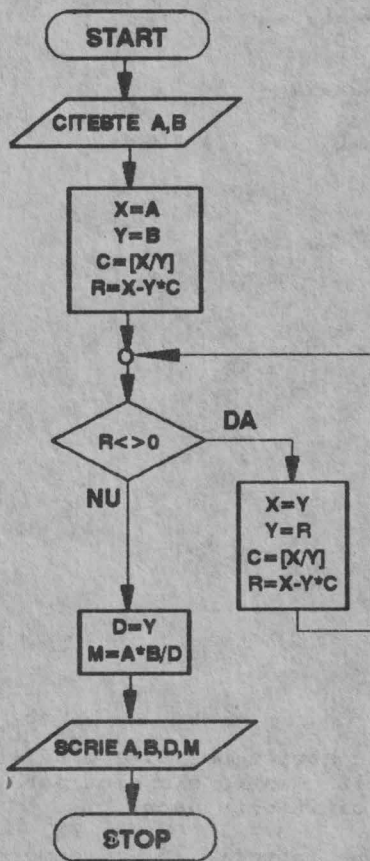
Fie a, b numerele date și d cel mai mare divizor comun al lor. Atunci cel mai mic multiplu comun se calculează cu formula $cmmmc = ab/d$. Valoarea lui d se obține folosind algoritmul lui Euclid.

algoritmul EUCLID 1 este:

```
citește a,b
x ← a
y ← b
c ← [x/y]
r ← x-y·c
cît-timp r≠0 execută
  x ← y
  y ← r
  c ← [x/y]
  r ← x-y·c
sfârșit-cît-timp
d ← y
m ← a·b/y
scrie a,b,d,m
stop
```

algoritmul EUCLID 2 este:

```
citește a,b
x ← a
y ← b
cît-timp x≠y execută
  dacă x>y atunci x ← x-y
  altfel y ← y-x
sfârșit-dacă
sfârșit-cît-timp
d ← y
m ← a·b/d
scrie a,b,d,m
stop
```



```

10 REM
20 REM *** ALGORITMUL LUI EUCLID - IMPARTIRE ***
30 REM
40 CLS
90 INPUT "A=";A;" B=";B
100 IF A<=0 OR B<=0 OR A<>INT (A) OR B<>INT (B) THEN
GO TO 90
110 LET X=A: LET Y=B
120 LET C=INT (X/Y): LET R=X-Y*C
130 IF R=0 THEN GO TO 150
140 LET X=Y: LET Y=R: GO TO 120
150 LET D=Y: LET M=A*B/D
160 PRINT "cmmdc(";A;" ";B;")=";D
170 PRINT "cmmmc(";A;" ";B;")=";M
  
```

```

cmmdc(259,3247)=1
cmmmc(259,3247)=840973
  
```

```

cmmdc(567,81)=81
cmmmc(567,81)=567
  
```

```

cmmdc(567,12)=3
cmmmc(567,12)=2268
  
```

```

40 CLS
50 PRINT "*****"
60 PRINT "    ALGORITMUL LUI EUCLID    *"
70 PRINT "    PRIN SCADERE            *"
80 PRINT "*****"
90 INPUT "A=";A;"    B=";B
100 IF A<=0 OR B<=0 OR A<>INT (A) OR B<>INT (B) THEN
    GO TO 90
110 LET X=A: LET Y=B
120 IF X=Y THEN GO TO 150
130 IF X>Y THEN LET X=X-Y: GO TO 120
140 LET Y=Y-X: GO TO 120
150 LET D=Y: LET M=A*B/D
160 PRINT "cmmdc(";A;" " ;B;" )=";D
170 PRINT "cmmmc(";A;" " ;B;" )=";M

```

```

*****
*    ALGORITMUL LUI EUCLID    *
*    PRIN SCADERE            *
*****

```

```

cmmdc(259,3247)=1
cmmmc(259,3247)=840973

```

```

cmmdc(567,81)=81
cmmmc(567,81)=567

```

```

cmmdc(567,12)=3
cmmmc(567,12)=2268

```

3.1 Se citesc numere naturale strict pozitive pînă la întilnirea numărului 0. Să se numere cite dintre ele sînt pare, presupunind că cel puțin primul număr citit este nenul.

(R: pag. 73)

3.2 Se citesc pe rînd numere întregi. Să se numere cite dintre ele împărțite la 7 dau restul 5 și cite împărțite la 13 dau restul 7. Citirea se oprește la întilnirea primului număr care nu este întreg.

(R: pag. 73)

3.3 Se citesc numere naturale pînă la întilnirea numărului 13. Știind că sînt cel puțin două numere, dintre care primul este sigur diferit de 13, să se afișeze perechile de numere citite consecutiv, care îndeplinesc condiția că al doilea număr se divide la primul.

(R: pag. 74)

3.4 Se citesc numere naturale pînă la întilnirea a două numere consecutive egale. Să se calculeze media aritmetică a numerelor citite, excluzînd ultimul număr (cel care este egal cu penultimul). Se presupune că sînt cel puțin trei numere, dintre care primele două sînt sigur distincte.

(R: pag. 75)

3.5 Aceeași problemă, presupunînd însă că sînt cel puțin două numere, nu obligatoriu distincte.

(R: pag. 76)

3.6 Aceeași problemă, excluzînd însă din media aritmetică ultimele două numere citite (cele care sînt egale între ele) și presupunînd că sînt cel puțin două numere, nu obligatoriu distincte.

(R: pag. 77)

3.7 Se dă un număr natural n , să se afle cite cifre 5 conține pătratul său.

(R: pag. 78)

3.8 Se citesc numere reale nenule pînă la citirea numărului 0. Să se calculeze suma primului cu cel de-al treilea, cu cel de-al cincilea etc. și produsul celui de-al doilea cu al patrulea, al șaselea etc.

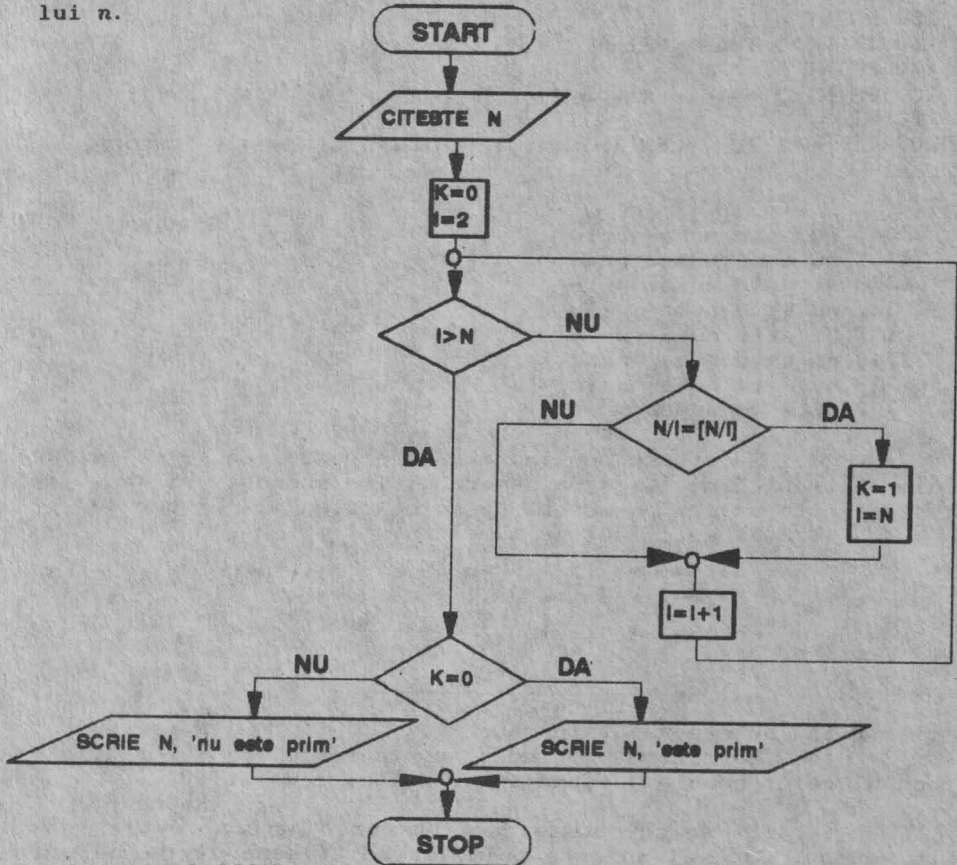
(R: pag. 78)

4. STRUCTURI REPETITIVE DE TIP FOR

Să se stabilească dacă un număr natural dat n este prim sau nu.

INDICAȚIE:

Știind că un număr prim are doi divizori: 1 și el însuși, vom verifica dacă numărul dat este divizibil cu unul din numerele cuprinse în intervalul $[2, \lfloor n/2 \rfloor]$. În momentul în care s-a găsit un divizor, vom ieși forțat din structura repetitivă deoarece este clar că numărul nu este prim. Pentru a ști care este modul în care s-a ieșit din FOR se folosește un switch inițializat cu zero înainte de intrarea în structura repetitivă și căruia i se atribuie valoarea 1 la găsirea unui divizor al lui n .



algoritmul PRIM este:

```

citește n
k ← 0
pentru i=2, [n/2] execută
  dacă n/i=[n/i] atunci k ← 1
  sfârșit-dacă
sfârșit-pentru
scrie n
dacă k=0 atunci scrie 'este prim'
  altfel scrie 'nu este prim'
sfârșit-dacă
stop
  
```

```

10 REM
20 REM *** N ESTE NUMAR PRIM ? ***
30 REM
40 CLS
80 INPUT "Cit este n? ";n
90 IF n<1 OR n<>INT n THEN GO TO 80
100 LET k=0
110 FOR i=2 TO INT (n/2)
120   IF n/i=INT (n/i) THEN LET k=1: LET i=n
130 NEXT i
140 GO SUB 200
150 PRINT n;
160 IF k<>0 THEN PRINT " nu";: GO TO 180
170 PRINT " ";
180 PRINT " este numar prim"
190 STOP
200 FOR u=1 TO 7-LEN (STR$ n): PRINT " ";: NEXT u: RETURN

```

```

2345 nu este numar prim
4321 nu este numar prim
1373   este numar prim
 567 nu este numar prim
 137   este numar prim
32768 nu este numar prim
  33 nu este numar prim,
  2   este numar prim

```

4.1 Se citesc pe rind n numere reale. Să se calculeze media aritmetică a cite două numere citite succesiv și să se afișeze aceste valori pe măsură ce se calculează. (R: pag.80)

4.2 Să se calculeze sumele:

$$a) S=1^2+3^2+5^2+\dots+n^2 \quad (n \text{ impar})$$

$$b) S=1 \cdot 1 \cdot 2 + 1 \cdot 2 \cdot 3 + \dots + 1 \cdot 2 \cdot 3 \cdot \dots \cdot n$$

$$c) S=\frac{1}{1} + \frac{1}{1 \cdot 2} + \dots + \frac{1}{1 \cdot 2 \cdot \dots \cdot n}$$

$$d) S=\frac{1+2+\dots+n}{(n+1)+(n+2)+\dots+(n+l)}$$

n și l fiind numere naturale date.

(R: pag. 80)

4.3 Se citesc pe rind n numere reale. Să se calculeze produsul celor nenule și suma celor mai mari decît 10.

(R: pag. 85)

4.4 Să se cerceteze dacă un număr natural dat n este prim sau nu. Dacă el nu este prim, să se afișeze divizorii săi și numărul acestora.

(R: pag. 85)

4.5 Să se scrie numerele prime mai mici decît 100 și numărul acestora.

(R: pag. 87)

4.6 Se citesc pe rind n numere reale. Să se calculeze raportul dintre suma algebrică a primului număr cu al treilea, al cincilea etc. și suma algebrică a celui de-al doilea cu al patrulea, al șaselea etc.

(R: pag. 87)

4.7 Să se calculeze valoarea expresiei:

$$E = \begin{cases} \frac{(n+1)(n+2)\dots(n+m)}{n-m+1} & \text{dacă } n > m+1 \\ n & \text{dacă } n = m+1 \\ \frac{(m+1)(m+2)\dots(m+n)}{m-n+1} & \text{dacă } n < m+1 \end{cases} \quad \begin{matrix} (n \text{ și } m \\ \text{numere} \\ \text{naturale} \\ \text{date)} \\ (R: \text{ pag. } 90) \end{matrix}$$

4.8 Să se calculeze, pentru un număr natural dat n , suma $s_n = u_1 + u_2 + \dots + u_n$, fără a calcula pe rînd valorile u_1, u_2, \dots, u_n , unde $u_i = \frac{1}{1 \cdot 2} + \frac{1 \cdot 2}{1 \cdot 2 \cdot 3} + \dots + \frac{1 \cdot 2 \cdot \dots \cdot i}{1 \cdot 2 \cdot \dots \cdot (i+1)}$ pentru orice $i \in \{1, 2, \dots, n\}$. (R: pag. 91)

4.9 Se dă un număr real x și un număr natural n . Să se calculeze suma $s_n = u_1 + u_2 + \dots + u_n$, unde:

$$u_i = 1 - \frac{1}{2} \cdot x + \frac{1 \cdot 3}{2 \cdot 4} \cdot x^2 + \dots + (-1)^i \cdot \frac{1 \cdot 3 \cdot \dots \cdot (2i-1)}{2 \cdot 4 \cdot \dots \cdot 2i} \cdot x^i$$

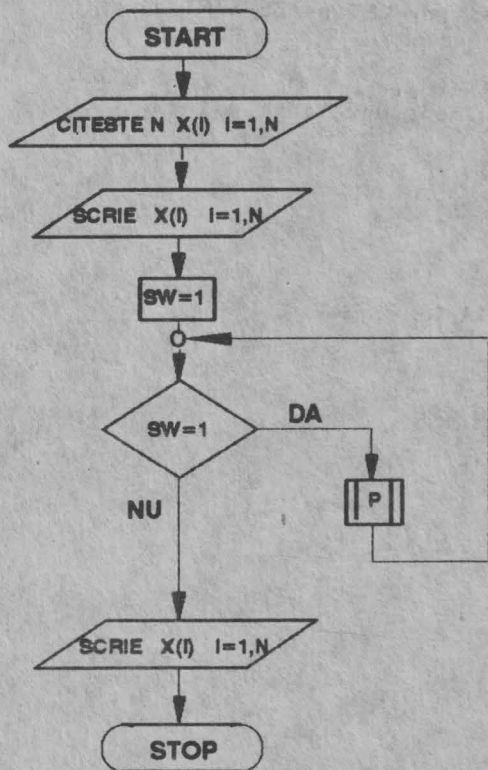
pentru orice $i \in \{1, 2, \dots, n\}$, fără a calcula pe rînd termenii u_1, u_2, \dots, u_n . (R: pag. 92)

5. ȘIRURI

Să se ordoneze crescător elementele unui șir dat folosind metoda de ordonare prin interschimbare.

INDICAȚIE:

Se inițializează un switch cu valoarea 0 apoi se parcurge șirul, comparînd elementele sale consecutive două câte două. Dacă nu sînt în ordine crescătoare, ele se inversează și se modifică valoarea switch-ului. Atîta vreme cît s-a făcut cel puțin o inversare (lucru semnalat de switch) nu putem fi siguri că șirul este ordonat, deci switch-ul se reinițializează cu 0 și se reia parcurgerea șirului. Algoritmul se încheie în cazul în care switch-ul păstrează valoarea 0 după parcurgerea șirului care în acest caz este sigur ordonat.



algoritmul ORDONARE este:

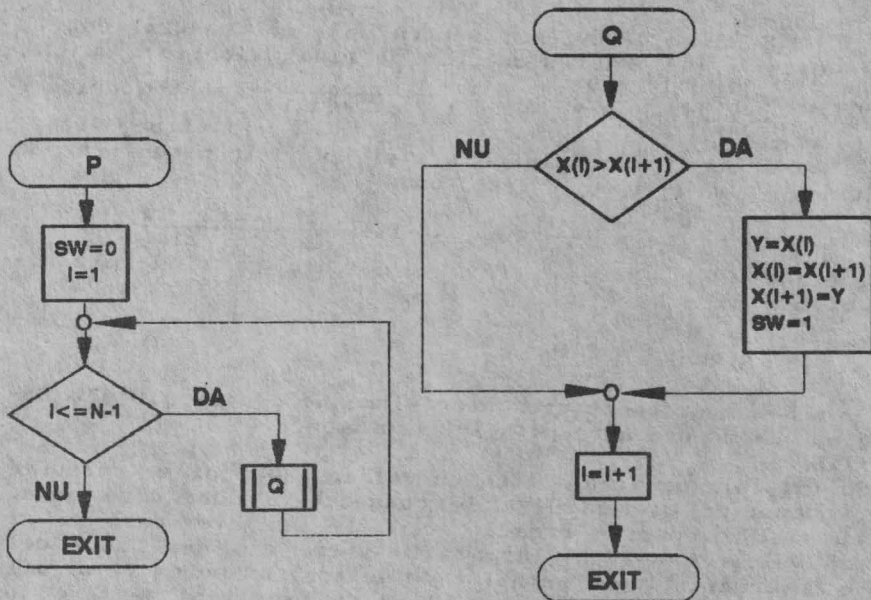
citește n
 pentru $i=1, n$ execută
 citește x_i
 sfîrșit-pentru
 pentru $i=1, n$ execută
 scrie x_i
 sfîrșit-pentru
 $sw \leftarrow 1$
 cît-tîmp $sw=1$ execută
 cheamă P
 sfîrșit-cît-tîmp
 pentru $i=1, n$ execută
 scrie x_i
 sfîrșit-pentru
 stop

procedura P este:

$sw \leftarrow 0$
 pentru $i=1, n-1$ execută
 cheamă Q
 sfîrșit-pentru
 sfîrșit-procedură

procedura Q este:

dacă $x_i > x_{i+1}$ atunci $y \leftarrow x_i$
 $x_i \leftarrow x_{i+1}$
 $x_{i+1} \leftarrow y$
 $sw \leftarrow 1$
 sfîrșit-dacă
 sfîrșit-procedură



```

40 CLS
50 PRINT "*****"
60 PRINT " * ORDONARE PRIN INTERSCHIMBARE *"
70 PRINT "*****"
80 INPUT "dimensiunea sirului n= ";n: LET n=ABS (INT n)
90
100 DIM x(n): LET sw=1
110
120 PRINT ""sirul initial x de dimensiune ";n;" este:"
130 PRINT
140
150 FOR i=1 TO n
160 INPUT "x(";i);"=";x(i)
170 PRINT x(i);" ";
180 NEXT i
190
200 PRINT ""LA O TRECERE PRIN SIR SE OBTINE""
210 IF sw=0 THEN GO TO 330
220 LET sw=0
230 FOR i=1 TO n-1
240 IF x(i)<=x(i+1) THEN GO TO 290
250 LET sw=1
260 LET y=x(i)
270 LET x(i)=x(i+1)
280 LET x(i+1)=y
290 NEXT i
300 GO SUB 360
310 GO TO 210
320
330 PRINT ""sirul ordonat este:"""
340 GO SUB 360
350 STOP
360 FOR i=1 TO n
370 PRINT x(i);" ";
380 NEXT i
390 PRINT : RETURN
  
```


sirul initial x de dimensiune 10 este:

89 -6 23 11 6 -89 0 13 34 -76

LA O TRECERE PRIN SIR SE OBTINE

-6 23 11 6 -89 0 13 34 -76 89
-6 11 6 -89 0 13 23 -76 34 89
-6 6 -89 0 11 13 -76 23 34 89
-6 -89 0 6 11 -76 13 23 34 89
-89 -6 0 6 -76 11 13 23 34 89
-89 -6 0 -76 6 11 13 23 34 89
-89 -6 -76 0 6 11 13 23 34 89
-89 -76 -6 0 6 11 13 23 34 89
-89 -76 -6 0 6 11 13 23 34 89

sirul ordonat este:

-89 -76 -6 0 6 11 13 23 34 89

5.1 Se dă un șir X de n numere reale. Să se formeze un nou șir Y avind elementele definite prin formula $y_i = \text{sgn}(x_i)$.

(R: pag.94)

5.2 Să se inverseze elementele egal depărtate de extremitățile unui șir dat de numere reale.

(R: pag.96)

5.3 Se dă un șir X cu n elemente numere întregi. Să se formeze șirul Y definit prin formula:

$$y_i = \begin{cases} \lfloor x_i/2 \rfloor & \text{dacă } x_i \text{ este impar} \\ x_i/6 & \text{dacă } x_i \text{ se divide la } 6 \\ x_i & \text{in rest} \end{cases} \quad (\text{R: pag.97})$$

5.4 Să se calculeze maximul, minimul și suma elementelor unui șir dat.

(R: pag.97)

5.5 Se dă un șir cu n elemente. Să se înlocuiască fiecare element al său cu media aritmetică a celorlalte $n-1$ elemente ale șirului.

(R: pag.99)

5.6 Să se determine maximul elementelor negative ale unui șir dat de numere reale.

(R: pag.100)

5.7 Se dă un șir de numere reale. Să se calculeze media aritmetică a elementelor sale cuprinse între două numere date a , b și să se formeze un nou șir cu celelalte elemente ale șirului inițial.

(R: pag.103)

5.8 Să se insereze între oricare două elemente alăturate ale unui șir dat X media lor aritmetică.

(R: pag.105)

5.9 Să se calculeze media geometrică a termenilor unui șir dat de numere reale care sînt cuprinși între două numere date. Să se numere și să se însumeze termenii pozitivi; să se numere și să se înmulțească termenii strict negativi.

(R: pag.106)

5.10 Să se ordoneze crescător elementele unui șir dat folosind algoritmul de sortare prin selecție.

(R: pag.107)

5.11 Se dă un șir de numere reale X de dimensiune n ; să se afișeze șirurile cu $n+1$ elemente care se pot forma inserind în toate modurile posibile în șirul dat elementul $y = \max(x_1, x_2, \dots, x_n)$, fără a forma un nou șir.

(R: pag.109)

5.12 Se dă un șir de numere reale. Să se mute la sfîrșitul șirului eventualele elemente nule: a) formînd un nou șir; b) fără a forma un nou șir.

(R: pag.110)

5.13 Să se elimine eventualele elemente nule ale unui șir dat și să se intercaleze între oricare două elemente consecutive de același semn ale șirului obținut produsul lor.

(R: pag.112)

5.14 Aceeași problemă, însă cu cerința ca șirul inițial să fie parcurs o singură dată și să nu se formeze un nou șir pentru eliminarea elementelor nule.

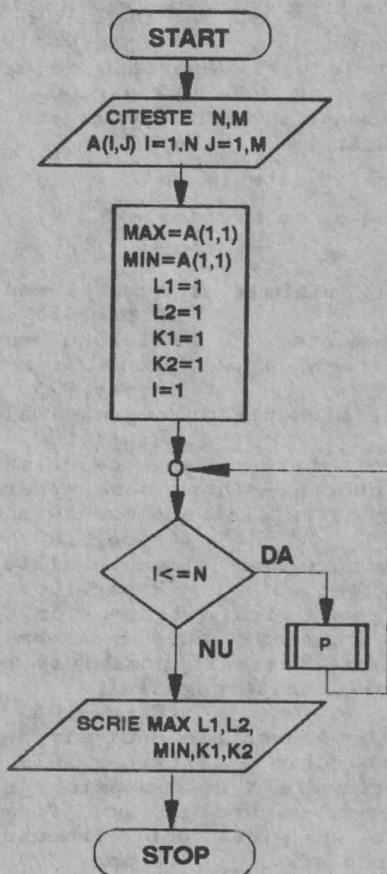
(R: pag.113)

6. MATRICE

Se dă o matrice A . Să se determine cel mai mare, respectiv cel mai mic element al său, să se afișeze aceste elemente și indicii lor.

INDICAȚIE:

Înainte de a începe parcurgerea matricei se inițializează maximul și minimul cu a_{11} , doi indici pentru maxim și doi pentru minim cu valoarea 1. Se parcurge apoi matricea pe linie sau pe coloană, se compară fiecare element cu maximul și cu minimul. Dacă este cazul, se modifică valoarea maximului, respectiv minimului și a indicilor corespunzători.



algoritmul **MINMAXMAT** este:

```

citește n,m
pentru i=1,n execută
  pentru j=1,m execută
    citește aij
  sfîrșit-pentru
sfîrșit-pentru
max ← a11
min ← a11
l1 ← 1
l2 ← 1
k1 ← 1
k2 ← 1
pentru i=1,n execută
  cheamă P
sfîrșit-pentru
scrie max,l1,l2,min,k1,k2
stop
  
```

procedura **P** este:

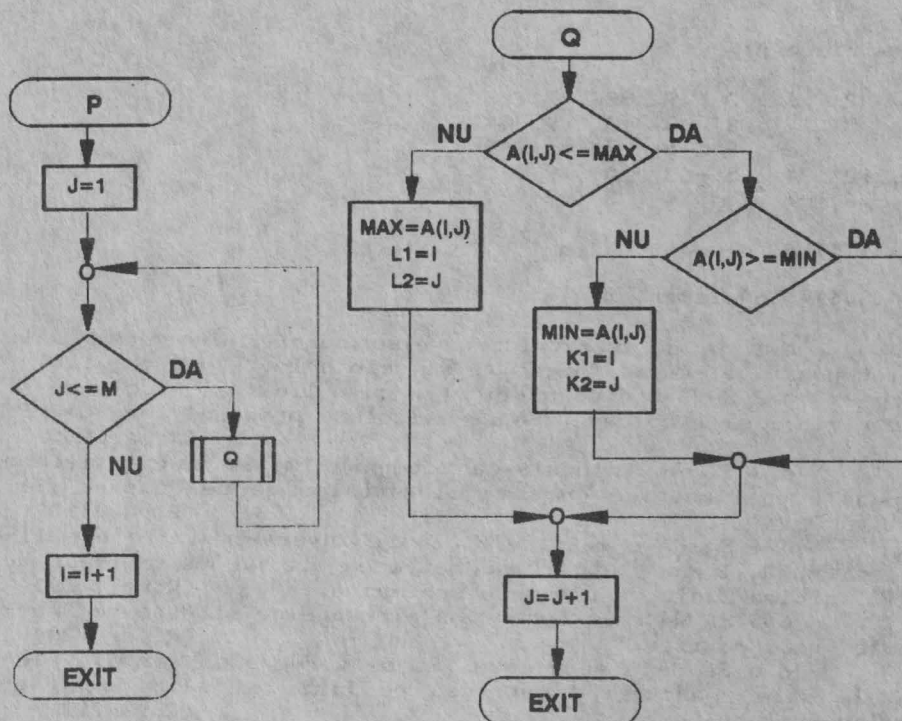
```

pentru j=1,m execută
  cheamă Q
sfîrșit-pentru
sfîrșit-procedură
  
```

procedura **Q** este:

```

dacă aij ≤ max
  atunci
    dacă aij < min
      atunci min ← aij
      k1 ← i
      k2 ← j
    sfîrșit-dacă
  altfel max ← aij
  l1 ← j
  l2 ← j
sfîrșit-dacă
sfîrșit-procedură
  
```



```

80 INPUT "linii n=";n;" coloane=";m: LET n=ABS (INT n):
LET m=ABS (INT m)
90 IF n=0 OR m=0 THEN GO TO 80
100 DIM a(n,m): LET len=4: REM len=lungimea pe care se sc
rie un element din matrice
110 PRINT "matricea A:"
120 FOR i=1 TO n
130 FOR j=1 TO m
140 INPUT "a(";(i);",";(j);")=";a(i,j)
150 GO SUB 320: PRINT a(i,j);
160 NEXT j
170 PRINT
180 NEXT i
190
200 LET max=a(1,1): LET l1=1: LET l2=1
210 LET min=a(1,1): LET k1=1: LET k2=1
220 FOR i=1 TO n
230 FOR j=1 TO m
240 IF a(i,j)>max THEN LET max=a(i,j): LET l1=i: LET
l2=j
250 IF a(i,j)<min THEN LET min=a(i,j): LET k1=i: LET
k2=j
260 NEXT j
270 NEXT i
280
290 PRINT "A(";l1;",";l2;")=";max;" element maxim"
300 PRINT "A(";k1;",";k2;")=";min;" element minim"
310 STOP
320 FOR u=1 TO len-LEN (STR$ a(i,j))
330 PRINT " ";
340 NEXT u: RETURN
  
```

matricea A:

2	6	0	9	88	-7	6
5	12	3	-8	0	9	44
3	2	4	5	7	-86	6
12	34	5	-65	0	9	8
-76	-7	5	4	3	2	0

A(1,5)=88 element maxim

A(3,6)=-86 element minim

6.1 Se dă o matrice; să se numere elementele sale strict pozitive, strict negative și cele nule. (R: pag. 114)

6.2 Se dau două matrice: A cu n linii și m coloane și B cu m linii și k coloane. Să se calculeze produsul lor: C=AxB. (R: pag. 116)

6.3 Să se înlocuiască elementul maxim de pe fiecare coloană a unei matrice cu suma elementelor de pe coloana respectivă. (R: pag. 118)

6.4 Se dă o matrice A. Să se inverseze între ele liniile 1 cu 2, 3 cu 4 etc. Dacă matricea are un număr impar de linii, ultima linie va rămâne neschimbată. (R: pag. 119)

6.5 Să se calculeze suma elementelor situate pe "marginile" unei matrice. (R: pag. 120)

6.6 Se dă o matrice A. Să se formeze un șir din elementele sale, parcurgând matricea pe linii și alt șir, parcurgând-o pe coloane. (R: pag. 121)

6.7 Fie A o matrice cu elementele numere reale. Să se formeze un șir X definit în felul următor: x_1 este maximul elementelor de pe coloanele 1 și 2, x_2 este maximul elementelor de pe coloanele 3 și 4 etc. Dacă matricea are un număr impar de coloane atunci ultimul element al șirului este maximul elementelor de pe ultima coloană a matricei. (R: pag. 122)

6.8 Se dă o matrice A. Să se introducă indicii elementelor subunitare pozitive în liniile unei matrice B cu două coloane. De asemenea să se formeze un șir X ce conține aceste elemente. (R: pag. 124)

6.9 Să se rearanjeze liniile unei matrice A astfel încât elementele de pe ultima sa coloană să fie în ordine crescătoare:

a) fără a forma o nouă matrice;

b) formind o nouă matrice. (R: pag. 126)

6.10 Se dau patru matrice: A cu m linii și n coloane, B cu m linii și l coloane, C cu k linii și n coloane și D cu k linii și l coloane, m, n, k, l fiind patru numere naturale date. Să se formeze matricea:

$$T = \begin{pmatrix} A & B \\ C & D \end{pmatrix} \quad (\text{R: pag. 130})$$

6.11 Se dau patru matrice: A cu 2n linii și m coloane, B cu n linii și k coloane, C cu 2n linii și o coloană și D cu m linii și k+1 coloane. Să se formeze matricea:

$$T = \begin{pmatrix} A & B & C \\ I & O & D \end{pmatrix}$$

unde O este matricea cu toate elementele nule cu n linii și k coloane iar I este matricea unitate cu m linii și m coloane.

(R: pag. 132)

6.12 Se dă o matrice pătratică A . Să se înlocuiască elementele sale situate pe diagonala principală și pe diagonala secundară cu valoarea 0. (R: pag.134)

6.13 Se dă o matrice pătratică A de dimensiune n . Să se formeze o matrice simetrică B de aceeași dimensiune, ale cărei elemente sînt definite prin relația:

$$b_{ij} = (a_{ij} + a_{ji}) / 2$$

efectuînd un număr minim de pași. (R: pag.135)

6.14 Fie T o matrice pătratică de dimensiune n și T' transpusa ei. Să se formeze matricea:

$$A = \begin{pmatrix} T & T' \\ T' & T \end{pmatrix}$$

cu un număr minim de pași. (R: pag.136)

6.15 Să se formeze un șir din elementele situate pe prima paralelă deasupra și prima paralelă dedesubtul diagonalei principale a unei matrice pătratice date:

a) în această ordine (adică mai întii cele de pe prima paralelă deasupra diagonalei principale urmate de cele de pe prima paralelă de dedesubtul diagonalei principale);

b) în orice ordine. (R: pag.137)

6.16 Se dă o matrice pătratică antisimetrică. Să se depună elementele sale într-un șir și să se formeze din nou matricea pornind de la șirul astfel format. (R: pag.139)

6.17 Se dă o matrice simetrică de dimensiune n . Să se depună elementele sale într-un șir și apoi, din acest șir să se formeze din nou matricea inițială. (R: pag.142)

6.18 Să se formeze un șir cu elementele strict pozitive situate deasupra diagonalei secundare a unei matrice pătratice, parcurgînd matricea pe linii. (R: pag.142)

6.19 Se dă o matrice pătratică A de dimensiune n cu elemente numere reale. Să se formeze șirul X cu elementele definite astfel:

$$\begin{cases} 1 & \text{dacă } \sum_{j=1}^i a_{ij} > \sum_{j=1}^i a_{ji} \\ -1 & \text{în caz contrar.} \end{cases} \quad (\text{R: pag.143})$$

7. ȘIRURI DE CARACTERE

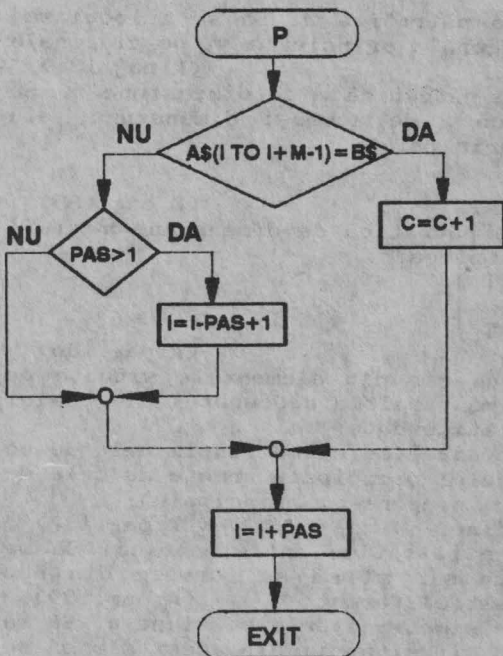
Să se scrie un program care numără de cîte ori un șir este cuprins în alt șir (adică toate caracterele sale se regăsesc în aceeași ordine în acest șir).

INDICAȚIE:

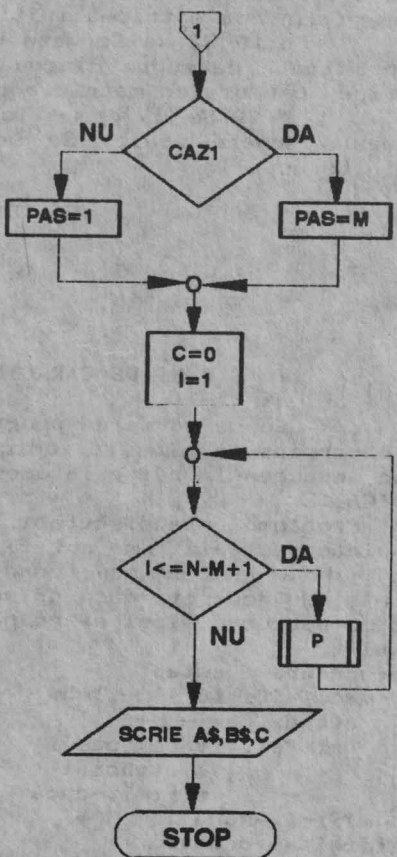
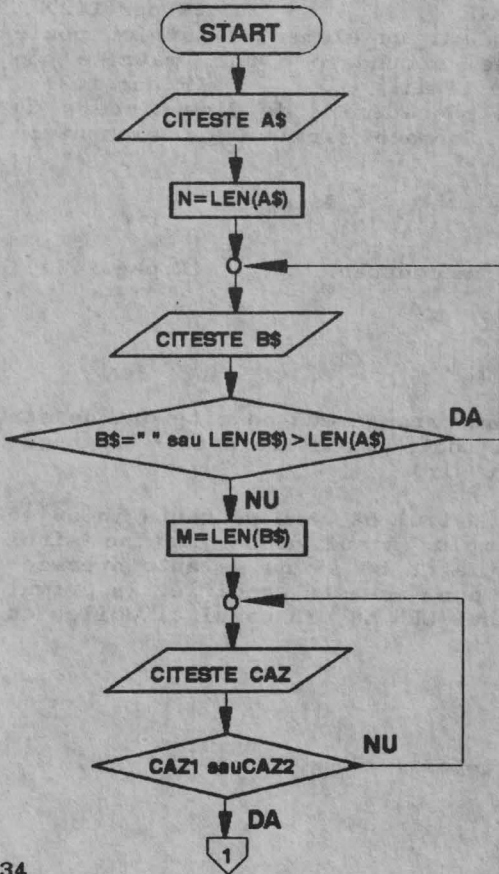
Problema este discutabilă: șirul $b\$\$$ care se caută în $a\$\$$ se autointersectează sau nu! Exemplu: șirul $ababa$ conține șirul aba o dată dacă se pune condiția ca $b\$\$$ să nu se autointersecteze și de două ori dacă nu se pune această condiție. În primul caz se parcurge șirul $a\$\$$ cu pasul $LEN\ b\$\$,$ în cazul al doilea cu pasul 1.

```

procedura P este:
  dacă a$(i to i+m-1)=b$
    atunci c ← c+1
    altfel [ dacă pas>1
            [ atunci i ← i-pas+1
            [ sfîrșit-dacă
  sfîrșit-dacă
sfîrșit-procedură
  
```



algoritmul ȘIRURI este:
 citește a\$
 $n \leftarrow \text{LEN}(a\$)$
 execută
 citește b\$
 ieșim-cînd (b\$='') or
 ($\text{LEN}(b\$) \leq \text{LEN}(a\$)$)
 $m \leftarrow \text{LEN}(b\$)$
 execută
 citește caz
 ieșim-cînd (caz=1) or
 (caz=2)
 dacă caz=1
 atunci pas $\leftarrow m$
 altfel pas $\leftarrow 1$
 sfîrșit-dacă
 $c \leftarrow 0$
 pentru $i=1, n-m+1, \text{pas}$ execută
 cheamă P
 sfîrșit-pentru
 scrie a\$, b\$, c
 stop



```

10 REM
20 REM *** CONTORIZARE APARITII SUBSIR ***
30 REM
40 PAPER 0: INK 7: BORDER 0: BRIGHT 1: CLS
50 PRINT "*****"
60 PRINT "* CONTORIZARE APARITII SUBSIR *"
70 PRINT "*****"
80 LET p$=" Sa se scrie un program care sa numere de cit
e ori un sir este cuprins in alt sir (adica toate caracterel
e sale se regasesc in aceasi ordine in sirul initial"
90 FOR i=1 TO LEN p$: PRINT p$(i);: BEEP 0.005,15: NEXT i
100 PRINT '/: FOR i=1 TO 32: PRINT "-": NEXT i: PRINT '/
110 PRINT AT 21,0;"Sirul initial:"
120 INPUT LINE a$
130 LET n=LEN a$: IF n=0 THEN GO TO 110
140 PRINT AT 21,0;"Sirul care se cauta:"
150 INPUT LINE b$
160 LET m=LEN b$: IF m=0 OR m>n THEN GO TO 140
170 DIM s$(320): PRINT AT 12,0;s$
180 INPUT "autointersectare-2 sau nu-1:";caz
190 LET caz=ABS (INT caz)
200 IF caz<>1 AND caz<>2 THEN GO TO 180
210 IF caz=2 THEN LET pas=1: GO TO 230
220 LET pas=m
230 LET c=0
240 FOR i=1 TO n-m+1 STEP pas
250 IF a$(i TO i+m-1)=b$ THEN LET c=c+1: GO TO 270
260 IF pas>1 THEN LET i=i-pas+1
270 NEXT i
280 PRINT AT 12,0;
290 PRINT "Sirul:"; PRINT 'a$
300 PRINT "contine sirul:"; PRINT 'b$
310 PRINT "de ";c;" ori"

```

```

*****
* CONTORIZARE APARITII SUBSIR *
*****

```

Sirul:

ababababababab

contine sirul:

aba

de 6 ori

Sirul:

ababababababab

contine sirul:

aba

de 3 ori

7.1 Se dă o variabilă de tip șir. Să se scrie un program care să distribuie în mod aleator elementele acestui șir și el însuși. (R: pag.144)

7.2 Se consideră un mesaj memorat într-o variabilă șir. Să se scrie un program care să "rotească" pe ecran acest mesaj (caracterele care "ies" prin stînga să "intre" prin dreapta), pînă la apăsarea unei anumite taste. (R: pag.146)

7.3 Să se scrie un program care, folosind proprietățile și instrucțiunile corespunzătoare lucrului cu șiruri de caractere, să permită introducerea de la tastatură a unui număr întreg format din maximum n caractere, interzicînd tastarea altor caractere. (R: pag.146)

7.4 Se cunoaște că în microinformatica actuală folosirea meniurilor de tip WINDOW (fereastră) este din ce în ce mai utilizată. Să se scrie un program care să pună pe ecran o fereastră de dimensiuni dorite, în locul dorit, care să conțină și atributele dorite (INK, PAPER, BRIGHT). De asemenea fereastra să aibă trasate marginile. (R: pag.148)

7.5 Să se scrie un program care memorează n nume de persoane și ocupațiile lor și care poate afișa, după dorință, lista persoanelor și ocupațiile lor, lista persoanelor care au o anumită ocupație și lista numerelor de ordine disponibile la un moment dat. (R: pag.150)

7.6 Se consideră un text în limba română, corect, memorat într-o variabilă șir. Să se scrie un program care :

- determină frecvența de apariție a literelor din text;
- numără cuvintele din text (cuvintele sînt separate prin spațiu, punct, virgulă, :, ;, paranteze, !, ?, /);
- să numere cite fraze sînt în text (o frază se încheie cu punct, !, ?).

(R: pag.155)

7.7 Să se scrie un program care:

- memorează n nume de elevi, mediile la matematică, fizică și limba română, toate presupuse corecte, sub forma unui tablou bidimensional cu elemente de tip șir de caractere și calculează media generală;
- sortează tabloul, la cerere, în ordine alfabetică, după cele trei medii sau după media generală;
- afișează tabloul sortat la b);
- modifică datele unui elev.

Observație:

-programul să conțină o singură rutină de sortare.

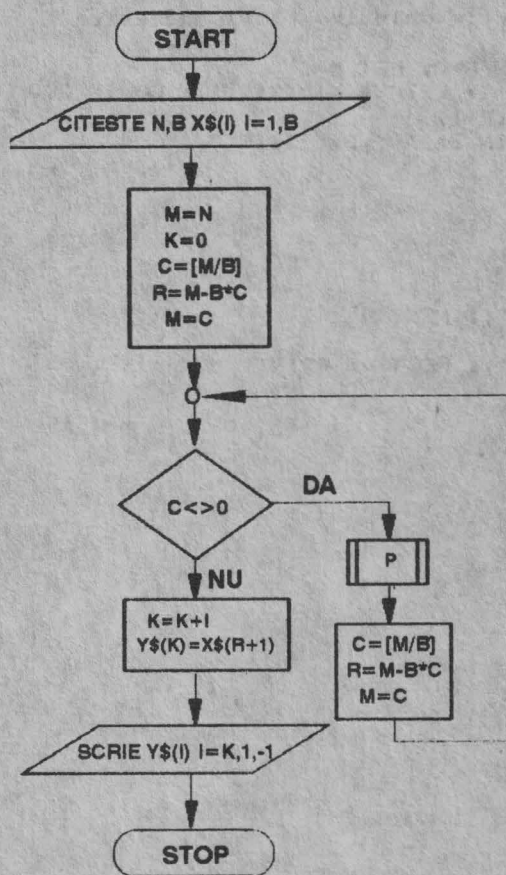
(R: pag.157)

8. PROBLEME DIVERSE

Să se treacă un număr natural dat n din baza 10 într-o bază oarecare b .

INDICAȚIE:

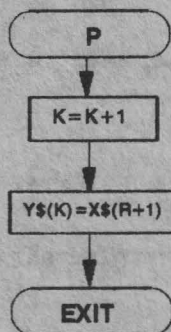
Cifrele bazei b se dau sub forma unui șir de caractere de lungime b (deoarece, pentru baze mai mari decît 10, cifrele zecimale nu sînt suficiente). Trecerea numărului în baza b se face prin împărțiri repetate cu b . Dacă restul împărțirii la b este i , $0 < i < (b-1)$, cifra corespunzătoare va fi elementul de indice $i+1$ al șirului de cifre (deoarece primul element este 0, al doilea este 1 etc.)



algoritmul CONVERSIE este:

```

citește n, b
pentru i=1, b execută
  citește x$(i)
sfârșit-pentru
m ← n
k ← 0
c ← [m/b]
r ← m-b·c
m ← c
cît-timp c≠0 execută
  cheamă P
  c ← [m/b]
  r ← m-b·c
  m ← c
sfârșit-cît-timp
k ← k+1
y$(k) ← x$(r+1)
pentru j=k, 1, -1 execută
  scrie y$(j)
sfârșit-pentru
stop
  
```



procedura P este:

```

k ← k+1
y$(k) ← x$(r+1)
  
```

sfârșit-procedură

```

10 CLS : LET z1=4: LET z2=24: GO SUB 300
20 PRINT " * TRECERA DIN BAZA 10 *"
30 PRINT " * INTR-O BAZA DATA B *": GO SUB 300
40 INPUT "Introduceti baza B ";B: DIM X$(B)
50 PRINT "' ' BAZA=";B
60 PRINT "' ' CIFRELE BAZEI ";B;" SINT:"
70 PRINT "' ' ";
80 PAUSE 50
90 FOR i=1 TO b
100 INPUT x$(i): PRINT x$(i); " ";: IF i=14 THEN PRINT : PRINT
" ";
110 PRINT " ";
120 NEXT i
130 PRINT : INPUT "Introduceti numarul de convertit ";N
140 DIM Y$(20): LET M=N
150 LET K=0
160 PRINT "' ' NUMARUL ESTE:";N
170
  
```

```

180 REM Se realizeaza conversia prin impartiri succesive
190
200 LET C=INT (M/B): LET R=M-B*C: LET M=C
210 IF C<>0 THEN LET K=K+1: LET Y$(K)=X$(R+1): GO TO 200
220 LET K=K+1: LET Y$(K)=X$(R+1)
230 PRINT "" CONVERTIT IN BAZA ";B;" ESTE:"
240 PRINT "" ""
250 FOR J=K TO 1 STEP -1
260 PRINT Y$(J);
270 NEXT J
280 PAUSE 0: CLS
290 STOP
300 FOR v=1 TO z1: PRINT " ";: NEXT v
310
320 FOR v=1 TO z2: PRINT "*";: NEXT v: PRINT
330 RETURN

```

```

*****
* TRECEREA DIN BAZA 10 *
* INTR-O BAZA DATA B *
*****

```

BAZA=2

CIFRELE BAZEI 2 SINT:

0 1

NUMARUL ESTE:167

CONVERTIT IN BAZA 2 ESTE:

10100111

8.1.1 Să se afișeze toate numerele binare cuprinse între 0 și 2^{n-1} , n fiind un număr natural dat, folosind adunarea cu 1 în baza 2. (R: pag.164)

8.1.2 Aceeași problemă folosind însă adunarea în zecimal. (R: pag.165)

8.1.3 Se dă un număr natural reprezentat într-o bază b sub forma unui șir de cifre. Să se mărească acest număr cu 1. (R: pag.166)

8.1.4 Să se efectueze suma a două numere naturale reprezentate într-o bază dată b sub formă de șiruri de cifre de lungime dată n , completate eventual cu zerouri nesemnificative. (R: pag.167)

8.1.5 Să se efectueze diferența a două numere naturale date, reprezentate ca în problema precedentă. (R: pag.169)

Să se descompună un număr natural dat n în factori primi.

INDICAȚIE:

Se va forma o matrice cu 2 coloane ce va conține în prima coloană factorii primi iar în cea de a doua, exponenții respectivi. Indicele de linie se inițializează cu zero și se mărește cu 1 pentru fiecare factor prim găsit. Dacă numărul dat este prim, indicele de linie păstrează valoarea zero.

algoritmul FACTORI este:

citește n

$m \leftarrow n$

$k \leftarrow 0$

$i \leftarrow 2$

$l \leftarrow \lfloor n/2 \rfloor$

cît-timp $i \leq l$ execută

cheamă P

sfîrșit-cît-timp

dacă $k=0$

atunci scrie n, 'este prim'

altfel [pentru $i=1, k$

execută

scrie $a_{i,1}, a_{i,2}$

sfîrșit-pentru

sfîrșit-dacă

stop

procedura P este:

dacă $m/i = \lfloor m/i \rfloor$

atunci $k \leftarrow k+1$

$a_{k,1} \leftarrow i$

$j \leftarrow 1$

cheamă Q

sfîrșit-dacă

$i \leftarrow i+1$

sfîrșit-procedură

procedura Q este:

$m \leftarrow m/i$

cît-timp $m/i = \lfloor m/i \rfloor$ execută

$j \leftarrow j+1$

$m \leftarrow m/i$

sfîrșit-cît-timp

$a_{k,2} \leftarrow j$

$l \leftarrow \lfloor m/2 \rfloor$

sfîrșit-procedură

```
10 LET Z1=4: LET Z2=22: GO SUB 300
20 PRINT " * DESCOMPUNEREA UNUI *"
30 PRINT " * NUMAR NATURAL DAT *"
40 PRINT " * IN FACTORI PRIMI *": GO SUB 300
50 DIM A(10,2)
60 INPUT "Introduceti nr. N=": N
70 LET M=0: LET I=2: LET L=INT (N/2)
80 LET N1=N
90 IF I>N THEN PRINT "" "": GO TO 110
100 GO SUB 220: GO TO 90
110 IF M=1 AND A(1,2)=1 THEN PRINT N1;" ESTE PRIM": GO TO 60
120 PRINT "" " DESCOMPUNEREA IN FACTORI "
130 PRINT " PRIMI A NUMARULUI ";N1;" ESTE:"""
140 FOR I=1 TO M
150 FOR V=1 TO 6-LEN STR$(A(I,1))
160 PRINT " ";
170 NEXT V
180 PRINT A(I,1);"";A(I,2)
190 NEXT I
200 GO TO 60
220 IF M<>I*INT (N/I) THEN GO TO 270
230 LET M=M+1: LET A(M,1)=I: LET K=1
240 LET N=N/I
250 IF N=I*INT (N/I) THEN LET K=K+1: GO TO 240
260 LET A(M,2)=K
270 LET I=I+1
280 RETURN
290
300 FOR V=1 TO Z1: PRINT " "": NEXT V
310 FOR V=1 TO Z2: PRINT " *": NEXT V
320 PRINT : RETURN
```

```
*****
* DESCOMPUNEREA UNUI *
* NUMAR NATURAL DAT *
* IN FACTORI PRIMI *
*****
```

DESCOMPUNEREA IN FACTORI
PRIMI A NUMARULUI 84 ESTE:

2^2
 3^1
 7^1

DESCOMPUNEREA IN FACTORI
PRIMI A NUMARULUI 210 ESTE:

2^1
 3^1
 5^1
 7^1

19 ESTE PRIM

8.2.1 Să se calculeze radicalul unui număr real a pozitiv dat, folosind formula de recurență:

$$x_n = \frac{a}{x_{n-1}} + x_{n-1} \quad (R: \text{pag.171})$$

$x_1 = 1.$

8.2.2 Să se determine cel mai mic număr de 5 cifre care începe și se termină cu aceeași cifră și pentru care restul la împărțirea cu p este r . (R: pag.172)

8.2.3 Se dă un număr natural n . Să se determine cel mai mic număr posibil de obținut din numărul dat, eliminând una dintre cifrele sale, fără a genera toate numerele ce se pot obține prin eliminarea unei cifre. (R: pag.173)

8.2.4 Să se găsească toate numerele n de patru cifre cu proprietatea că există trei cifre a, b, c astfel încit:

$$n = abc + bca + cab \quad (R: \text{pag.174})$$

8.2.5 Se dă un număr natural k . Să se găsească toate posibilitățile de a scrie acest număr ca sumă de numere întregi consecutive. (R: pag.175)

8.2.6 Fiind date două numere naturale m și n , să se verifice următoarea relație:

$$\text{cmmdc}(n+m, \text{cmmmc}(n, m)) = \text{cmmdc}(n, m). \quad (R: \text{pag.176})$$

8.2.7 Să se verifice pentru un număr natural n dat următoarea proprietate: numărul perechilor de numere naturale consecutive, mai mici decit n , al căror produs se divide cu n este egal cu 2^{k-1} , k fiind numărul factorilor primi ai lui n . (R: pag.178)

8.2.8 Se consideră un șir format din numere naturale mai mari decit 1. Să se formeze un nou șir cu eventualele elemente ale șirului dat care sînt întregi liberi de pătrate. (R: pag.180)

8.2.9 Aceeași problemă, cu diferența ca noul șir să conțină eventualele elemente ale șirului inițial ce se pot scrie ca produs de numere prime distincte. (R: pag.181)

8.2.10 Să se găsească toate numerele naturale mai mici decit un număr natural m dat care sînt palindroame (adică citite de la stînga la dreapta sau de la dreapta la stînga reprezintă același număr) și toate numerele naturale k mai mici decit m care sînt perfecte (adică suma divizorilor pozitivi mai mici strict decit k să fie egală cu k). (R: pag.183)

8.3.1. Să se rezolve ecuația de gradul II cu coeficienți reali $ax^2+bx+c=0$, luând în considerare toate cazurile posibile. (R: pag.185)

8.3.2 Să se calculeze valoarea expresiei cu coeficienți complecși:

$$E = \begin{cases} 3z^2 - 6az + b & \text{dacă } |z| \leq 1 \\ 6az - b & \text{dacă } |z| > 1 \end{cases}$$

pentru o valoare dată a variabilei complexe z (a, b date, complexe). (R: pag.187)

Se dau trei puncte prin coordonatele lor să se stabilească dacă ele formează sau nu un triunghi echilateral.

```

10 REM
20 REM *** FORMEAZA TRIUNGHI ECHILATERAL ? ***
30 REM
40 CLS
50 PRINT "*****"
60 PRINT "*          FORMEAZA TRIUNGHI          *"
70 PRINT "*          ECHILATERAL ?            *"
80 PRINT "*****"
90 INPUT "Coordonatele lui A(x1,y1)" x1=";X1;"
y1=";Y1"
100 INPUT "Coordonatele lui B(x2,y2)" x2=";X2;"
y2=";Y2"
110 INPUT "Coordonatele lui C(x3,y3)" x3=";X3;"
y3=";Y3"
120 LET D1=SQR ((X1-X2)*(X1-X2)+(Y1-Y2)*(Y1-Y2))
130 LET D2=SQR ((X1-X3)*(X1-X3)+(Y1-Y3)*(Y1-Y3))
140 LET D3=SQR ((X2-X3)*(X2-X3)+(Y2-Y3)*(Y2-Y3))
150 PRINT "PUNCTELE:" A(";X1;", "Y1;") B("
;X2;", "Y2;") C(";X3;", "Y3;")
160 IF D1=D2 AND D1=D3 THEN PRINT "FORMEAZA TRIUNGHI
ECHILATERAL" DE LATURA L=";D1: GO TO 180
170 PRINT "NU FORMEAZA TRIUNGHI ECHILATERAL"
180 STOP

```

```

*****
*          FORMEAZA TRIUNGHI          *
*          ECHILATERAL ?            *
*****

```

PUNCTELE:
A(0,0)
B(3,0)
C(0,4)

NU FORMEAZA TRIUNGHI ECHILATERAL

PUNCTELE:
A(-3,0)
B(3,0)
C(0,5.1961524)

FORMEAZA TRIUNGHI ECHILATERAL
DE LATURA L=6

8.4.1. Se dau trei puncte prin coordonatele lor. Să se calculeze perimetrul și aria triunghiului determinat de cele trei puncte.

(R: pag.188)

8.4.2. Să se determine coordonatele centrului și raza cercului circumscris unui triunghi dat prin coordonatele celor trei vîrfuri.

(R: pag.190)

8.4.3. Se dau trei puncte necoliniare prin coordonatele lor. Să se verifice dacă ele formează sau nu un triunghi isoscel, scrierea mesajelor făcîndu-se obligatoriu la sfîrșitul algoritmului.

(R: pag.193)

Să se interclaseze două șiruri X și Y ordonate crescător.

INDICAȚIE:

Se compară primele elemente ale șirurilor date, x_i și y_i . Acela care este mai mic se depune în șirul pe care-l formăm și se mărește indicele corespunzător. Comparația se face în continuare între x_i și y_j . Cînd toate elementele unuia dintre cele două șiruri au fost depuse în noul șir, elementele rămase ale celuiilalt șir trebuie și ele depuse în continuare în noul șir.

algoritmul INTERCLASARE este:

```

citește n,m
[ pentru i=1,n execută
  citește  $x_i$ 
sfîrșit-pentru
[ pentru j=1,m execută
  citește  $y_j$ 
sfîrșit-pentru
i ← 1
j ← 1
l ← 0
s ← 0
[ cît-timp s=0 execută
  cheamă P
sfîrșit-cît-timp
[ pentru l=1,n+m execută
  scrie  $u_l$ 
sfîrșit-pentru
stop
  
```

procedura P este:

```

dacă  $x_i < y_j$ 
  atunci
    l ← l+1
     $u_l \leftarrow x_i$ 
    i ← i+1
  dacă  $i > n$ 
    atunci
      s ← 1
    cît-timp  $j \leq m$  execută
      l ← l+1
       $u_l \leftarrow y_j$ 
      j ← j+1
    sfîrșit-cît-timp
  sfîrșit-dacă
  altfel l ← l+1
     $u_l \leftarrow y_j$ 
    j ← j+1
  cheamă P1
sfîrșit-dacă
sfîrșit-procedură
  
```

procedura P1 este:

```

dacă  $j > m$ 
  atunci
    s ← 1
  cît-timp  $i \leq n$  execută
    l ← l+1
     $u_l \leftarrow x_i$ 
    i ← i+1
  sfîrșit-cît-timp
  sfîrșit-dacă
sfîrșit-procedură
  
```

```

10 CLS : LET z1=2: LET z2=28: GO SUB 430
20 PRINT " * DIN DOUA SIRURI ORDONATE *"
30 PRINT " * CRESCATOR SE FORMEAZA *"
40 PRINT " * UN NOU SIR *"
50 GO SUB 430
60 INPUT "Introduceti dim sirurilor N=";N;" M=";M
70 DIM X(N): DIM Y(M): DIM U(N+M)
80 PRINT "' ' " SIRURILE SINT:"' '
90 PRINT " X=";
100 FOR I=1 TO N
110 INPUT X(I): PRINT X(I);" ";
120 NEXT I
130 PRINT ")" "' ' " Y=";
140 FOR J=1 TO M
150 INPUT Y(J): PRINT Y(J);" ";
160 NEXT J
170 PRINT ")" "' '
180 LET I=1: LET J=1: LET L=0: LET S=0
190 IF S<>0 THEN GO TO 350
200 IF X(I)<Y(J) THEN GO TO 280
210 LET L=L+1: LET U(L)=Y(J): LET J=J+1
220 IF J<=M THEN GO TO 190
230 LET S=1
240 FOR K=I TO N
250 LET L=L+1: LET U(L)=X(K)
260 NEXT K
270 GO TO 190
280 LET L=L+1: LET U(L)=X(I): LET I=I+1
290 IF I<=N THEN GO TO 190
300 LET S=1
310 FOR K=J TO M
320 LET L=L+1: LET U(L)=Y(K)
330 NEXT K
340 GO TO 190
350 PRINT "' ' " SIRUL NOU FORMAT ESTE:"' ' " U=";
360 PRINT U(1);" ";
370 FOR I=2 TO N+M
380 IF U(I)<>U(I-1) THEN PRINT U(I);" ";
390 NEXT I
400 PRINT ")" "' '
410 PAUSE 0: CLS : STOP
420
430 FOR V=1 TO Z1: PRINT " ";: NEXT V
440 FOR V=1 TO Z2: PRINT "*"":: NEXT V
450 PRINT : RETURN

```

```

*****
* DIN DOUA SIRURI ORDONATE *
* CRESCATOR SE FORMEAZA *
* UN NOU SIR *
*****

```

SIRURILE SINT:

X=(-1 0 3 7 8 12)

Y=(-6 -3 1 5 9)

SIRUL NOU FORMAT ESTE:

U=(-6 -3 -1 0 1 3 5 7 8 9 12)

8.5.1 Să se genereze șirul cu n elemente (n fiind un număr natural dat) de forma:

$1, -1, -1, 1, -1, -1, -1, 1, -1, -1, -1, -1, 1, \dots$

(R: pag. 194)

8.5.2 Se consideră șirul de caractere S_n definit astfel: $S_1 = "a"$, $S_2 = "b"$, $S_{n+2} = S_n S_{n+1}$ pentru orice $n \geq 3$.

Să se calculeze, fără a număra, câte caractere "a" și câte caractere "b" conține termenul S_k , k fiind un număr natural dat și apoi să se tipărească primii k termeni ai șirului.

(R: pag. 195)

8.5.3. Se dă un șir ordonat crescător și un număr real y . Să se stabilească dacă există un element al șirului egal cu y folosind algoritmul de căutare binară.

(R: pag. 195)

8.5.4. Să se ordoneze crescător elementele unui șir dat folosind algoritmul de sortare prin inserție.

(R: pag. 197)

8.5.5. Să se rezolve ecuația $p_1 \cdot s = p_2$, unde p_1 și p_2 sînt două permutări ale mulțimii $\{1, 2, \dots, n\}$ date ca șiruri, fără a folosi alte șiruri.

(R: pag. 198)

8.5.6. Se dă un șir de numere reale. Să se verifice dacă șirul este sau nu o progresie aritmetică sau o progresie geometrică.

(R: pag. 201)

8.5.7 Să se genereze toate numerele întregi de câte trei cifre nenule citind în prealabil cifrele care pot să apară în configurația unui număr după cifrele $1, 2, \dots, 9$.

(R: pag. 203)

8.5.8 Să se formeze toate șirurile de cifre zecimale care satisfac următoarele proprietăți: primul element al șirului este 1 sau 2; șirul este strict crescător; diferența dintre oricare două elemente consecutive ale sale este 2 sau 3; ultimul element al șirului este 8 sau 9.

(R: pag. 205)

8.5.9 Fie A o matrice pătratică de dimensiune n . Să se treacă elementele sale într-un șir X în următoarea ordine:

$a_{11}, a_{21}, a_{22}, a_{12}, a_{31}, a_{32}, a_{33}, a_{23}, a_{13}, a_{41}, \dots$

(parcurgînd fiecare linie i de la primul său element pînă la cel aflat pe diagonala principală și apoi parcurgînd coloana i de la diagonala principală în sus). Să se regăsească apoi un element oarecare al matricei în șir fără a parcurge șirul (direct în funcție de indicii săi).

(R: pag. 207)

8.5.10 Reciproc, avînd șirul din problema anterioară, să se formeze matricea din care provine șirul și, dîndu-se un indice l , să se precizeze locul ocupat de elementul x_l al șirului în matrice.

(R: pag. 208)

8.5.11. Se dă o matrice A cu n linii și m coloane avînd drept elemente numere reale distincte două cite două. Dîndu-se doi indici, $1 \leq n$ și $k \leq m$, să se formeze șirul X definit în felul următor:

$x_1 = a_{lk}$;

$x_2 =$ maximul elementelor matricei care au indicele de linie mai mic strict decît l ;

x_3 = minimul elementelor matricei care au indicele de linie mai mic strict decât indicele de linie al lui x_2 ;

x_4 = maximul elementelor matricei care au indicele de linie mai mic strict decât indicele de linie al lui x_3 etc.

(R: pag. 210)

8.5.12 Se dă o matrice A pătratică cu n linii având drept elemente numerele 0 și 1 (adică o matrice booleană). Să se găsească șirul X de indici cu următoarele proprietăți: elementele șirului sînt distincte două câte două; oricare ar fi l și m două elemente consecutive ale sale, $a_{lm} = 1$; considerînd elementul x_i , succesorul său x_{i+1} este cel mai mic indice cu proprietățile cerute. Se cere să se determine cel mai lung șir cu proprietățile enunțate.

(R: pag. 212)

8.5.13 Se dă o matrice A cu n linii și m coloane. Să se rearanjeze liniile acestei matrice astfel încît elementele de pe prima coloană să fie în ordine crescătoare, iar acele linii care au aceleași elemente pe prima coloană să fie în ordinea crescătoare a elementelor primei coloane pe care ele au elemente diferite.

(R: pag. 213)

8.5.14 Să se elimine dintr-o matrice A cu n linii și m coloane linia l și coloana k , fără a folosi o nouă matrice.

(R: pag. 214)

8.5.15 Să se formeze matricea pătratică cu n linii care are:

- elementele de pe diagonale egale cu 1;
- elementele de pe primele paralele deasupra, respectiv dedesubtul celor două diagonale, care nu au luat anterior valoarea 1, egale cu 2
- elementele de pe cea de-a doua paralelă situată deasupra, respectiv dedesubtul celor două diagonale, care nu au primit anterior valoarea 1 sau 2, egale cu 3 etc.

(R: pag. 215)

8.5.16 Se dă o matrice pătratică A și un număr natural k . Să se determine matricea A^k

(R: pag. 217)

Fiind date m mulțimi să se genereze produsul lor cartezian.

INDICAȚIE:

Cardinalele mulțimilor se dau sub formă de șir, iar din elementele celor m mulțimi se formează o matrice cu m linii și numărul de coloane egal cu maximul cardinalelor celor m mulțimi. Elementele produsului cartezian se generează folosind un șir Y de dimensiune m ale cărui elemente au următoarea semnificație: elementul y_i indică al cîtelea element al mulțimii i se află pe locul i al elementului produsului cartezian ce se formează. Se pornește de la șirul $Y = (1, 1, \dots, 1)$. Pentru un anumit șir Y se generează următorul în ordine lexicografică ce corespunde unui element al produsului cartezian în felul următor: se găsește cel mai mare indice i cu proprietatea $y_i < n_i$ unde n_i este cardinalul mulțimii i . Succesorul lui Y este șirul $(y_1, \dots, y_{i-1}, y_i + 1, 1, \dots, 1)$.

```

60 PAUSE 100: CLS
70 INPUT "Introduceti numarul de multimi ";M
80 DIM N(M): DIM Y(M)
90 DIM A(M,20)
100 PRINT "    MULTIMILE SINT"^^
110 FOR I=1 TO M
120   INPUT "Cite elemente are multimea ";(I); "? ";N(I)
130   PRINT #0;"Introduceti elementele multimii ";i: PAUSE 50
140   PRINT "    {";
150   FOR J=1 TO N(I)
160     INPUT A(I,J): LET VV=A(I,J): GO SUB 450
170   NEXT J
180   PRINT "    }"
190 NEXT I
200 PRINT ^^ "    ELEMENTELE PRODUSULUI CARTE--"
210 PRINT "    ZIAN SINT:"^^^
220 LET KK=0
230 FOR I=1 TO M
240   LET Y(I)=1
250 NEXT I
260 IF KK=0 THEN PRINT
270 PRINT "    (";
280 FOR I=1 TO M
290   LET VV=A(I,Y(I)): GO SUB 450
300 NEXT I
310 PRINT "    )";: LET KK=1-KK
320 LET K=0
330 FOR J=M TO 1 STEP -1
340   IF Y(J)<N(J) THEN LET K=J: LET J=1
350 NEXT J
360 IF K=0 THEN GO TO 430
370 LET Y(K)=Y(K)+1
380 IF K=M THEN GO TO 260
390 FOR J=K+1 TO M
400   LET Y(J)=1
410 NEXT J
420 GO TO 260
430 PAUSE 0: CLS : STOP
440
450 FOR U=1 TO 3-LEN STR$(VV): PRINT " ";: NEXT U
460 PRINT VV;: RETURN

```

MULTIMILE SINT

```

{ 1 2 }
{ 3 4 }
{ 5 }

```

ELEMENTELE PRODUSULUI CARTE-
ZIAN SINT:

```

( 1 3 5 ) ( 1 4 5 )
( 2 3 5 ) ( 2 4 5 )

```

8.6.1 Se dau două șiruri X și Y cu n și respectiv m elemente. Să se verifice dacă șirurile reprezintă mulțimi; dacă nu, să se elimine elementele care se repetă, fără a folosi alte șiruri și apoi să se efectueze reuniunea, intersecția și diferența celor două mulțimi.

(R: pag. 219)

8.6.2 Fie X o mulțime cu n elemente reale, m un număr natural dat, a cel mai mare respectiv b cel mai mic element a mulțimii X . Se consideră intervalele:

$$I_j = [b + (j-1)h, b + jh], \quad j=1, 2, \dots, m-1 \quad \text{și}$$

$$I_m = [a-h, a] \quad \text{unde } h = (a-b)/m$$

Fie $A_j = X \cap I_j$, $j=1, 2, \dots, m$. Să se afișeze mulțimile A_j de cardinal maxim.

(R: pag. 221)

8.6.3 Se dă produsul cartezian a m mulțimi. Să se genereze mulțimile inițiale.

(R: pag. 224)

8.6.4 Se dau m submulțimi ale unei mulțimi cu k elemente. Să se numere și să se afișeze elementele care se găsesc în r submulțimi din cele m date, folosind doar două șiruri.

(R: pag. 225)

8.6.5 Să se genereze toate submulțimile unei mulțimi date sub formă de șir, folosind vectorul caracteristic.

(R: pag. 227)

8.6.6 Aceeași problemă, însă generarea submulțimilor să se facă în ordinea crescătoare a numărului lor de elemente.

(R: pag. 229)

8.6.7 Să se genereze permutările mulțimii $\{1, 2, \dots, n\}$ folosind vectorii de inversiune.

(R: pag. 230)

8.6.8 Să se genereze aranjamentele de k elemente ale mulțimii $\{1, 2, \dots, n\}$ folosind vectorii caracteristici.

(R: pag. 232)

8.6.9 Se dă o mulțime de numere. Să se scrie toate numerele care se pot obține ca sumă a maximum m elemente ale mulțimii date.

(R: pag. 233)

8.6.10 Folosind șiruri ordonate lexicografic, să se genereze permutările mulțimii $\{1, 2, \dots, n\}$.

(R: pag. 235)

8.6.11 Să se genereze, cu aceeași metodă, toate aranjamentele de n luate câte k .

(R: pag. 237)

8.6.12 Să se genereze, cu aceeași metodă, toate combinările de n luate câte k .

(R: pag. 239)

8.6.13 Să se genereze toate combinările de n luate câte k folosind vectorii caracteristici.

(R: pag. 240)

8.6.14 Să se genereze toate numerele de telefon de câte m cifre formate cu n cifre date.

(R: pag. 241)

8.6.15 Se dă un șir X cu n elemente. Să se formeze un nou șir Y cu n elemente, definit după cum urmează:

$$y_1 = 1 + (x_2 + x_3 + \dots + x_n) + (x_2 \cdot x_3 + x_2 \cdot x_4 + \dots + x_{n-1} \cdot x_n) + \dots + x_2 \cdot x_3 \cdot \dots \cdot x_n,$$

$$y_2 = 1 + (x_1 + x_3 + \dots + x_n) + (x_1 \cdot x_3 + x_1 \cdot x_4 + \dots + x_{n-1} \cdot x_n) + \dots + x_1 \cdot x_3 \cdot \dots \cdot x_n,$$

...

$$y_n = 1 + (x_1 + x_2 + \dots + x_{n-1}) + (x_1 \cdot x_2 + x_1 \cdot x_3 + \dots + x_{n-2} \cdot x_{n-1}) + \dots + x_1 \cdot x_2 \cdot \dots \cdot x_{n-1}$$

(R: pag. 243)

Se dă o funcție cu domeniul de definiție și codomeniul mulțimii finite. Să se studieze injectivitatea și surjectivitatea ei.

INDICAȚIE:

Au fost luate în considerare două reprezentări posibile ale funcțiilor. Prima reprezentare folosește un șir X ce are numărul de elemente, (n) , egal cu cardinalul domeniului de definiție, elementele șirului fiind numere naturale mai mici decât cardinalul codomeniului. Interpretarea este următoarea: $x_i = j$ înseamnă că imaginea prin funcția considerată a elementului al i -lea din domeniul de definiție este al j -lea din codomeniu. În această reprezentare faptul că funcția este bine definită se verifică prin apartenența la mulțimea $\{1, 2, \dots, m\}$ (m fiind cardinalul codomeniului) a fiecărui element al șirului. Pentru a verifica dacă funcția este injectivă, respectiv surjectivă se procedează în felul următor: se numără câte elemente distincte conține șirul ce definește funcția. Dacă acest număr este egal cu cardinalul domeniului de definiție, atunci funcția este injectivă, iar dacă el este egal cu cardinalul codomeniului, ea este surjectivă. Cea de-a doua reprezentare folosește o matrice booleană A cu n linii și m coloane, elementele ei fiind definite astfel: dacă elementului i din domeniul de definiție îi corespunde elementul j din codomeniu, atunci $a_{ij} = 1$, în caz contrar $a_{ij} = 0$. Funcția este bine definită dacă suma elementelor de pe fiecare linie este egală cu 1. Funcția este injectivă dacă suma elementelor de pe fiecare coloană este cel mult egală cu 1 și este surjectivă dacă această sumă este cel puțin egală cu 1

algoritmul *FUNCȚIE* este:

```

cheamă INTR
dacă caz=1 atunci cheamă NUMARA
    [
    dacă k=n atunci scrie 'funcție injectivă'
    altfel scrie 'funcția nu e inj.'
    sfîrșit-dacă
    dacă k=m atunci scrie 'funcție surject.'
    altfel scrie 'funcție nesurj.'
    sfîrșit-dacă
    altfel cheamă BINE_DEF
    [
    dacă sw=1 atunci scrie 'nu e bine def.'
    altfel cheamă P
    sfîrșit-dacă
    ]
sfîrșit-dacă
stop
    
```

procedura *INTR* este:

```

citește n,m,caz
dacă caz=1 atunci
    [
    pentru i=1,n execută
    citește f_i
    sfîrșit-pentru
    altfel
    [
    pentru i=1,n execută
    [
    pentru j=1,m execută
    citește a_ij
    sfîrșit-pentru
    sfîrșit-pentru
    ]
    ]
sfîrșit-dacă
sfîrșit-procedură
    
```



```

procedura NUMARA este:
  k ← 0
  i ← 1
  [ cît-timp i ≤ n execută
    [ dacă i in f atunci k ← k+1
      [ sfîrșit-dacă
        i ← i+1
      ]
    ]
  ]
sfîrșit-cît-timp
sfîrșit-procedură

```

```

procedura P este:
  cheamă INJECT
  [ dacă sw=1 atunci scrie 'funcția nu e injectivă'
    altfel scrie 'funcția e injectivă'
  ]
  sfîrșit-dacă
  cheamă SURJECT
  [ dacă sw=1 atunci scrie 'funcția nu e surjectivă'
    altfel scrie 'funcția e surjectivă'
  ]
  sfîrșit-dacă
sfîrșit-procedură

```

```

procedura BINE_DEF este:
  sw ← 0
  [ pentru i=1, n execută
    [ cheamă Q
      [ sfîrșit-pentru
        sfîrșit-procedură
      ]
    ]
  ]

```

```

procedura INJECT este:
  sw ← 0
  [ pentru j=1, m execută
    [ cheamă R
      [ dacă s > 1 atunci sw ← 1
        [ sfîrșit-dacă
          sfîrșit-pentru
            sfîrșit-procedură
        ]
      ]
    ]
  ]

```

```

procedura R este:
  [ pentru i=1, n execută
    [ s ← s + ai,j
      [ sfîrșit-pentru
        sfîrșit-procedură
      ]
    ]
  ]

```

```

procedura Q este:
  s ← 0
  [ pentru j=1, m execută
    [ s ← s + ai,j
      [ sfîrșit-pentru
        [ dacă s ≠ 1 atunci sw ← 1
          [ sfîrșit-dacă
            sfîrșit-procedură
          ]
        ]
      ]
    ]
  ]

```

```

procedura SURJECT este:
  sw ← 0
  [ pentru j=1, m execută
    [ cheamă R
      [ dacă s < 1 atunci sw ← 1
        [ sfîrșit-dacă
          sfîrșit-pentru
            sfîrșit-procedură
          ]
        ]
      ]
    ]
  ]

```

8.7.1 Se dă o relație binară definită între elementele unei mulțimi finite. Să se studieze proprietățile acesteia și să se precizeze dacă ea este de ordine sau de echivalență.

(R: pag. 245)

8.7.2 Se dă o operație binară definită între elementele unei mulțimi finite. Să se studieze proprietățile sale și să se precizeze dacă ea determină sau nu o structură de grup pe mulțimea de definiție.

(R: pag. 248)

9. PROBLEME PROPUSE

9.1 Să se descompună un număr natural dat (în toate modurile posibile) în produs de trei factori distincți.

9.2 Să se verifice dacă un număr natural dat n se divide cu 7, 11 respectiv 13 folosind următorul criteriu de divizibilitate: condiția necesară și suficientă ca un număr să se dividă cu 7, 11 respectiv 13 este ca modulul diferenței dintre numărul obținut cu ultimele sale 3 cifre și numărul obținut cu celelalte cifre ale sale, să se dividă cu 7, 11 respectiv 13.

9.3 Să se determine toate numerele naturale mai mici decât 1000 care au indicatorul lui Euler o putere a lui 2 (indicatorul lui Euler al unui număr natural n este egal cu numărul numerelor naturale mai mici decât n și prime cu el).

9.4 Să se găsească numerele naturale de trei cifre care sînt egale cu suma tuturor numerelor obținute cu cite două dintre cifrele sale.

9.5 Să se determine cel mai mic număr avînd prima cifră 1 cu următoarea proprietate: dacă această cifră 1 se mută la sfîrșitul numărului, se obține un număr de 3 ori mai mare. Se va lucra cu variabile de tip șir de caractere.

9.6 Se dă un număr real x și un număr natural n . Să se calculeze, efectuînd un număr minim de pași, suma:

$$S_n = \frac{1}{x+1} + \frac{2}{(x+1)(x+2)} + \dots + \frac{n}{(x+1)(x+2)\dots(x+n)}$$

9.7 Să se determine numărul variantelor în care se poate schimba o bancnotă de 100 lei avînd la dispoziție bancnote de 5, 10 și 25 lei și apoi să se afișeze aceste posibilități. Generalizare.

9.8 Fie X și Y două șiruri cu m respectiv n elemente. Să se decidă dacă elementele șirului Y apar ca elemente consecutive ale șirului X :

- în aceeași ordine în care sînt în șirul Y ;
- în orice ordine.

9.9 Să se genereze șirul 1,1,2,2,2,3,3,3,...

9.10 Se dă un șir cu n elemente reale dintre care ultimele m sînt nule. Să se insereze, dacă este posibil, între oricare două dintre primele $n-m$ elemente ale șirului media lor aritmetică, renunțînd la atîtea zerouri cite este necesar, dar fără a modifica dimensiunea șirului.

9.11 Se consideră un șir cu n elemente dintre care primele m sînt strict pozitive, ordonate crescător iar următoarele sînt nule. Dîndu-se un număr strict pozitiv y ,

- să se insereze numărul y în șir;
- să se elimine din șir elementele egale cu y fără a modifica dimensiunea șirului.

9.12 Să se rearanjeze liniile unei matrice astfel:

- elementele de pe o coloană dată l să fie ordonate crescător;
- elementele maximale de pe liniile matricei să fie ordonate crescător.

9.13 Se dă o matrice pătratică cu n linii, care are elementele situate pe o aceeași linie distincte două cite două. Să se rearanjeze liniile ei în așa fel încît maximul de pe fiecare linie să se afle pe diagonala principală, dacă acest lucru este posibil iar în caz contrar să se afișeze un mesaj.

9.14 Să se rearanjeze liniile unei matrice pătratică date și elementele de pe fiecare linie, astfel încît să fie satisfăcute simultan proprietățile:

- a) elementele de pe fiecare linie să fie în ordine crescătoare;
- b) suma elementelor de pe diagonala principală să fie minimă.

9.15 Se consideră un cub format din n^3 cubulețe. Cunoșcînd numărul asociat fiecărui cubuleț, să se calculeze sumele numerelor asociate cubulețelor de pe cele patru diagonale.

9.16 Se dau trei tije. Pe prima dintre ele sînt puse 8 inele de dimensiuni diferite, cele mai mici fiind puse deasupra celor mai mari. Să se depună inelele pe cea de-a treia tijă folosind-o pe cea de-a doua, fără a pune un inel mai mare peste unul mai mic (turnurile din Hanoi).

9.17 Se consideră o populație de viruși care se află în anumite noduri ale unei rețele de n^2 pătrățele. Un virus supraviețuiește dacă are doi sau trei vecini, altfel moare. Într-un nod liber cu doi sau trei vecini se naște un virus. Să se vizualizeze evoluția populației de viruși.

9.18 Se consideră n puncte în plan, date prin coordonatele lor. Să se determine care dintre tripletele formate cu punctele considerate determină triunghiuri echilaterale.

9.19 Să se calculeze coeficienții ecuațiilor dreptelor determinate de cîte două din n puncte date și care trec printr-un punct dat.

9.20 Se dau două cercuri C_1 și C_2 prin coordonatele centrelor și prin razele lor. Să se determine căreia dintre mulțimile $C_1 \cup C_2$, $C_1 \cap C_2$, $C_1 \setminus C_2$, $C_2 \setminus C_1$ și $C_1 \Delta C_2$ aparține un punct oarecare căruia i se cunosc coordonatele.

9.21 Se consideră un poligon convex cu n laturi și se notează cu H mulțimea triunghiurilor ce se formează cu virfurile poligonului. Să se numere și să se afișeze toate perechile de triunghiuri din H care au o latură comună și cel puțin un punct de intersecție în afara laturii comune.

9.22 Se dau triunghiurile ABC , MNP , XYZ înscrise succesiv unul în celălalt. Să se determine toate modurile în care se poate asocia fiecărui virf al celor trei triunghiuri cîte una dintre cifrele zecimale distincte astfel încît suma celor trei cifre de pe fiecare din laturile triunghiurilor ABC și MNP să fie egală cu un număr dat k .

9.23 Să se ordoneze în toate modurile posibile elementele mulțimii $\{1, 2, \dots, n\}$ astfel încît numerele 1, 2, 3 să fie unul după celălalt și în această ordine.

9.24 La 9 clase trebuie repartizați 3 profesori de matematică, fiecărui profesor repartizindu-i-se cîte 3 clase. Numerotînd clasele de la 1 la 9, să se afișeze toate modurile de repartizare posibile, calculînd în prealabil cîte sînt.

9.25 Din 11 persoane, dintre care 7 bărbați și 4 femei, trebuie formată o delegație de 5 persoane dintre care cel puțin două femei. Considerînd că se dau numele bărbaților, respectiv ale femeilor în cîte un șir de caractere, să se afișeze toate delegațiile ce se pot forma.

9.26 Dintr-un grup de 15 persoane aflate la o vîntoare se formează echipe de cîte 3 persoane, astfel încît nici una dintre cele 7 zile cît durează vîntoarea, nici un participant să nu fie împreună cu vreunul dintre cei cu care a mai fost în echipă. Să se genereze componența echipelor pentru fiecare zi, calculînd și numărul posibilităților existente.

9.27 Un bibliotecar dorește să distribuie 30 de cărți cu titluri diferite pe patru rafturi, neavînd importanță ordinea cărților așezate pe același raft. Să se afișeze modurile posibile de aranjare a lor.

9.28 La o cofetărie există 6 specialități de prăjitură. Să se afișeze modurile în care se pot forma cartioane diferite cu cîte 10 prăjituri fiecare.

9.29 Se consideră o mulțime cu n elemente și un număr natural k nenul mai mic decît n . Să se calculeze cîte submulțimi cu k elemente ale mulțimii date satisfac pe rînd condițiile de mai jos și să se afișeze aceste submulțimi:

- conțin p obiecte date;
- nu conțin nici unul din q obiecte date;
- conțin exact un obiect din p obiecte date;
- conțin un obiect dat dar nu conțin un altul;
- conțin cel puțin unul din p obiecte date;
- conțin r obiecte din p obiecte date dar nu conțin alte q obiecte date.

9.30 Se dau n litere într-un șir. Să se calculeze și să se afișeze aranjamentele celor n litere luate cîte m care:

- incep cu una, respectiv două litere date;
- conțin una, respectiv două litere date;
- incep cu p litere date, în ordinea în care acestea apar în șirul inițial;
- incep cu p litere date în orice ordine.

9.31 Să se calculeze coeficienții unui polinom cu rădăcini reale cunoscînd cele n rădăcini ale sale.

9.32 Fără a rezolva ecuația cu coeficienți reali $ax^3+bx^2+cx+d=0$ ($a \neq 0$), să se calculeze suma $S_n = x_1^n + x_2^n + x_3^n$, unde x_1, x_2, x_3 sînt rădăcinile ecuației date, iar n este un număr natural mai mare decît 3.

9.33 Se dau două polinoame prin coeficienții lor. Să se afle cmmdc al lor aplicînd algoritmul lui Euclid.

9.34 Pe mulțimea $\{1, 2, \dots, n\}$ se consideră definite două operații binare. Să se stabilească dacă acestea determină sau nu pe mulțimea dată o structură de:

- inel (comutativ sau nu, cu sau fără divizori ai lui zero);
- corp (comutativ sau nu).

9.35 Să se rezolve, folosind variabile tip șir de caractere și funcțiile STR\$ și VAL, problemele: *CONVERSIE*, 8.1.3, 8.1.4, 8.1.5, 8.2.3, 8.2.1, 8.5.7.

III. INDICAȚII ȘI SOLUȚII

1.1 Se citesc raza R , înălțimea H , după care se aplică formulele cunoscute și se afișează rezultatele obținute (valoarea lui π este implementată ca și constantă).

algoritmul 1.1 este:

citește r, h

$al1 \leftarrow 2 \cdot \pi \cdot r \cdot h$

$al2 \leftarrow \pi \cdot r \cdot \sqrt{h^2 + r^2}$

$at1 \leftarrow 2 \cdot \pi \cdot r \cdot (r+h)$

$at2 \leftarrow \pi \cdot r \cdot \left(r + \sqrt{h^2 + r^2} \right)$

$v1 \leftarrow \pi \cdot h \cdot r^2$

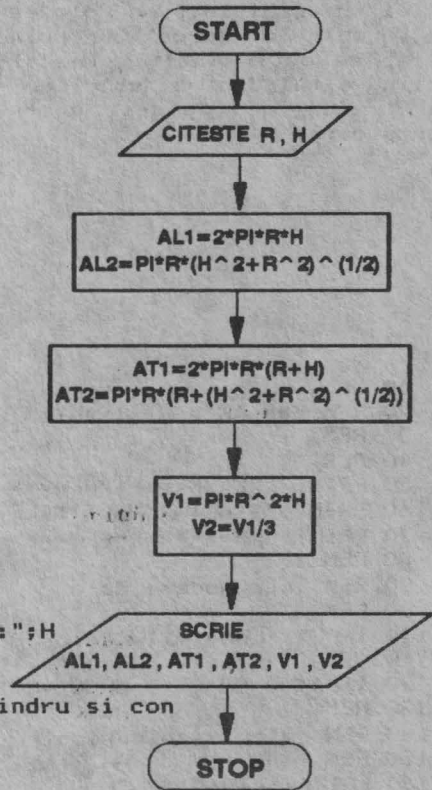
$v2 \leftarrow \frac{v1}{3}$

scrie $al1, al2, at1, at2, v1, v2$

stop

```

110 INPUT "Raza cilindrului:";R
120 IF R<=0 THEN GO TO 110
130 INPUT "Inaltimea cilindrului:";H
140 IF H<=0 THEN GO TO 130
150 REM
160 REM calcul arie laterala cilindru si con
170 REM
180 LET AL1=2*PI*R*H
190 LET AL2=PI*R*(H^2+R^2)^0.5
200 REM
210 REM calcul arie totala cilindru si con
220 REM
230 LET AT1=2*PI*R*(R+H)
240 LET AT2=PI*R*(R+SQR (H*H+R*R))
250 REM
260 REM calcul volume
270 REM
280 LET V1=PI*R^2*H
290 LET V2=V1/3
300 REM
310 REM extragere rezultate
320 REM
330 PRINT "Cilindrul cu R=";R;" si H=";H;" are:"
340 PRINT "A1=";AL1' "At=";AT1' "V=";V1'
350 PRINT "Conul cu R=";R;" si H=";H;" are:"
360 PRINT "A1=";AL2' "At=";AT2' "V=";V2
    
```



Cilindrul cu R=3 si H=5 are:
 Al=94.24778
 At=150.79645
 V=141.37167

Conul cu R=3 si H=5 are:
 Al=54.955427
 At=83.229761
 V=47.12389

1.2 Pentru calculul ariei triunghiului, se folosește formula lui Heron. Cunoscând aria, se obțin înălțimile calculând cîtul dintre dublul ariei și lungimea laturii corespunzătoare.

algoritmul 1.2 este:
 citește a,b,c

$$p \leftarrow \frac{a+b+c}{2}$$

$$s \leftarrow \sqrt{p \cdot (p-a) \cdot (p-b) \cdot (p-c)}$$

$$ha \leftarrow \frac{2 \cdot s}{a}$$

$$hb \leftarrow \frac{2 \cdot s}{b}$$

$$hc \leftarrow \frac{2 \cdot s}{c}$$

scrie s,ha,hb,hc
 stop

```

10 REM
20 REM *** ARIA SI INALTIMILE TRIUNGHIIULUI ***
30 REM
40 CLS
50 PRINT "*****"
60 PRINT "* ARIA/INALTIMILE TRIUNGHIIULUI *"
70 PRINT "*****"
80 REM
90 REM introducere date
100 REM
110 INPUT "Lungimile celor 3 laturi:";"A=";A;" B=";B;" C
=";C
120 IF A<=0 OR B<=0 OR C<=0 THEN GO TO 110
130 REM
140 REM calcul semiperimetru
150 REM
160 LET P=(A+B+C)/2
170 REM
180 REM calcul arie triunghi
190 REM
200 LET S=(P*(P-A)*(P-B)*(P-C))^(1/2)
210 REM
220 REM calcul inaltimi
230 LET HA=2*S/A
240 LET HB=2*S/B
250 LET HC=2*S/C
260 REM
270 REM afisare rezultate
280 REM
290 PRINT "Triunghiul cu laturile:";A;" ";B;" ";C;" are:"
300 PRINT " Aria S=";S
310 PRINT : PRINT "si inaltimile:"
320 PRINT " Ha=";HA;" Hb=";HB;" Hc=";HC

```

 * ARIA/INALTIMILE TRIUNGHIULUI *

Triunghiul cu laturile:3,4,5
 are:
 Aria S=6

si inaltimele:
 Ha=4 Hb=3 Hc=2.4

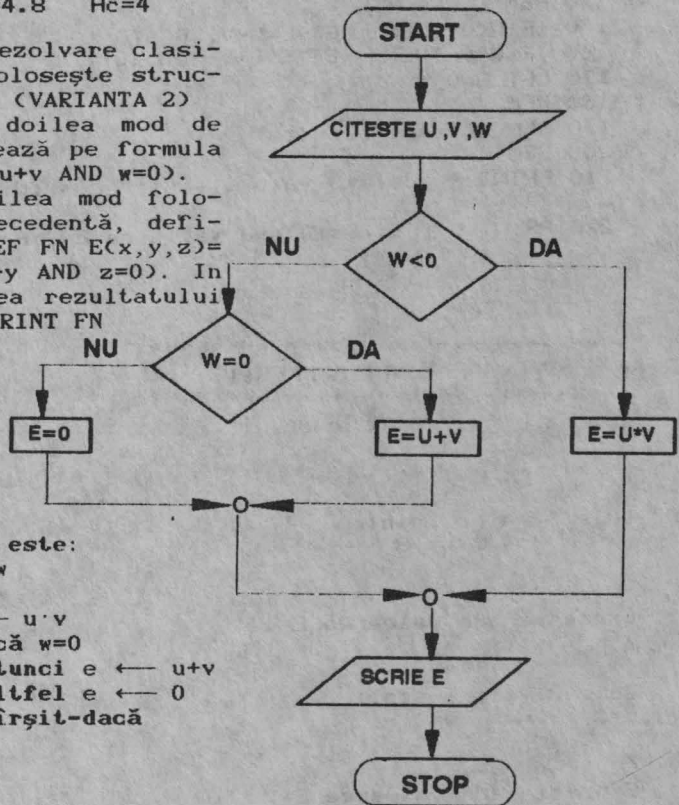
Triunghiul cu laturile:5,5,6
 are:
 Aria S=12

si inaltimele:
 Ha=4.8 Hb=4.8 Hc=4

1.3 0 rezolvare clasi-
 că a problemei folosește struc-
 tura alternativă. (VARIANTA 2)

Un al doilea mod de
 rezolvare se bazează pe formula
 $E=(u \cdot v \text{ AND } w < 0) + (u + v \text{ AND } w = 0)$.

Al treilea mod folo-
 sește formula precedentă, defi-
 nind o funcție DEF FN E(x,y,z)=
 $(x \cdot y \text{ AND } z < 0) + (x + y \text{ AND } z = 0)$. In
 acest caz scrierea rezultatului
 se face direct: PRINT FN
 E(u,v,w).



algoritmul 1.3 este:

citește u,v,w

dacă w < 0

atunci e ← u·v

altfel

dacă w = 0

atunci e ← u+v

altfel e ← 0

sfârșit-dacă

sfârșit-dacă

scrie e

stop

```

110 INPUT "Introdu cele 3 valori u,v,w:/" "u=";U;" v=";V;"
" w=";W
120 REM
150 LET E=(U*V AND W<0)+(U+V AND W=0)
160 REM
190 PRINT "//" "Cele 3 valori sint:/" "u=";U;" v=";V;" w="
;W
200 PRINT "//" "Expresia are valoarea E=";E
  
```

VARIANTA 2:

```

10 REM
20 REM *** VALOAREA UNEI EXPRESII ***
30 REM
40 CLS
50 PRINT "*****"
60 PRINT "*      VALOAREA UNEI EXPRESII      *"
70 PRINT "*****"
80 REM
90 REM introducere date
100 REM
110 INPUT "Introdu cele 3 valori u,v,w: " u="";U;"   v="";V;"
    " w="";W
120 REM
130 REM descriere structura alternativa
140 REM
150 IF W<0 THEN LET E=U*V: GO TO 190
160 IF W=0 THEN LET E=U+V: GO TO 190
170 LET E=0
180 REM
190 REM afisare rezultat
200 REM
210 PRINT ^^"Cele 3 valori sint:"^"u="";U;"   v="";V;"   w="
;W
220 PRINT ^^"Expresia are valoarea E=";E

```

```

*****
*      VALOAREA UNEI EXPRESII      *
*****

```

Cele 3 valori sint:
u=3 v=4 w=-8

Expresia are valoarea E=12

Cele 3 valori sint:
u=3 v=4 w=0

Expresia are valoarea E=7

Cele 3 valori sint:
u=3 v=4 w=5

Expresia are valoarea E=0

1.4

```

10 REM
20 REM *** VALOAREA UNEI FUNCTII ***
30 REM
40 CLS
50 PRINT "*****"
60 PRINT "*      VALOAREA UNEI FUNCTII      *"
70 PRINT "*****"
80 REM
90 REM introducere valoare
100 REM
110 INPUT "Introdu valoarea lui x:";x
120 REM
130 REM descriere structura alternativa
140 REM
150 IF x<5 THEN LET F=3*x-5: GO TO 180
160 IF x>10 THEN LET F=9*x+1: GO TO 180
170 LET F=10
180 REM
190 REM afisare rezultat
200 REM
210 PRINT "F(";x;")=";F

```

```

*****
*      VALOAREA UNEI FUNCTII      *
*****

```

```

F(1)=-2
F(7)=10
F(15)=136

```

```

10 REM
20 REM *** VALOAREA UNEI FUNCTII ***
30 REM
40 DEF FN F(x)=(3*x-5 AND x<5)+(10 AND (5<=x AND x<=10))+
(9*x+1 AND x>10)
50 CLS
60 PRINT "*****"
70 PRINT "*      VALOAREA UNEI FUNCTII      *"
80 PRINT "*****"
90 REM
100 REM introducere valoare x
110 REM
120 INPUT "Introdu valoarea lui x:";x
130 REM
140 REM afisare rezultat
150 REM
160 PRINT "F(";x;")=";FN F(x)

```

```

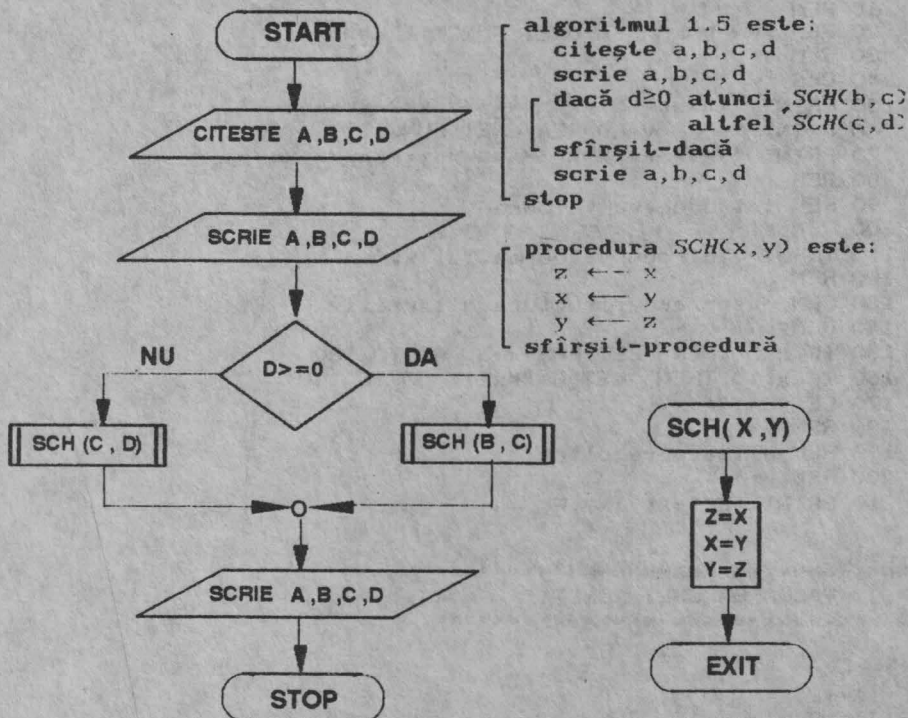
*****
*      VALOAREA UNEI FUNCTII      *
*****

```

```

F(1)=-2
F(7)=10
F(15)=136

```



```

40 CLS
50 PRINT "*****"
60 PRINT "* UTILIZARE PROCEDURA SCH(x,y) *"
70 PRINT "*****"
80 REM
90 REM introducerea celor 4 valori
100 REM
110 INPUT "Introdu cele 4 valori a,b,c,d: "/" "a=";A;" b=
";B;" c=";C;" d=";D
120 PRINT "Cele 4 valori sint:"/" ";A;" ";B;" ";C;
";D
130 REM
140 REM utilizare procedura
150 REM
160 IF D>=0 THEN LET X=B: LET Y=C: GO SUB 270: LET B=X:
LET C=Y: GO TO 190
170 LET X=C: LET Y=D: GO SUB 270: LET C=X: LET D=Y
180 REM
190 REM afisare rezultat
200 REM
210 PRINT "Dupa schimbare valorile sint:"
220 PRINT " ";A;" ";B;" ";C;" ";D
230 STOP
240 REM
250 REM procedura SCH
260 REM
270 LET Z=X: LET X=Y: LET Y=Z: RETURN
  
```

```
*****
* UTILIZARE PROCEDURA SCH(x,y) *
*****
```

Cele 4 valori sint:

1 2 3 4

Dupa schimbare valorile sint:

1 3 2 4

Cele 4 valori sint:

1 2 3 -4

Dupa schimbare valorile sint:

1 2 -4 3

Inversarea valorii a două variabile, de exemplu x și y, se face prin intermediul unei a treia variabile z, în felul următor: mutăm pe x în z (pentru a nu pierde valoarea sa) apoi mutăm pe y în x, iar în final y ia valoarea lui z (în care s-a păstrat valoarea inițială a lui x). Având în vedere că problema cere inversarea, în cazul $d < 0$, a lui b cu c și în caz contrar inversarea lui c cu d, s-a folosit în algoritm o procedură care realizează inversarea valorilor a două variabile. Procedura se apelează într-un caz cu parametrii efectivi b și c, iar în celălalt caz cu parametrii efectivi c și d.

1.6

```
10 REM
20 REM *** CALCUL VALOARE EXPRESIE ***
80 INPUT "Introdu cele 4 valori a,b,c:" a=";A;" b=";B;"
c=";C;" d=";D
90 PRINT "Cele 4 valori sint:" a " ;A;" b " ;B;" c " ;C;"
";D
100 REM
110 REM calcul valoare expresie
120 REM
130 IF A+C>2*D AND B>0 THEN LET E=D-3*B: GO TO 160
140 IF A+C>2*D AND B<0 THEN LET E=D+3*B: GO TO 160
150 LET E=4
160 PRINT "Valoarea expresiei este:" E=";E
```

Cele 4 valori sint:

1 2 3 -4

Valoarea expresiei este:

E=-10

Cele 4 valori sint:

1 -4 3 -2

Valoarea expresiei este:

E=-14

Cele 4 valori sint:

1 0 3 -2

Valoarea expresiei este:

E=4

Cele 4 valori sint:

1 2 3 4

Valoarea expresiei este:

E=4

1.7 Avind in vedere că in toate cazurile intervin modulele lui a și b , acestea se calculează înainte de a ramifica algoritmul in funcție de cele trei cazuri. Comparatia dintre $c+2d$ și 7 este necesară doar atunci cind $c+2d > 5$.

```

algoritmul 1.7 este:
citește a,b,c,d
dacă a ≥ 0 atunci x ← a
    altfel x ← -a
sfârșit-dacă
dacă b ≥ 0 atunci y ← b
    altfel y ← -b
sfârșit-dacă
dacă c+2·d ≥ 5
    atunci cheamă P
    altfel
        dacă x-3 ≥ y+6
            atunci f ← x-3
            altfel f ← y+6
        sfârșit-dacă
    sfârșit-dacă
scrie f
stop
    
```

```

procedura P este:
    dacă c+2·d ≤ 7
        atunci cheamă P1
        altfel cheamă P2
    sfârșit-dacă
sfârșit-procedură
    
```

```

procedura P1 este:
    dacă x-3 ≥ y+6
        atunci f ← y+6
        altfel f ← x-3
    sfârșit-dacă
sfârșit-procedură
    
```

```

procedura P2 este:
    dacă x+y ≥ 18
        atunci f ← 18
        altfel f ← x+y
    sfârșit-dacă
sfârșit-procedură
    
```

```

10 REM
20 REM *** CALCUL VALOARE EXPRESIE ***
30 REM
40 CLS
50 PRINT "*****"
60 PRINT "*   CALCUL VALOARE EXPRESIE   *"
70 PRINT "*****"
80 INPUT "Introdu cele 4 valori a,b,c,d: " a="";A;" b="";B;"
" c="";C;" d="";D
90 PRINT "Cele 4 valori sint: " a;" " b;" " c;" "
";D
100 REM
110 REM calcul valoare expresie
120 REM
130 IF A >= 0 THEN LET X=A: GO TO 150
140 LET X=-A
150 IF B >= 0 THEN LET Y=B: GO TO 170
160 LET Y=-B
170 IF C+2*D >= 5 THEN GO SUB 220: GO TO 200
180 IF X-3 >= Y+6 THEN LET F=X-3: GO TO 200
190 LET F=Y+6
200 PRINT "Valoarea expresiei este: " F;" E=";F
210 STOP
220 REM procedura P
230 IF C+2*D <= 7 THEN GO SUB 260: GO TO 250
240 GO SUB 300
250 RETURN
260 REM procedura P1
270 IF X-3 >= Y+6 THEN LET F=Y+6: GO TO 290
280 LET F=X-3
290 RETURN
300 REM procedura P2
310 IF X+Y >= 18 THEN LET F=18: GO TO 330
320 LET F=X+Y
330 RETURN
    
```

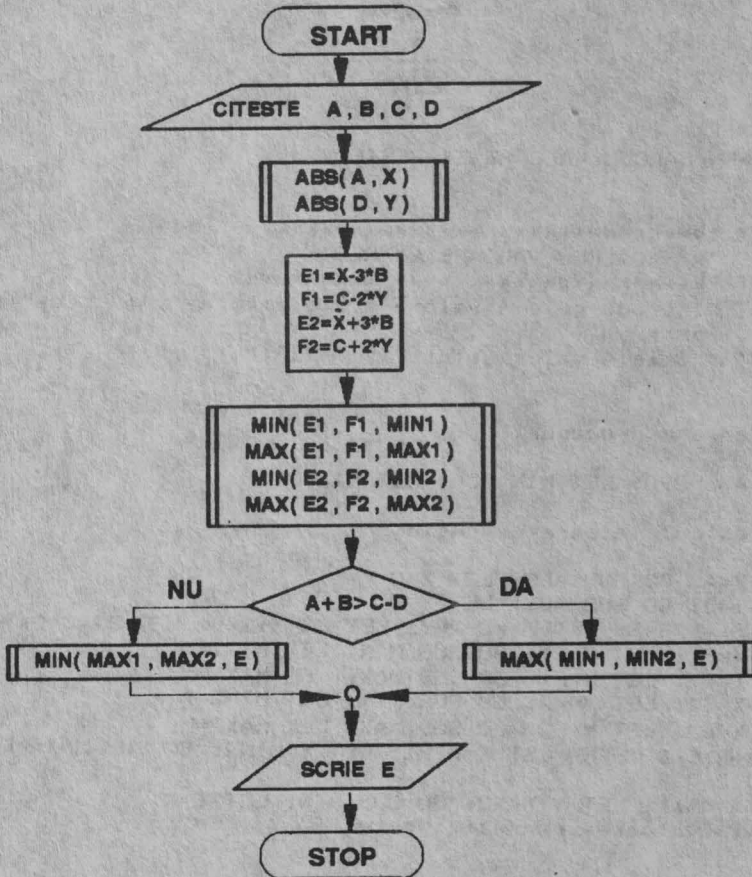
 * CALCUL VALOARE EXPRESIE *

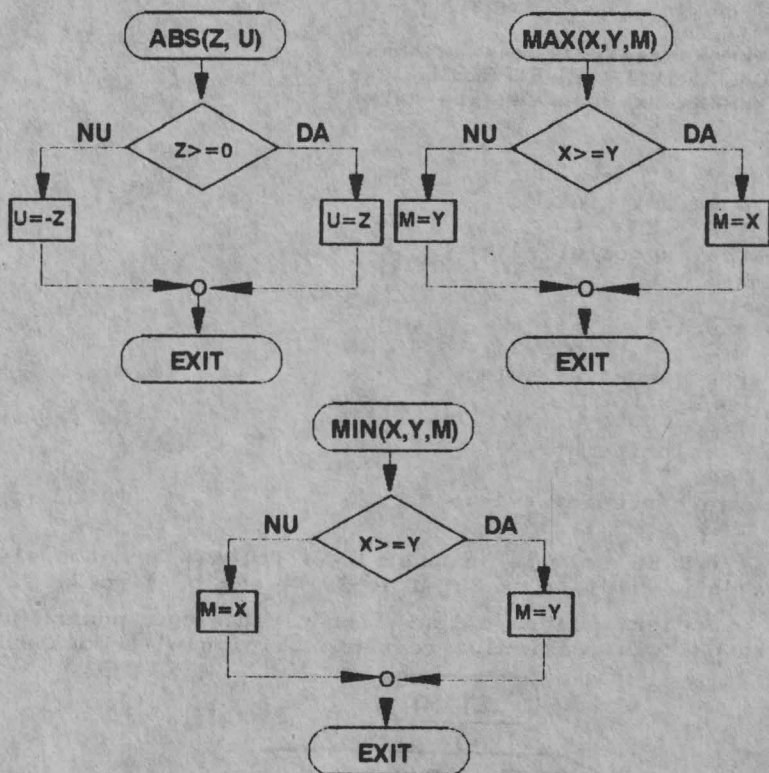
Cele 4 valori sint:
 1 2 3 -4
 Valoarea expresiei este:
 E=8

Cele 4 valori sint:
 1 -4 3 2
 Valoarea expresiei este:
 E=-2

Cele 4 valori sint:
 1 2 3 4
 Valoarea expresiei este:
 E=3

1.8 In soluția propusă s-au folosit următoarele notații: $x=|a|$, $y=|d|$, $e_1=x-3b$; $f_1=c-2y$, $e_2=x+3b$, $f_2=c+2y$ și s-au folosit proceduri pentru calculul modului unui număr, precum și pentru calculul maximului respectiv minimului a două numere.





```

10 REM
20 REM *** CALCUL VALOARE EXPRESIE ***
30 REM
40 CLS
50 PRINT "*****"
60 PRINT "*  CALCUL VALOARE EXPRESIE  *"
70 PRINT "*****"
80 INPUT "Introdu cele 4 valori a,b,c,d: " / "a=";A;" b=";B;"
c=";C;" d=";D
90 PRINT / "Cele 4 valori sint:" / " " ;A;" " ;B;" " ;C;" "
;D
100 REM
110 REM adrese proceduri
120 REM
130 LET ABS=290: LET MIN=330: LET MAX=370
140 REM
150 REM calcul valoare expresie
160 REM
170 LET Z=A: GO SUB ABS: LET X=U
180 LET Z=D: GO SUB ABS: LET Y=U
190 LET E1=X-3*B: LET F1=C-2*Y: LET E2=X+3*B: LET F2=C+2*Y
200 LET X=E1: LET Y=F1: GO SUB MIN: LET MIN1=M
210 LET X=E1: LET Y=F1: GO SUB MAX: LET MAX1=M
220 LET X=E2: LET Y=F2: GO SUB MIN: LET MIN2=M
230 LET X=E2: LET Y=F2: GO SUB MAX: LET MAX2=M
240 IF A+B>C-D THEN LET X=MIN1: LET Y=MIN2: GO SUB MAX: LET E
=M: GO TO 260
250 LET X=MAX1: LET Y=MAX2: GO SUB MIN: LET E=M
260 PRINT "Valoarea expresiei este:" / " " E=";E
270 STOP
  
```

```

280 REM procedura ABS
290 IF Z>=0 THEN LET U=Z: GO TO 310
300 LET U=-Z
310 RETURN
320 REM procedura MIN
330 IF X>=Y THEN LET M=Y: GO TO 350
340 LET M=X
350 RETURN
360 REM procedura MAX
370 IF X>=Y THEN LET M=X: GO TO 390
380 LET M=Y
390 RETURN

```

```

*****
*   CALCUL VALOARE EXPRESIE   *
*****

```

Cele 4 valori sint:

1 2 3 4

Valoarea expresiei este:

E=7

Cele 4 valori sint:

1 2 3 -4

Valoarea expresiei este:

E=-5

1.9 Lungimea laturii
pătratului, respectiv raza
cercului se calculează în n
pași cu formulele:

$$l_n = 1 \cdot \sqrt{2^{\lfloor n/2 \rfloor}}$$

$$r_n = 1 \cdot \sqrt{2^{\lfloor (n-1)/2 \rfloor - 1}}$$

În funcție de paritatea lui n
ultima figură circumscrisă es-
te pătrat sau cerc.

algoritmul 1.9 este:

citește l, n

$k \leftarrow \lfloor n/2 \rfloor$

$lp \leftarrow 1 \cdot \sqrt{2^k}$

$ap \leftarrow 1^2 \cdot 2^k$

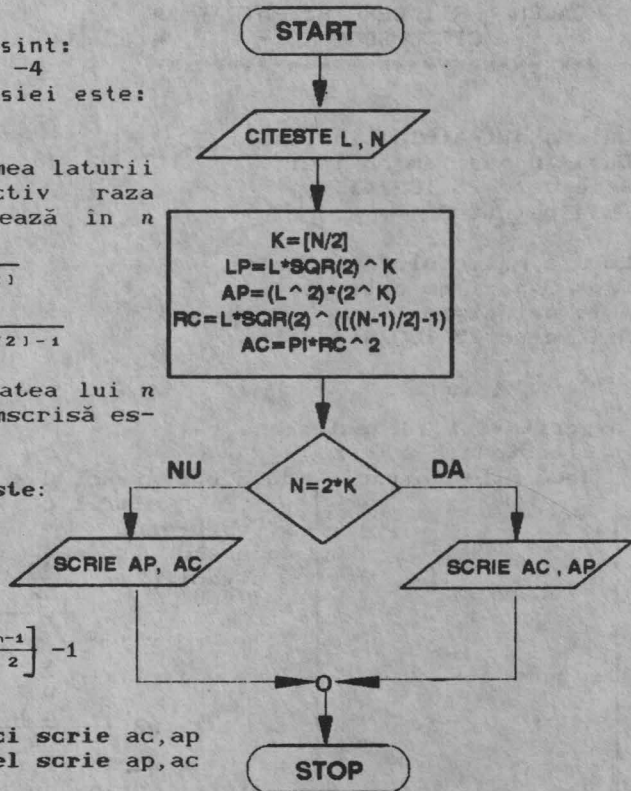
$rc \leftarrow 1 \cdot \left[\sqrt{2} \right]^{\lfloor \frac{n-1}{2} \rfloor - 1}$

$ac \leftarrow \pi \cdot rc^2$

dacă $n=2 \cdot k$ atunci scrie ac, ap
altfel scrie ap, ac

sfârșit-dacă

stop



```

20 REM *** CALCUL ARII CERC SI PATRAT CIRCUMSCRISE ***
30 REM
40 CLS
50 PRINT "*****"
60 PRINT "* CALCUL ARII CERC SI PATRAT *"
70 PRINT "* CIRCUMSCRISE *"
80 PRINT "*****"
90 INPUT "Introdu latura patratului l=";L
100 INPUT "Introdu numarul de pasi n=";N
110 LET K=INT (N/2)
120 LET LP=L*(SQR 2)^K
130 LET AP=L^2*2^K
140 LET RC=L*(SQR 2)^(INT ((N-1)/2)-1)
150 LET AC=PI*RC^2
160 PRINT "Latura patratului initial:";L
170 PRINT "Dupa ";N;" pasi am obtinut:"
180 IF N=2*K THEN PRINT "Arie cerc=";AC;"Arie patrat=";AP
: GO TO 200
190 PRINT "Arie patrat=";AP;"Arie cerc=";AC
200 STOP

```

```

*****
* CALCUL ARII CERC SI PATRAT *
* CIRCUMSCRISE *
*****

```

```

Latura patratului initial:1
Dupa 10 pasi am obtinut:
Arie cerc=25.132741
Arie patrat=32

```

```

Latura patratului initial:2
Dupa 5 pasi am obtinut:
Arie patrat=16
Arie cerc=25.132741

```

1.10

algoritmul 1.10. este:

citește a,b,c

dacă $a < b+c$ atunci

dacă $b \leq c$ atunci $d \leftarrow c-b$

altfel $d \leftarrow b-c$

sfârșit-dacă

dacă $a > d$

atunci $p \leftarrow \frac{a+b+c}{2}$

$s \leftarrow \sqrt{p \cdot (p-a) \cdot (p-b) \cdot (p-c)}$

$r1 \leftarrow \frac{s}{p}$

$r2 \leftarrow \frac{a \cdot b \cdot c}{4 \cdot s}$

scrie r1,r2

altfel scrie 'nu reprezintă

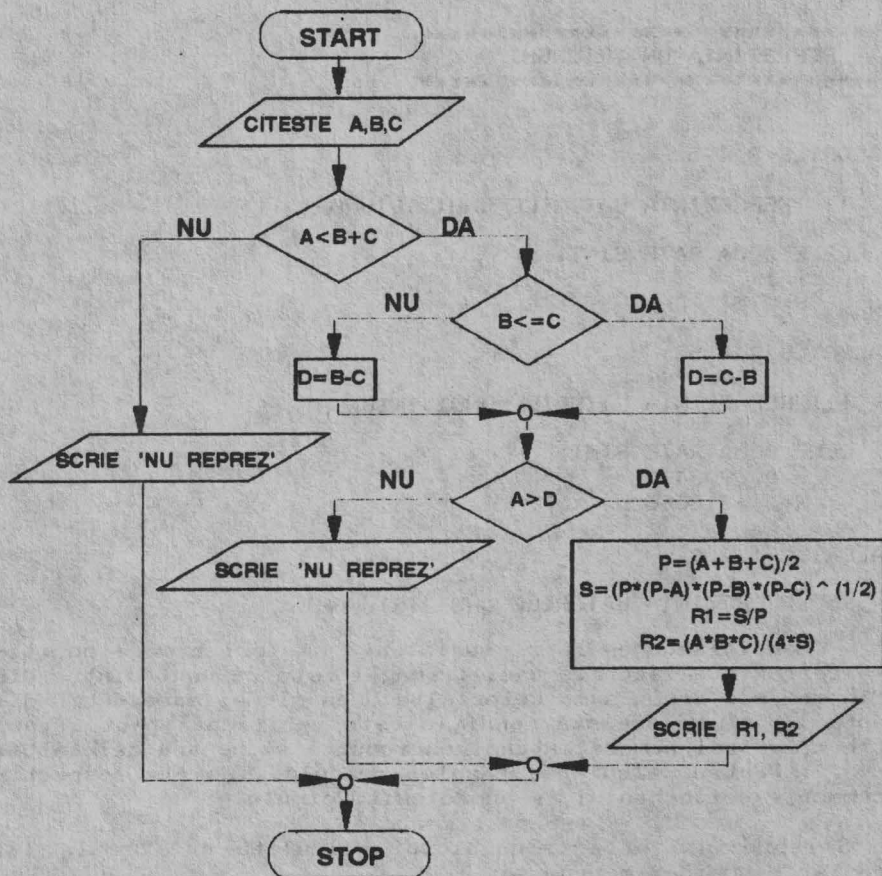
laturile unui triunghi'

sfârșit-dacă

altfel scrie 'nu reprezintă laturile
unui triunghi'

sfârșit-dacă

stop



```

10 REM
20 REM *** REPREZINTA UN TRIUNGHI ? ***
30 REM
40 CLS
50 PRINT "*****"
60 PRINT "* REPREZINTA UN TRIUNGHI ? *"
70 PRINT "*****"
80 INPUT "Cele 3 valori sint:" a=";A;" b=";B;" c=";C
90 IF A<=0 OR B<=0 OR C<=0 THEN GO TO 80
100 PRINT "VALORILE ";A;" ";B;" ";C
110 IF A>=B+C THEN PRINT " NU REPREZINTA LATURILE UNUI T
RIUNGHI": GO TO 200
120 IF B<=C THEN LET D=C-B: GO TO 140
130 LET D=B-C
140 IF A<=D THEN PRINT " NU REPREZINTA LATURILE UNUI TRI
UNGHI": GO TO 200
150 LET P=(A+B+C)/2
160 LET S=SQR (P*(P-A)*(P-B)*(P-C))
170 LET R1=S/P: LET R2=(A*B*C)/(4*S)
180 PRINT " !!! REPREZINTA LATURILE UNUI TRIUNGHI !!!"
CELE DOUA RAZE SINT:"
190 PRINT " r=";R1" R=";R2
200 STOP
  
```

```

*****
*   REPREZINTA UN TRIUNGHI ?   *
*****

```

VALORILE 3,4,5

!!! REPREZINTA LATURILE UNUI TRIUNGHI !!!

CELE DOUA RAZE SINT:

r=1
R=2.5

VALORILE 2,4,5

!!! REPREZINTA LATURILE UNUI TRIUNGHI !!!

CELE DOUA RAZE SINT:

r=0.69084928
R=2.6318068

VALORILE 1,2,3

NU REPREZINTA LATURILE UNUI TRIUNGHI

Condiția necesară și suficientă ca trei numere pozitive să reprezinte laturile unui triunghi este ca unul dintre ele să fie mai mic decât suma celorlalte două și mai mare decât diferența lor. Dacă această condiție este verificată pentru unul dintre cele trei numere, atunci ea rezultă și pentru celelalte.

Pentru calcularea razelor cercului înscris respectiv circumscris triunghiului, s-au folosit formulele:

$$r = S/p \quad R = abc/4S$$

unde S reprezintă aria triunghiului calculată cu formula lui Heron iar p semiperimetrul său.

1.11 Știind că:

$A < 90^\circ \Leftrightarrow a^2 < b^2 + c^2$, $A = 90^\circ \Leftrightarrow a^2 = b^2 + c^2$, $A > 90^\circ \Leftrightarrow a^2 > b^2 + c^2$
va trebui comparat pătratul celei mai mari dintre laturile triunghiului căreia i se opune unghiul cel mai mare cu suma pătratelor celorlalte două laturi.

```

10 REM
20 REM *** CE FEL DE TRIUNGHI ? ***
30 REM
40 CLS
50 LPRINT "*****"
60 LPRINT "*           CE FEL DE TRIUNGHI ?           *"
70 LPRINT "*****"

80 INPUT "Cele 3 valori sint:"/" a=";A;" b=";
B;" c=";C
90 REM
100 REM *** verificare valori a,b,c ***
110 REM
120 IF A<=0 OR B<=0 OR C<=0 THEN GO TO 80
130 IF A>=B+C THEN GO TO 80
140 IF B<=C THEN LET D=C-B: GO TO 160
150 LET D=B-C
160 IF A<=D THEN GO TO 80

```

```

170 REM
180 REM *** valorile a,b,c sint CORECTE!!! ***
190 REM
200 LPRINT "VALORILE ";A;" ";B;" ";C"
210 LPRINT "REPREZINTA LATURILE TRIUNGHIULUI "
"      *** ";
220 GO SUB 300: REM *** determina LMAX ***
230 IF LMAX^2<L1^2+L2^2 THEN LPRINT "ASCUTITUN
GHIC";: GO TO 260
240 IF LMAX^2=L1^2+L2^2 THEN LPRINT "DREPTUNGHI
C";: GO TO 260
250 LPRINT "OBTUZUNGHI";
260 LPRINT " ***": STOP
270 REM
280 REM *** latura maxima ***
290 REM
300 IF A>=B AND A>=C THEN LET LMAX=A: LET L1=B
: LET L2=C: RETURN
310 IF B>=A AND B>=C THEN LET LMAX=B: LET L1=A
: LET L2=C: RETURN
320 LET LMAX=C: LET L1=A: LET L2=B: RETURN

```

```

*****
*      CE FEL DE TRIUNGHI ?      *
*****

```

VALORILE 6,6,6

REPREZINTA LATURILE TRIUNGHIULUI
 *** ASCUTITUNGHI ***

VALORILE 6,8,10

REPREZINTA LATURILE TRIUNGHIULUI
 *** DREPTUNGHI ***

VALORILE 2,4,5

REPREZINTA LATURILE TRIUNGHIULUI
 *** OBTUZUNGHI ***

2.1 Calculul modulelor și însumarea lor se face într-o procedură apelată de trei ori. Având în vedere că numerele se citesc pe rând, procedura nu va avea parametri, citirea făcându-se în cadrul ei. Variabila în care se calculează suma se inițializează cu zero înainte de prima apelare a procedurii, iar calcularea mediei aritmetice se face după ultima sa apelare.

```

[ algoritmul 2.1. este:
  s ← 0
  cheamă P
  cheamă P
  cheamă P
  s ←  $\frac{s}{3}$ 
  scrie s
stop

```

```

[ procedura P este:
  citește x
  dacă not (x ≥ 0)
    atunci x ← -x
  sfârșit-dacă
  scrie x
  s ← s+x
sfârșit-procedură

```

```

10 REM
20 REM *** MEDIE ARITMETICA MODULE ***
30 REM
40 CLS
50 PRINT "*****"
60 PRINT "* MEDIE ARITMETICA MODULE *"
70 PRINT "*****"
80 LET S=0
90 PRINT "/"
100 GO SUB 140: GO SUB 140: GO SUB 140
110 LET S=S/3
120 PRINT ""MEDIA ARITMETICA A MODULELOR:"/" M=";S
130 STOP
140 INPUT "x=";X: PRINT X;" ";
150 IF X<0 THEN LET X=-X
160 LET S=S+X
170 RETURN

```

```

*****
* MEDIE ARITMETICA MODULE *
*****

```

7 -2 9

MEDIA ARITMETICA A MODULELOR:

M=6

-2 -8 -5

MEDIA ARITMETICA A MODULELOR:

M=5

6 7 9

MEDIA ARITMETICA A MODULELOR:

M=7.3333333

2.2 Un contor k , inițializat cu zero, va număra pentru câte dintre numerele citite este îndeplinită condiția cerută în enunț, iar o variabilă P , inițializată cu 1, va păstra produsul celorlalte numere. Sînt trei cazuri care se iau în considerare: $k=4$, caz în care toate numerele citite îndeplinesc condiția; $k=0$, ceea ce corespunde cazului în care nici unul dintre numere nu îndeplinește condiția; $k=1,2$, sau 3 caz în care se calculează atît contorul cit și produsul.

```

10 REM
20 REM *** CONTOR SI PRODUS ***
30 REM
40 CLS
50 PRINT "*****"
60 PRINT "*"      CONTOR SI PRODUS      "*"
70 PRINT "*****"
80 LET K=0: LET P=1
90 PRINT "  ";
100 GO SUB 150: GO SUB 150: GO SUB 150: GO SUB 150
110 IF K=0 THEN PRINT ""NICI UNUL DIN NUMERE NU DA R
EST 7 LA IMPARTIRE CU 13""PRODUSUL CELOR PATRU P=";P
: GO TO 140
120 IF K=4 THEN PRINT ""TOATE CELE 4 NUMERE DAU REST
7 LA IMPARTIRE CU 13": GO TO 140
130 PRINT ""K;" DIN CELE PATRU NUMERE DAU REST 7 LA IM
PARTIRE CU 13""PRODUSUL CELORLALTE ";4-K;" ESTE ";P
140 STOP
150 INPUT "x=";X
160 IF X-13*INT (X/13)=7 THEN LET K=K+1: PRINT X;" ";
: RETURN
170 LET P=P*X
180 RETURN

```

```

*****
*      CONTOR SI PRODUS      *
*****

```

7 20

2 DIN CELE PATRU NUMERE DAU REST 7 LA IMPARTIRE CU 13
PRODUSUL CELORLALTE 2 ESTE 55

NICI UNUL DIN NUMERE NU DA REST 7 LA IMPARTIRE CU 13
PRODUSUL CELOR PATRU P=24

137 20 7 33

TOATE CELE 4 NUMERE DAU REST 7 LA IMPARTIRE CU 13

2.3 Notăm numerele cu a, b, c, d iar contoarele cu k, l respectiv cu n . Divizibilitatea se verifică în procedura P care are un parametru formal x care primește pe rînd valorile numerelor ale căror proprietate de divizibilitate se verifică.

algoritmul 2.3. este:

citește a, b, c, d

$k \leftarrow 0$

$l \leftarrow 0$

$n \leftarrow 0$

cheamă $P(a)$

cheamă $P(b)$

cheamă $P(c)$

cheamă $P(d)$

scrie k, l, n

stop

procedura $P(x)$ este:

dacă $x=15 \cdot [x/15]$

atunci $k \leftarrow k+1$

$l \leftarrow l+1$

$n \leftarrow n+1$

altfel dacă $x=3 \cdot [x/3]$

atunci $k \leftarrow k+1$

altfel dacă $x=5 \cdot [x/5]$

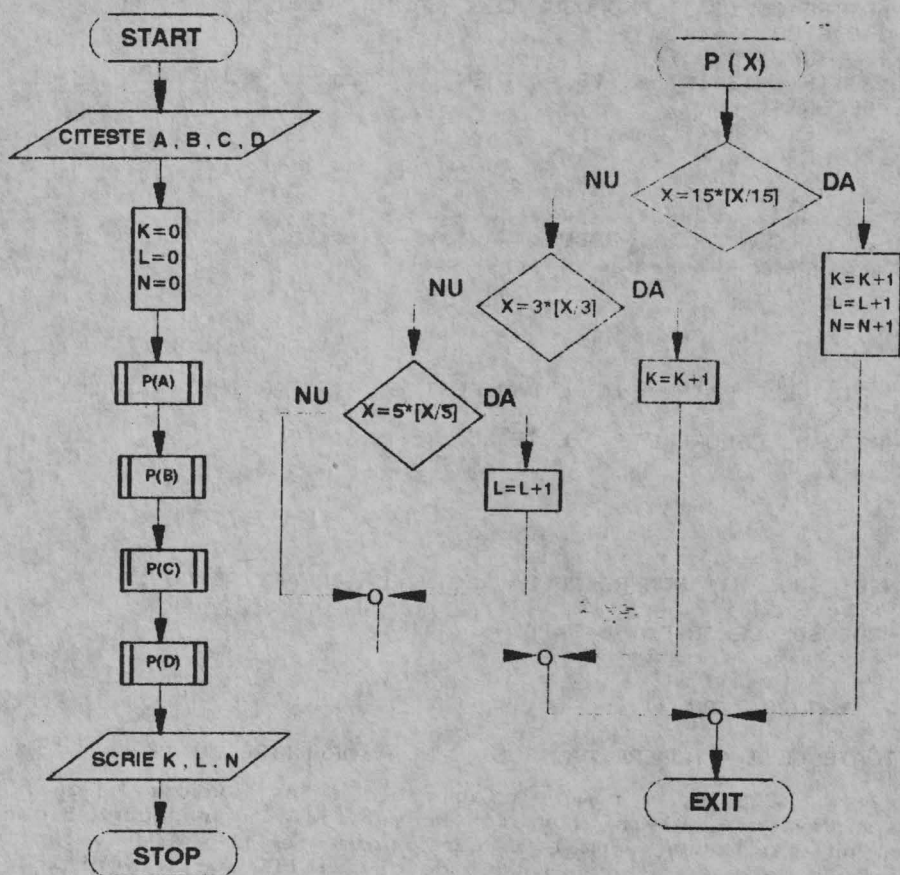
atunci $l \leftarrow l+1$

sfârșit-dacă

sfârșit-dacă

sfârșit-dacă

sfârșit-procedură



```

10 REM
20 REM *** CONTORIZARE DIVIZIBILITATE CU 3,5,15 ***
30 REM
40 CLS
50 PRINT "*****"
60 PRINT "* CONTORIZARE DIVIZIBILITATE *"
70 PRINT "* CU 3,5 SI 15 *"
80 PRINT "*****"
90 INPUT "A=";A;" B=";B;" C=";C;" D=";D
100 LET K=0: LET L=0: LET N=0
110 PRINT " ";
120 LET X=A: GO SUB 180: LET X=B: GO SUB 180
130 LET X=C: GO SUB 180: LET X=D: GO SUB 180
140 PRINT "' " ";K;" NUMERE SE DIVID CU 3"
150 PRINT " ";L;" NUMERE SE DIVID CU 5"
160 PRINT " ";N;" NUMERE SE DIVID CU 15"
170 STOP
180 PRINT X;" ";
190 IF X=15*INT (X/15) THEN LET K=K+1: LET L=L+1: LET
N=N+1: GO TO 220
200 IF X=3*INT (X/3) THEN LET K=K+1: GO TO 220
210 IF X=5*INT (X/5) THEN LET L=L+1
220 RETURN

```

```

*****
* CONTORIZARE DIVIZIBILITATE *
* CU 3,5 SI 15 *
*****

```

135 15 33 5

```

3 NUMERE SE DIVID CU 3
3 NUMERE SE DIVID CU 5
2 NUMERE SE DIVID CU 15

```

2.4 Sînt necesare patru inițializări: un produs, o sumă și două contoare, unul pentru numerele strict pozitive, celălalt pentru cele negative. Înainte de calculul celor două medii, trebuie testate contoarele, pentru a nu omite posibilitatea ca unul dintre ele să fie nul.

<pre> algoritmul 2.4. este: s ← 0 p ← 1 l ← 0 k ← 0 cheamă P cheamă P cheamă P dacă l=0 atunci scrie 'nu sînt numere negative' altfel s ← $\frac{s}{l}$ scrie s sfîrșit-dacă dacă k=0 atunci scrie 'nu sînt numere strict pozitive' altfel p ← $\sqrt[k]{p}$ scrie p sfîrșit-dacă stop </pre>	<pre> procedura P este: citește x dacă x>0 atunci p ← p·x k ← k+1 altfel s ← s+x l ← l+1 sfîrșit-dacă sfîrșit-procedură </pre>
---	---

```

10 REM
20 REM *** MEDIA ARITMETICA SI GEOMETRICA ***
30 REM
40 CLS
50 PRINT "*****"
60 PRINT "* MEDIA ARITMETICA/GEOMETRICA *"
70 PRINT "*****"
80 LET S=0: LET P=1: LET K=0: LET L=0
90 PRINT " ";
100 GO SUB 160: GO SUB 160: GO SUB 160
110 IF L=0 THEN PRINT "//NU SINT NUMERE NEGATIVE":
GO TO 130
120 LET S=S/L: PRINT "//MEDIA ARITMETICA CELOR NEGAT
IVE:"" MA=";S
130 IF K=0 THEN PRINT "//NU SINT NUMERE STRICT POZI
TIVE": GO TO 150
140 LET P=P*(1/K): PRINT "//MEDIA GEOMETRICA A CELOR
STRICT POZITIVE:"" MG=";P
150 STOP
160 INPUT "x=";X: PRINT X;" ";
170 IF X>0 THEN LET P=P*X: LET K=K+1: GO TO 190
180 LET S=S+X: LET L=L+1
190 RETURN
*****
* MEDIA ARITMETICA/GEOMETRICA *
*****

```

11 -7 8

MEDIA ARITMETICA CELOR NEGATIVE:
MA=-7

MEDIA GEOMETRICA A CELOR STRICT POZITIVE:
MG=9.3808315

-8 -5 3

MEDIA ARITMETICA CELOR NEGATIVE:
MA=-6.5

MEDIA GEOMETRICA A CELOR STRICT POZITIVE:
MG=3

2 4 7

NU SINT NUMERE NEGATIVE

MEDIA GEOMETRICA A CELOR STRICT POZITIVE:
MG=3.8258624

-2 -6 -23

MEDIA ARITMETICA CELOR NEGATIVE:
MA=-10.333333

NU SINT NUMERE STRICT POZITIVE

3.1

algoritmul 3.1. este:

```

k ← 0
citește x
cît-timp x≠0 execută
  [ dacă x/2=[x/2] atunci k ← k+1
  [ sfîrșit-dacă
  citește x
  sfîrșit-cît-timp
  dacă k=0 atunci scrie 'nici un număr par'
  altfel scrie k
  sfîrșit-dacă
stop

```

```

10 REM
20 REM *** CONTORIZARE NUMERE PARE ***
30 REM
40 CLS
50 PRINT "*****"
60 PRINT "* CONTORIZARE NUMERE PARE *"
70 PRINT "*****"
80 LET K=0
90 INPUT "x=";X
100 IF X<0 THEN GO TO 90
110 PRINT X;" ";
120 IF X=0 THEN GO TO 150
130 IF X/2=INT (X/2) THEN LET K=K+1
140 GO TO 90
150 PRINT ""
160 IF K=0 THEN PRINT "NICI UN NUMAR PAR": GO TO 180
170 PRINT "S-AU INTRODUS ";K;" NUMERE PARE"
180 STOP

```

```

*****
* CONTORIZARE NUMERE PARE *
*****

```

6 7 4 3 2 6 89 78 67 5 6 0

S-AU INTRODUS 6 NUMERE PARE

3.2

```

10 REM
20 REM *** CONTORIZARE NUMERE ***
30 REM
40 CLS
50 PRINT "*****"
60 PRINT "* CONTORIZARE NUMERE INTREGI *"
70 PRINT "* N=C1*7+5 *"
80 PRINT "* N=C2*13+7 *"
90 PRINT "*****"
100 LET K=0: LET L=0
110 INPUT "x=";X: PRINT X;" ";
120 IF X<>INT (X) THEN GO TO 160
130 IF X-7*INT (X/7)=5 THEN LET K=K+1
140 IF X-13*INT (X/13)=7 THEN LET L=L+1
150 GO TO 110
160 PRINT ""
170 PRINT K;" NUMERE AU PROPRIETATEA:"
180 PRINT " N=C1*7+5""
190 PRINT L;" NUMERE AU PROPRIETATEA:"
200 PRINT " N=C2*13+7""

```

```

*****
*  CONTORIZARE NUMERE INTREGI  *
*           N=C1*7+5           *
*           N=C2*13+7          *
*****

```

5 4 6 90 87 19 20 7 40 72 137 8 26 0.5

4 NUMERE AU PROPRIETATEA:
N=C1*7+5

4 NUMERE AU PROPRIETATEA:
N=C2*13+7

3.3 Deoarece știm că primul număr este diferit de 13, acesta se citește într-o variabilă, s-o numim x, care nu se prelucrează. Se citește apoi următorul număr într-o altă variabilă, numită y și se compară cu 13. Dacă y=13, algoritmul se încheie. Dacă nu, se verifică dacă y se divide la x, caz în care se scriu ambele variabile. Înainte de a citi alt număr, valoarea lui y trebuie mutată în x, pentru a nu se pierde.

```

10 REM
20 REM *** VERIFICARE DIVIZIBILITATE ***
30 REM
40 CLS
50 PRINT "*****"
60 PRINT "* VERIFICARE DIVIZIBILITATE *"
70 PRINT "* PERECHI DE NUMERE *"
80 PRINT "*****"
90 LET K=0: LET L=0
100 INPUT "x=";X
110 IF X<>INT(X) OR X<=0 OR X=13 THEN GO TO 100
120 INPUT "y=";Y
130 IF Y=13 THEN GO TO 170
140 IF Y/X=INT(Y/X) THEN LET U=Y: GO SUB 200: PRINT Y;" D
IVIZIBIL CU ";: LET U=X: GO SUB 200: PRINT X
150 LET X=Y
160 INPUT "y=";Y: GO TO 130
170 IF Y/X=INT(Y/X) THEN LET U=Y: GO SUB 200: PRINT Y;" D
IVIZIBIL CU ";: LET U=X: GO SUB 200: PRINT X
180 STOP
190 REM !!! scriere cu FORMAT a numerelor !!!
200 FOR I=1 TO 4-LEN(STR$(U)): PRINT " ";: NEXT I: RETURN

```

```

*****
* VERIFICARE DIVIZIBILITATE *
* PERECHI DE NUMERE *
*****

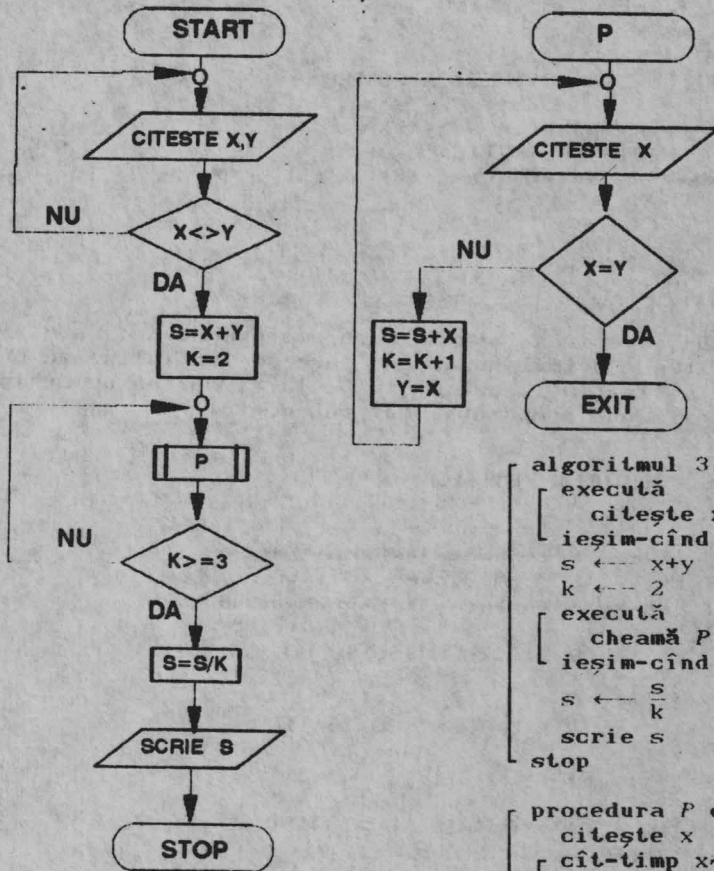
```

```

12 DIVIZIBIL CU 3
96 DIVIZIBIL CU 12
64 DIVIZIBIL CU 32
65 DIVIZIBIL CU 5
130 DIVIZIBIL CU 65
1300 DIVIZIBIL CU 130
90 DIVIZIBIL CU 45
9 DIVIZIBIL CU 3

```

3.4 Se citesc primele două numere, se calculează suma lor și se inițializează un contor cu valoarea 2 (pentru media aritmetică). Se citește următorul număr, care se compară cu precedentul. Dacă ele sînt egale, se calculează media aritmetică. Dacă nu, ultimul citit se adaugă la sumă, se reține și se mărește contorul cu 1, după care se citește următorul număr.



algoritmul 3.4. este:

```

execută
  citește x,y
  ieșim-cînd  $x \neq y$ 
   $s \leftarrow x+y$ 
   $k \leftarrow 2$ 
execută
  cheamă P
  ieșim-cînd  $k \geq 3$ 
   $s \leftarrow \frac{s}{k}$ 
  scrie s
stop
  
```

procedura P este:

```

citește x
cît-timp  $x \neq y$  execută
   $s \leftarrow s+x$ 
   $k \leftarrow k+1$ 
   $y \leftarrow x$ 
  citește x
sfîrșit-cît-timp
sfîrșit-procedură
  
```

```

10 REM
20 REM *** MEDIE ARITMETICA ***
30 REM
40 CLS
50 PRINT "*****"
60 PRINT "*   CALCUL MEDIE ARITMETICA   *"
70 PRINT "*****"
80 INPUT "x="; X
90 IF X<>INT X OR X<0 THEN GO TO 80
100 PRINT X; " ";
110 INPUT "y="; Y
120 IF Y=X OR Y<>INT Y OR Y<0 THEN GO TO 110
  
```

```

130 PRINT Y;" ";
140 LET S=X+Y: LET K=2
150 INPUT "x=";X
160 IF X<>INT X OR X<0 THEN GO TO 150
170 IF X=Y THEN GO TO 200
180 PRINT X;" ";
190 LET S=S+X: LET K=K+1: LET Y=X: GO TO 150
200 IF K<3 THEN GO TO 150
210 LET S=S/K
220 PRINT ""
230 PRINT "MEDIA ARITMETICA ESTE: ";S

```

```

*****
*   CALCUL MEDIE ARITMETICA   *
*****

```

3 6 89 7 0 23 11 3 5 0 7 6 3

MEDIA ARITMETICA ESTE:12.538462

3.5 Deosebirea față de problema precedentă este următoarea: se citește primul număr, cu care se inițializează suma, iar contorul se inițializează cu 1. De la al doilea număr începând, trebuie făcută comparația cu precedentul.

```

10 REM
20 REM *** MEDIE ARITMETICA ***
30 REM
40 CLS
50 PRINT "*****"
60 PRINT "*   CALCUL MEDIE ARITMETICA   *"
70 PRINT "*****"
80 INPUT "x=";X
90 IF X<>INT X OR X<0 THEN GO TO 80
100 PRINT X;" ";
110 INPUT "y=";Y
120 IF Y<>INT Y OR Y<0 THEN GO TO 110
130 PRINT Y;" ";
140 LET S=X+Y: LET K=2
150 INPUT "x=";X
160 IF X<>INT X OR X<0 THEN GO TO 150
170 IF X=Y THEN GO TO 200
180 PRINT X;" ";
190 LET S=S+X: LET K=K+1: LET Y=X: GO TO 150
200 LET S=S/K
210 PRINT ""
220 PRINT "MEDIA ARITMETICA ESTE: ";S

```

```

*****
*   CALCUL MEDIE ARITMETICA   *
*****

```

13 13

MEDIA ARITMETICA ESTE:13

2 4 5 7 0 3 4 55 61 1

MEDIA ARITMETICA ESTE:14.2

3.6 Atit suma cit și contorul se inițializează cu zero iar înainte de calculul mediei contorul se compară cu zero deoarece dacă el este nul înseamnă că primele două numere sînt egale, caz în care nu se calculează media aritmetică ci se afișează un mesaj.

algoritmul 3.6. este:

```

s ← 0
k ← 0
citește y
citește x
cît-timp x≠y execută
  s ← s+y
  k ← k+1
  y ← x
  citește x
sfîrșit-cît-timp
dacă k=0
  atunci scrie 'nu sînt numere cu proprietatea cerută'
  altfel s ←  $\frac{s}{k}$ 
  scrie s
sfîrșit-dacă

```

stop

```

10 REM
20 REM *** MEDIE ARITMETICA ***
30 REM
40 CLS
50 PRINT "*****"
60 PRINT "* CALCUL MEDIE ARITMETICA *"
70 PRINT "*****"
80 LET S=0: LET K=0
90 INPUT "y=";Y
100 IF Y<>INT Y OR Y<0 THEN GO TO 90
110 PRINT Y;" ";
120 INPUT "x=";X
130 IF X<>INT X OR X<0 THEN GO TO 120
140 IF X=Y THEN GO TO 170
150 PRINT X;" ";
160 LET S=S+Y: LET K=K+1: LET Y=X: GO TO 120
170 PRINT ""
180 IF K=0 THEN PRINT " NU SINT NUMERE CU PROPRIETAT
EA CERUTA": GO TO 210
190 LET S=S/K
200 PRINT "MEDIA ARITMETICA A PRIMELOR ";K;" NUMERE ES
TE:""" S=";S
210 STOP

```

```

*****
* CALCUL MEDIE ARITMETICA *
*****

```

11 33 55 6 3 0 4 5 1

MEDIA ARITMETICA A PRIMELOR 8 NUMERE ESTE:

S=14.625

13

NU SINT NUMERE CU PROPRIETATEA CERUTA

3.7 Se generează pe rând cifrele lui n^2 (prin împărțiri succesive cu 10). Imediat după generare se compară cifrele cu 5. Un contor, inițializat cu zero la începutul algoritmului, va număra cifrele egale cu 5.

algoritmul 3.7. este:

```

citește n
m ← n2 l ← 0
cît timp m ≠ 0 execută
    neamă P
sfârșit-cît-timp
scrie l
stop
    
```

procedura P este:

```

j ← [m/10]
k ← m-10·j
dacă k=5 atunci l ← l+1
sfârșit-dacă
m ← j
sfârșit-procedură
    
```

```

10 REM
20 REM *** CITE CIFRE 5 CONTINE UN NUMAR ? ***
30 REM
40 CLS
50 PRINT "*****"
60 PRINT "* CITE CIFRE 5 ARE UN NUMAR *"
70 PRINT "*****"
80 INPUT "n=";N
90 IF N<>INT N OR N<0 THEN GO TO 80
100 LET M=N*N: LET L=0
110 IF M<>0 THEN GO SUB 140: GO TO 110
120 PRINT "NUMARUL ";: GO SUB 180: PRINT N*N;" CONTINE
";L;" CIFRE 5"
130 STOP
140 LET J=INT (M/10): LET K=M-10*J
150 IF K=5 THEN LET L=L+1
160 LET M=J
170 RETURN
180 FOR u=1 TO 6-LEN (STR$ (n*n)): PRINT " ";: NEXT u:
RETURN
    
```

```

*****
* CITE CIFRE 5 ARE UN NUMAR *
*****
    
```

```

NUMARUL 50625 CONTINE 2 CIFRE 5
NUMARUL 1936 CONTINE 0 CIFRE 5
NUMARUL 65025 CONTINE 2 CIFRE 5
NUMARUL 25 CONTINE 1 CIFRE 5
NUMARUL 15129 CONTINE 1 CIFRE 5
    
```

3.8 Se inițializează suma cu zero, produsul cu 1 și două contoare cu zero. Valoarea switch-ului utilizat în program indică dacă numărul nenul citit va fi termen în sumă (sw=0) sau factor în produs (sw=1). Înainte de a trece la citirea următorului număr, switch-ul trebuie modificat (dacă a fost zero ia valoarea 1 și invers). După citirea numărului 0, înainte de scrierea sumei, respectiv a produsului, se testează contoarele pentru a vedea dacă unul sau ambele sînt nule.

algoritmul 3.8. este:

```
sw ← 0
s ← 0
p ← 1
k ← 0
l ← 0
citește x
cît-timp x≠0 execută
  cheamă P
  citește x
sfîrșit-cît-timp
dacă k=0 atunci scrie 'primul număr nul'
  altfel scrie s
  [ dacă l=0 atunci scrie 'al doilea număr nul'
    altfel scrie p
  sfîrșit-dacă
sfîrșit-dacă
stop
```

procedura P este:

```
dacă sw=0 atunci s ← s+x
  k ← k+1
  altfel p ← p·x
  l ← l+1
sfîrșit-dacă
sw ← 1-sw
```

sfîrșit-procedură

```
10 REM
20 REM *** SUMA SI PRODUS ***
30 REM
40 CLS
50 PRINT "*****"
60 PRINT "*          SUMA SI PRODUS          *"
70 PRINT "*****"
80 LET SW=0: LET S=0: LET P=1: LET K=0: LET L=0
90 INPUT "x=";X
100 IF X=0 THEN GO TO 120
110 GO SUB 180: GO TO 90
120 PRINT
130 IF K=0 THEN PRINT "PRIMUL NUMAR NUL": GO TO 170
140 PRINT "SUMA ESTE S=";S
150 IF L=0 THEN PRINT "AL DOILEA NUMAR NUL": GO TO 170
160 PRINT "PRODUSUL ESTE P=";P
170 STOP
180 IF SW=0 THEN PRINT X;" ";; LET S=S+X: LET K=K+1: GO
TO 200
190 PRINT INVERSE 1;X; INVERSE 0;" ";; LET P=P*X: LET L
=L+1
200 LET SW=1-SW
210 RETURN
```

```
*****
*          SUMA SI PRODUS          *
*****
```

4 4 1 3 54 -8 9 3 -7 11 3 9

SUMA ESTE S=22
PRODUSUL ESTE P=-10206

4.1 Se va folosi un contor care va lua $n-1$ valori, deoarece se cere media aritmetică a câte două elemente. Contorul va varia între 1 și $n-1$ sau între 2 și n . Primul număr va fi citit înaintea structurii repetitive și apoi, pentru fiecare valoare a contorului se va citi un nou număr.

algoritmul 4.1. este:
 citește n, y
 pentru $i=1, n-1$ execută
 cheamă P
 sfârșit-pentru
 stop

procedura P este:
 citește x
 $ma \leftarrow \frac{x+y}{2}$
 scrie ma
 $y \leftarrow x$
 sfârșit-procedură

```

10 REM
20 REM *** CALCUL MEDII ARITMETICE ***
30 REM
40 CLS
50 PRINT "*****"
60 PRINT "*   CALCUL MEDII ARITMETICE   *"
70 PRINT "*****"
80 INPUT "n=";N
90 IF N<=1 THEN GO TO 80
100 PRINT "SE VOR INTRODUCE ";N;" NUMERE"
110 PRINT "MEDIILE ARITMETICE A DOUA NUMERE CONSECUTIV
E SINT:"
120 INPUT "y=";Y
130 FOR I=1 TO N-1
140   INPUT "x=";X
150   LET MA=(X+Y)/2
160   PRINT MA;" "
170   LET Y=X
180 NEXT I

*****
*   CALCUL MEDII ARITMETICE   *
*****

```

SE VOR INTRODUCE 10 NUMERE

MEDIILE ARITMETICE A DOUA NUMERE CONSECUTIVE SINT:

5.5 15 -33 -44.5 2 8.5 -20.5 -15.5 7.5

4.2 b) Se cere calcularea unei sume de produse. În fiecare etapă se modifică produsul din etapa anterioară, iar produsul nou obținut se adaugă la sumă.

d) Se poate folosi o structură repetitivă al cărei contor variază între 1 și n pentru calcularea numărătorului și altă structură repetitivă al cărei contor variază între $n+1$ și $n+1$ pentru calcularea numitorului.

algoritmul 4.2.a. este:
 citește n
 $s \leftarrow 0$
 pentru $i=1, n, 2$ execută
 $s \leftarrow s+i^2$
 sfârșit-pentru
 scrie s
 stop

algoritmul 4.2.b. este:
 citește n
 $s \leftarrow 0$
 $p \leftarrow 1$
 pentru $i=1, n$ execută
 $p \leftarrow p \cdot i$
 $s \leftarrow s+p$
 sfârșit-pentru
 scrie s
 stop


```

10 REM
20 REM *** SUME DE PATRATE ***
30 REM
40 CLS
50 PRINT "*****"
60 PRINT "*   CALCUL SUME DE PATRATE   *"
70 PRINT "*****"
80 INPUT "n=";N
90 IF N<1 THEN GO TO 80
100 PRINT "N=";N
110 LET S=0
120 FOR I=1 TO N STEP 2
130   PRINT I;"^2";
140   LET S=S+I^2
150   IF I<N-1 THEN PRINT "+";
160 NEXT I
170 PRINT "=";S
180 PRINT "Suma patratelor numerelor " "naturale impare
<= ";n;" este:" " S=";S

```

```

*****
*   CALCUL SUME DE PATRATE   *
*****

```

N=5

$$1^2+3^2+5^2=35$$

Suma patratelor numerelor
naturale impare <= 5 este:

$$S=35$$

N=10

$$1^2+3^2+5^2+7^2+9^2=165$$

Suma patratelor numerelor
naturale impare <= 10 este:

$$S=165$$

N=15

$$1^2+3^2+5^2+7^2+9^2+11^2+13^2+15^2=680$$

Suma patratelor numerelor
naturale impare <= 15 este:

$$S=680$$

```

10 REM
20 REM *** SUME DE FACTORIALE ***
30 REM
40 CLS
50 PRINT "*****"
60 PRINT "* CALCUL SUME DE FACTORIALE *"
70 PRINT "*****"
80 INPUT "n=";N
90 IF N<1 THEN GO TO 80
100 PRINT "N=";N
110 LET S=0; LET P=1
120 FOR I=1 TO N
130 PRINT I;"!";
140 LET P=P*I
150 LET S=S+P
160 IF I<N THEN PRINT "+";
170 NEXT I
180 PRINT "=";S
190 PRINT "Suma factorialelor primelor ";n;"numere nat
urale este:" S=";S

```

```

*****
* CALCUL SUME DE FACTORIALE *
*****

```

N=5

1!+2!+3!+4!+5!=153

Suma factorialelor primelor 5
numere naturale este:

S=153

N=11

1!+2!+3!+4!+5!+6!+7!+8!+9!+10!+11!=43954713

Suma factorialelor primelor 11
numere naturale este:

S=43954713

N=3

1!+2!+3!=9

Suma factorialelor primelor 3
numere naturale este:

S=9

```

10 REM
20 REM *** SUMA DE 1/FACTORIAL ***
30 REM
40 CLS
50 PRINT "*****"
60 PRINT "* CALCUL SUMA DE 1/factorial *"
70 PRINT "*****"
80 INPUT "n=";N
90 IF N<1 THEN GO TO 80
100 PRINT "N=";N
110 LET S=0: LET P=1
120 FOR I=1 TO N
130 PRINT I;"/";I;"!";
140 LET P=P*I
150 LET S=S+1/P
160 IF I<N THEN PRINT "+";
170 NEXT I
180 PRINT "=";S
190 PRINT "Suma celor ";n;" termeni este:"
S=";S

```

```

*****
* CALCUL SUMA DE 1/factorial *
*****

```

N=5

$$1/1!+1/2!+1/3!+1/4!+1/5!=1.7166667$$

Suma celor 5 termeni este:

$$S=1.7166667$$

N=8

$$1/1!+1/2!+1/3!+1/4!+1/5!+1/6!+1/7!+1/8!=1.7182788$$

Suma celor 8 termeni este:

$$S=1.7182788$$

N=9

$$1/1!+1/2!+1/3!+1/4!+1/5!+1/6!+1/7!+1/8!+1/9!=1.7182815$$

Suma celor 9 termeni este:

$$S=1.7182815$$

N=10

$$1/1!+1/2!+1/3!+1/4!+1/5!+1/6!+1/7!+1/8!+1/9!+1/10!=1.7182818$$

Suma celor 10 termeni este:

$$S=1.7182818$$

```

10 REM
20 REM *** CALCUL EXPRESIE ***
30 REM
40 CLS
50 PRINT "*****"
60 PRINT "*          CALCUL CIT DE SUME          *"
70 PRINT "*****"
80 INPUT "n=";N
90 IF N<1 THEN GO TO 80
100 INPUT "l=";L
110 IF L<1 THEN GO TO 100
120 PRINT "/N=";N;"/"  "L=";L"/"
130 LET S1=0; LET S2=0
140 FOR I=1 TO N
150   LET S1=S1+I
160 NEXT I
170 FOR I=1 TO L
180   LET S2=S2+N+I
190 NEXT I
200 PRINT "          S1          ";S1"/"  "-----=";S1/S2"/"
      S2          ";S2

```

```

*****
*          CALCUL CIT DE SUME          *
*****

```

N=5 L=4

```

S1   15
-----=-----=0.5
S2   30

```

N=10 L=7

```

S1   55
-----=-----=0.56122449
S2   98

```

N=20 L=20

```

S1   210
-----=-----=0.3442623
S2   610

```

N=50 L=30

```

S1   1275
-----=-----=0.64885496
S2   1965

```

N=13 L=2

```

S1   91
-----=-----=3.137931
S2   29

```


4.3

```

20 REM *** SUMA SI PRODUS ***
30 REM
40 CLS
50 PRINT "*****"
60 PRINT "*          SUMA SI PRODUS          *"
70 PRINT "*****"
80 INPUT "Cite numere se introduc? n=";N
90 IF N<1 THEN GO TO 80
100 PRINT "Se introduc ";N;" numere"
110 LET S=0: LET P=1
120 FOR I=1 TO N
130   INPUT "numarul ";(i);" =";X
140   PRINT X;" ";
150   IF X<>0 THEN LET P=P*X
160   IF X>10 THEN LET S=S+X
170 NEXT I
180 PRINT
190 PRINT "Produsul numerelor nenule este:" P=
";P"
200 PRINT "Suma numerelor >10 este:" S=";S

```

```

*****
*          SUMA SI PRODUS          *
*****

```

Se introduc 10 numere

3 5 -8 2 33 11 -7 0 11.5 -1.1

Produsul numerelor nenule este:
P=-7714476

Suma numerelor >10 este:
S=55.5

4.4 Un contor care va număra divizorii proprii ai lui n se inițializează cu zero, apoi, într-o structură repetitivă se testează dacă numerele naturale cuprinse între 2 și $[n/2]$ îl divid pe n (aceste limite sînt cele între care se pot găsi divizorii proprii ai lui n). Divizorul găsit se scrie iar contorul se mărește cu 1. După ieșirea din structura repetitivă valoarea contorului indică dacă numărul este prim sau nu.

algoritmul 4.4. este:

```

citește n
k ← 0
[ pentru i=2, [n/2] execută
  [ cheamă P
  sfîrșit-pentru
dacă k=0 atunci scrie n, 'este prim'
  altfel scrie k
sfîrșit-dacă
stop

```

procedura P este:

```

[ dacă n/i=[n/i] atunci k ← k+1
  scrie i
sfîrșit-dacă
sfîrșit-procedură

```

```

10 REM
20 REM *** N ESTE NUMAR PRIM ? ***
30 REM
40 CLS
50 PRINT "*****"
60 PRINT "*      N ESTE NUMAR PRIM ?      *"
70 PRINT "*****"
80 INPUT "Cit este n? ";n
90 IF n<1 OR n<>INT n THEN GO TO 80
100 PRINT "N=";N
110 PRINT "  Divizori:"
120 LET k=0
130 FOR i=2 TO INT (n/2)
140 IF n/i=INT (n/i) THEN LET k=k+1: GO SUB 200: PRINT i
150 NEXT i
160 PRINT
170 IF k=0 THEN PRINT n;" este numar prim": GO TO 190
180 PRINT n;" are ";k;" divizori proprii"
190 STOP
200 FOR u=1 TO 12-LEN (STR$ (i)): PRINT " "; NEXT u: RETURN

```

```

*****
*      N ESTE NUMAR PRIM ?      *
*****

```

N=2345
Divizori:

5
7
35
67
335
469

2345 are 6 divizori proprii

N=1373
Divizori:

1373 este numar prim

N=256
Divizori:

2
4
8
16
32
64
128

256 are 7 divizori proprii

4.5

```

10 REM
20 REM *** NUMERE PRIME < 100 ***
30 REM
40 CLS
50 PRINT "*****"
60 PRINT "*      NUMERE PRIME < 100      *"
70 PRINT "*****"
80 LET nr=0
90 FOR n=2 TO 100
100  LET k=0
110  FOR i=2 TO INT (n/2)
120  IF n/i=INT (n/i) THEN LET k=1: LET i=n
130  NEXT i
140  IF k=0 THEN GO SUB 180: PRINT n: LET nr=nr+1
150 NEXT n
160 PRINT "Exista ";nr;" numere prime < 100"
170 STOP
180 FOR u=1 TO 4-LEN (STR$ n): PRINT " "; NEXT u: RETURN

```

```

*****
*      NUMERE PRIME < 100      *
*****

```

```

  2  3  5  7 11 13 17 19 23 29 31 37 41 43 47
53 59 61 67 71 73 79 83 89 97

```

Exista 25 numere prime < 100

4.6 Descriem doi algoritmi. Primul folosește un switch pentru a ști dacă numărul citit trebuie adăugat la prima sumă sau la cea de-a doua, numerele citindu-se într-o structură repetitivă cu pasul 1 de la 1 la n . A doua posibilitate este folosirea unei structuri repetitive cu pasul 2, în fiecare etapă urmînd a fi citite cîte două numere dintre care primul va fi adăugat primei sume iar celălalt celei de-a doua. La ieșirea din structura repetitivă ne situăm în unul dintre următoarele cazuri: dacă n este par au fost citite toate numerele, iar dacă n este impar se mai citește un număr care va fi adăugat sumei corespunzătoare numărătorului. În ambele variante, înainte de a calcula valoarea raportului trebuie considerată și posibilitatea ca numitorul să fie nul, caz în care se tipărește un mesaj.

VARIANTA 1

algoritmul 4.6.a. este:

```

citește n
sw ← 0
s1 ← 0
s2 ← 0
pentru i=1, n execută
  cheamă P
sfîrșit-pentru
dacă s2=0
  atunci scrie s1, 's2=0'
  altfel r ←  $\frac{s1}{s2}$ 
  scrie r
sfîrșit-dacă
stop

```

procedura P este:

```

citește x
dacă sw=0
  atunci s1 ← s1+x
  altfel s2 ← s2+x
sfîrșit-dacă
sw ← 1-sw
sfîrșit-procedură

```

```

10 REM
20 REM *** RAPORT DE SUME ALGEBRICE ***
30 REM
40 CLS
50 PRINT "*****"
60 PRINT "*   RAPORT DE SUME ALGEBRICE   *"
70 PRINT "*****"
80 INPUT "cite numere citești ? "; n
90 PRINT "Se citești "; n; " numere"
100 LET sw=0: LET s1=0: LET s2=0
110 FOR i=1 TO n
120   INPUT "elementul "; i; " : "; x
130 PRINT x; " ";
140   IF sw=0 THEN LET s1=s1+x: GO TO 160
150   LET s2=s2+x
160   LET sw=1-sw
170 NEXT i
180 PRINT ""
190 IF s2=0 THEN PRINT "   s1="; s1; "   s2="; s2: GO TO 240
200 LET r=s1/s2
210 PRINT "           "; s1
220 PRINT "   R=-----="; r
230 PRINT "           "; s2
240 STOP

```

```

*****
*   RAPORT DE SUME ALGEBRICE   *
*****

```

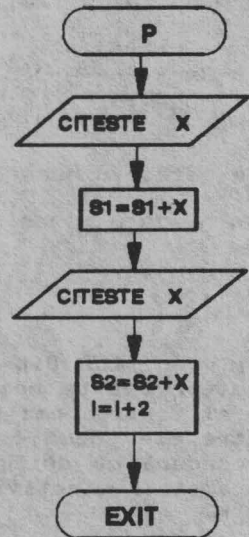
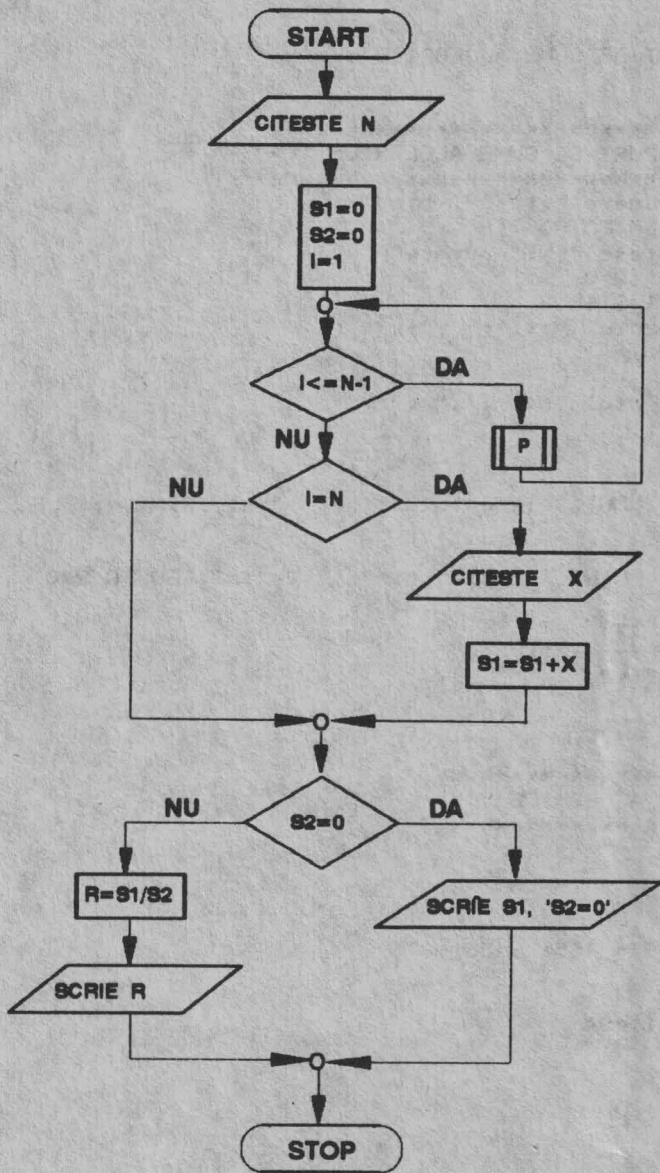
Se citești 10 numere

3 5 66 -89 0 123 -876 4651 -456 -876

-1263
 $R = \frac{-1263}{3814} = -0.3311484$

VARIANTA 2

<pre> [algoritmul 4.6.b. este: citește n s1 ← 0 s2 ← 0 pentru i=1,n-1,2 execută [cheamă P [sfârșit-pentru [dacă i=n atunci citește x [s1 ← s1+x [sfârșit-dacă [dacă s2=0 atunci scrie s1, 's2=0' [altfel r ← $\frac{s1}{s2}$ [scrie r [sfârșit-dacă stop </pre>	<pre> [procedura P este: citește x s1 ← s1+x citește x s2 ← s2+x sfârșit-procedură </pre>
--	--



```

10 REM
20 REM *** RAPORT DE SUME ALGEBRICE ***
30 REM
40 CLS
50 PRINT "*****"
60 PRINT "*   RAPORT DE SUME ALGEBRICE   *"
70 PRINT "*****"
80 INPUT "cite numere citesc ? ";n
90 IF n<0 THEN GO TO 80
100 PRINT "Se citesc ";N;" numere"
110 LET s1=0: LET s2=0
120 FOR i=1 TO n-1 STEP 2
130   INPUT "elementul ";(i);" :";x
140   PRINT x;" ";
150   LET s1=s1+x
160   INPUT "elementul ";(i+1);" :";x
170   PRINT x;" ";
180   LET s2=s2+x
190 NEXT i
200 IF i=n THEN INPUT "elementul ";(n);" ";x: PRINT x;: LE
T s1=s1+x
210 PRINT "
220 IF s2=0 THEN PRINT "   s1=";s1'   s2=";s2: GO TO 270
230 LET r=s1/s2
240 PRINT "           ";s1
250 PRINT "   R=-----=";r
260 PRINT "           ";s2
270 STOP

```

```

*****
*   RAPORT DE SUME ALGEBRICE   *
*****

```

Se citesc 10 numere

```

3 5 66 -89 0 123 -876 4651 -456 -876
-1263
R=-----=-0.3311484
3814

```

Se citesc 6 numere

```

34 0 -78 0 123 0
s1=79
s2=0

```

4.7 Din modul de definiție a expresiei rezultă că numitorul este nenul. Valoarea expresiei în primul și în cel de-al treilea caz se obține făcând aceleași calcule, schimbând între ele, dacă este nevoie, numerele m și n . Se va folosi o procedură cu doi parametri formali, apelată în primul caz cu parametrii efectivi n și m , iar în ultimul caz cu parametrii m și n .

algoritmul 4.7. este:

citește n, m

dacă $n \leq m+1$ atunci

- dacă $n=m+1$ atunci $e \leftarrow n$
- altfel cheamă $P(m, n)$
- sfârșit-dacă

altfel cheamă $P(n, m)$

sfârșit-dacă

scrie e

stop

procedura $P(x, y)$ este:

$p \leftarrow 1$

pentru $i=1, y$ execută

$p \leftarrow p \cdot (x+i)$

sfârșit-pentru

$e \leftarrow \frac{p}{x-y+1}$

sfârșit-procedură

```
10 REM
20 REM *** CALCUL VALOARE EXPRESIE ***
30 REM
40 CLS
50 PRINT "*****"
60 PRINT "*   CALCUL VALOARE EXPRESIE   *"
70 PRINT "*****"
80 INPUT "introdu n: "; n; " si m: "; m
90 IF n<0 OR m<0 OR n<>INT n OR m<>INT m THEN GO TO 80
100 IF n<=m+1 THEN GO TO 120
110 LET x=n: LET y=m: GO SUB 160: GO TO 140
120 IF n=m+1 THEN LET E=n: GO TO 140
130 LET x=m: LET y=n: GO SUB 160
140 PRINT "E("; n; ", "; m; ")="; E
150 STOP
160 LET p=1
170 FOR i=1 TO y
180   LET p=p*(x+i)
190 NEXT i
200 LET E=p/(x-y+1)
210 RETURN
```

```
*****
*   CALCUL VALOARE EXPRESIE   *
*****
```

$E(6, 3)=126$
 $E(6, 5)=6$
 $E(6, 8)=720720$

4.8 Observînd că $\frac{1 \cdot 2 \cdot \dots \cdot i}{1 \cdot 2 \cdot \dots \cdot (i+1)} = \frac{1}{i+1}$ pentru $i=1, 2, \dots, n$

deci: $u_1 = \frac{1}{2}, u_2 = \frac{1}{2} + \frac{1}{3}, \dots, u_n = \frac{1}{2} + \frac{1}{3} + \dots + \frac{1}{n} + \frac{1}{n+1}$.

Insumînd aceste egalități se obține suma cerută:

$$S_n = \frac{n}{2} + \frac{n-1}{3} + \dots + \frac{2}{n} + \frac{1}{n+1}$$

```

10 REM
20 REM *** CALCUL Sn=u1+u2+...+un ***
30 REM
40 CLS
50 PRINT "*****"
60 PRINT "*   CALCUL Sn=u1+u2+...+un   *"
70 PRINT "*****"
80 INPUT "introdu n:";n
90 IF n<0 OR n<>INT n THEN GO TO 80
100 LET sn=0
110 FOR k=1 TO n
120   LET sn=sn+(n+1-k)/(k+1)
130 NEXT k
140 PRINT "S";: LET x=n: GO SUB 160: PRINT n;"=";: LET
x=INT (sn): GO SUB 160: PRINT sn
150 STOP
160 FOR u=1 TO 4-LEN (STR$ (x))
170   PRINT " ";
180 NEXT u
190 RETURN

```

```

*****
*   CALCUL Sn=u1+u2+...+un   *
*****

```

```

S 1= 0.5
S 2= 1.3333333
S 3= 2.4166667
S 4= 3.7
S 5= 5.15
S 7= 8.4607143
S 9= 12.218651
S 10= 14.238528
S 30= 66.871846
S 100= 328.12241
S 300=1296.3678
S 500=2408.9994
S 1000=5499.4428

```

4.9 Folosind notația:

$$t_j = (-1)^j \cdot \frac{1 \cdot 3 \cdot \dots \cdot (2 \cdot j - 1)}{2 \cdot 4 \cdot \dots \cdot (2 \cdot j)} \cdot x^j$$

se obține $u_i = 1 + t_1 + \dots + t_i$ și $s_n = n \cdot t_1 + (n-1) \cdot t_2 + \dots + t_n$. Se inițializează suma cu zero și două produse cu 1 (pentru calculul numărătorului respectiv al numitorului fracției din definiția lui t_j). În acest fel este necesară o singură structură repetitivă. Având în vedere că nu se poate calcula $(-1)^j$ respectiv x^j pentru $x < 0$, calculatorul generând eroarea "invalid argument", trebuie calculate permanent valorile respective prin înmulțiri, în ciclu, cu -1, respectiv x, ținând cont că:

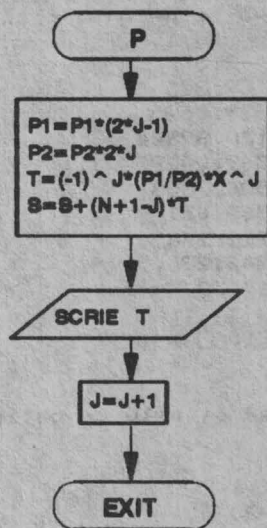
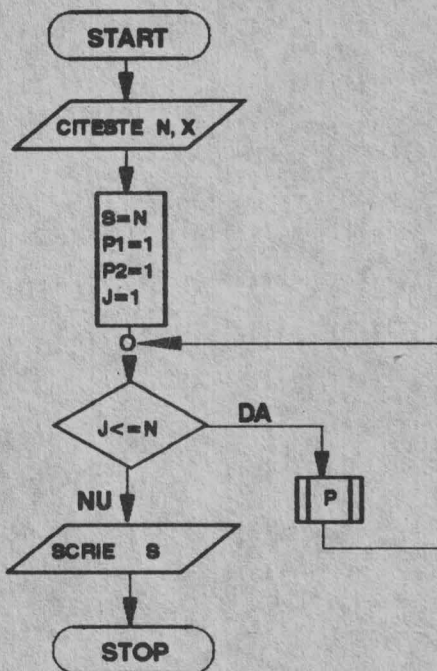
$$(-1)^j = (-1)^{j-1} \cdot (-1) \text{ și } x^j = x^{j-1} \cdot x$$

Pentru aceasta se fac inițializările:

$$\text{munu}=1 \text{ și } \text{px}=1$$

iar în ciclu se vor calcula astfel:

$$\text{munu}=\text{munu} \cdot (-1) \text{ și } \text{px}=\text{px} \cdot x$$



algoritmul 4.9. este:

```

citește n, x
s ← n
p1 ← 1
p2 ← 1
pentru j=1, n execută
  cheamă P
sfârșit-pentru
scrie s
stop
  
```

procedura P este:

```

p1 ← p1 · (2 · j - 1)
p2 ← p2 · 2 · j
t ← (-1)j ·  $\frac{p1}{p2}$  · xj
s ← s + (n + 1 - j) · t
scrie t
sfârșit-procedură
  
```

```

80 INPUT "introdu n:";n;" si valoarea lui x:";x
90 IF n<0 OR n<>INT n THEN GO TO 80
100 PRINT "'Pentru x=";x;" si n=";n;" se obtin:"
110 LET s=n: LET p1=1: LET p2=1: LET munu=1: LET px=x
120 FOR j=1 TO n
130 LET p1=p1*(2*j-1)
140 LET p2=p2*2*j
150 LET munu=munu*(-1)
160 LET px=px*x
170 LET t=munu*p1/p2*px
180 LET s=s+(n+1-j)*t
190 PRINT "t"; LET v=j: LET a=2
200 GO SUB 250: PRINT j;"=";
210 LET a=7: LET v=t: GO SUB 250: PRINT t
220 NEXT j
230 PRINT "'S";n;"=";s
240 STOP
250 FOR u=1 TO a-LEN (STR$ (INT v))
260 PRINT " ";
270 NEXT u
280 RETURN
  
```

Pentru $x=5$ si $n=10$ se obtin:

t 1= -2.5
 t 2= 9.375
 t 3= -39.0625
 t 4= 170.89844
 t 5= -769.04297
 t 6= 3524.7803
 t 7= -16365.051
 t 8= 76711.178
 t 9= -362247.23
 t10=1720674.3

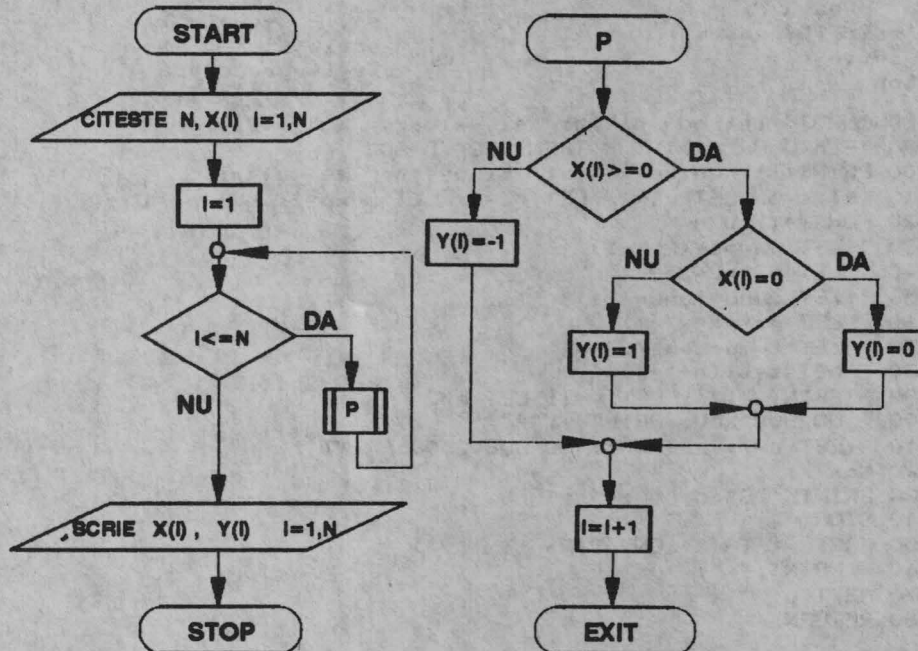
S10=1174816

Pentru $x=4$ si $n=10$ se obtin:

t 1= -2
 t 2= 6
 t 3= -20
 t 4= 70
 t 5= -252
 t 6= 924
 t 7= -3432
 t 8= 12870
 t 9= -48620
 t10= 184756

S10=115880

5.1 In afara solutiei prezentate, problema se poate rezolva folosind functia predefinita SGN deoarece pentru orice $i=1,2,\dots,n$ are loc egalitatea $y_i = \text{SGN}(x_i)$.



```

algoritmul 5.1. este:
  citește n
  pentru i=1, n execută
    citește  $x_i$ 
  sfârșit-pentru
  cheamă P
  pentru i=1, n execută
    scrie  $x_i$ 
  sfârșit-pentru
  pentru i=1, n execută
    scrie  $y_i$ 
  sfârșit-pentru
stop

```

```

procedura P este:
  pentru i=1, n execută
    dacă  $x_i \geq 0$ 
      atunci
        dacă  $x_i = 0$ 
          atunci  $y_i \leftarrow 0$ 
          altfel  $y_i \leftarrow 1$ 
        sfârșit-dacă
      altfel  $y_i \leftarrow -1$ 
    sfârșit-dacă
  sfârșit-pentru
sfârșit-procedură

```

```

10 REM
20 REM *** FORMARE SIR  $y(i)=SGN(x(i))$  ***
30 REM
40 CLS
50 PRINT "*****"
60 PRINT "* FORMARE SIR  $y(1)=SGN(x(1))$  *"
70 PRINT "*****"
80 INPUT "dimensiunea sirului n=" ; n
90 IF n<0 OR n<>INT n THEN GO TO 80
100 DIM x(n): DIM y(n)
110 FOR i=1 TO n
120 INPUT "x(";i);"=" ; x(i)
130 NEXT i
140 FOR i=1 TO n
150 IF x(i)<0 THEN LET y(i)=-1: GO TO 180
160 IF x(i)=0 THEN LET y(i)=0: GO TO 180
170 LET y(i)=1
180 NEXT i
190 PRINT "sirul initial x de dimensiune ";n;" este:"
200 FOR i=1 TO n
210 PRINT x(i);" ";
220 NEXT i
230 PRINT ""sirul nou format y este:"
240 FOR i=1 TO n
250 PRINT y(i);" ";
260 NEXT i
270 STOP

```

```

*****
* FORMARE SIR  $y(i)=SGN(x(i))$  *
*****

```

sirul initial x de dimensiune 10 este:

-8 0 2 6 3 -3 0 0 -7 5

sirul nou format y este:

-1 0 1 1 1 -1 0 0 -1 1

5.2 Intr-o structură repetitivă de contor i ce variază între 1 și $\lfloor n/2 \rfloor$ (n este dimensiunea șirului) se inversează, pentru fiecare valoare a lui i , elementele de indice i și $n-i+1$. Dacă problema ar fi cerut doar afișarea elementelor șirului în ordine inversă nu ar fi fost necesară modificarea șirului, ci doar afișarea acestuia de la ultimul element la primul.

algoritmul 5.2. este:
 citește n
 pentru $i=1, n$ execută
 citește x_i
 sfârșit-pentru
 pentru $i=1, n$ execută
 scrie x_i
 sfârșit-pentru
 pentru $i=1, \lfloor n/2 \rfloor$ execută
 $y \leftarrow x_i$
 $x_i \leftarrow x_{n-i+1}$
 $x_{n-i+1} \leftarrow y$
 sfârșit-pentru
 pentru $i=1, n$ execută
 scrie x_i
 sfârșit-pentru
 stop

```

40 CLS
50 PRINT "*****"
60 PRINT "* INVERSARE ELEMENTE IN SIR *"
70 PRINT "*****"
80 INPUT "dimensiunea sirului n= ";n
90 IF n<=0 OR n<>INT n THEN GO TO 80
100
110 DIM x(n): REM dimensionare sir
120
130 PRINT "sirul initial x de dimensiune ";n;" este:"
140
150 FOR i=1 TO n
160 INPUT "x(";i);")=";x(i)
170 PRINT x(i);" ";
180 NEXT i
190
200 FOR i=1 TO INT (n/2)
210 LET y=x(i)
220 LET x(i)=x(n+1-i)
230 LET x(n+1-i)=y
240 NEXT i
250
260 PRINT ""sirul inversat este:"
270 FOR i=1 TO n
280 PRINT x(i);" ";
290 NEXT i
300 PRINT
310 STOP

```

 * INVERSARE ELEMENTE IN SIR *

sirul initial x de dimensiune 11 este:

3 6 -9 0 1 -87 34 9 2 0 33

sirul inversat este:

33 0 2 9 34 -87 1 0 -9 6 3

5.3

```

10 REM
20 REM *** FORMARE SIR ***
30 REM
40 CLS
80 INPUT "dimensiunea sirului n= ";n
90 IF n<=0 OR n<>INT n THEN GO TO 80
100
110 DIM x(n): DIM y(n): REM dimensionare siruri
120
130 PRINT "sirul initial x de dimensiune ";n;" este:"
140
150 FOR i=1 TO n
160   INPUT "x(";i);"=";x(i)
170   PRINT x(i);" ";
180   IF x(i)/2<>INT (x(i)/2) THEN LET y(i)=INT (x(i)/2):
GO TO 210
190   IF x(i)/6=INT (x(i)/6) THEN LET y(i)=x(i)/6: GO TO
210
200   LET y(i)=x(i)
210 NEXT i
220
230 PRINT ""sirul nou format y este:"
240 FOR i=1 TO n
250   PRINT y(i);" ";
260 NEXT i
270 PRINT
280 STOP

```

sirul initial x de dimensiune 13 este:

5 3 -8 -9 0 12 18 -6 -36 98 32 -12 100

sirul nou format y este:

2 1 -8 -5 0 2 3 -1 -6 98 32 -2 100

5.4 Se inițializează suma, maximul și minimul cu primul element al șirului, apoi se parcurge șirul începînd cu al doilea element. Fiecare element se adună la sumă și se compară cu maximul și minimul, care se modifică dacă este cazul.

algoritmul 5.4. este:

```

citește n
pentru i=1,n execută
  citește  $x_i$ 
sfîrșit-pentru
 $s \leftarrow x_1$ 
 $\max \leftarrow x_1$ 
 $\min \leftarrow x_1$ 
pentru i=2,n execută
  cheamă P
sfîrșit-pentru
scrie s,max,min
stop

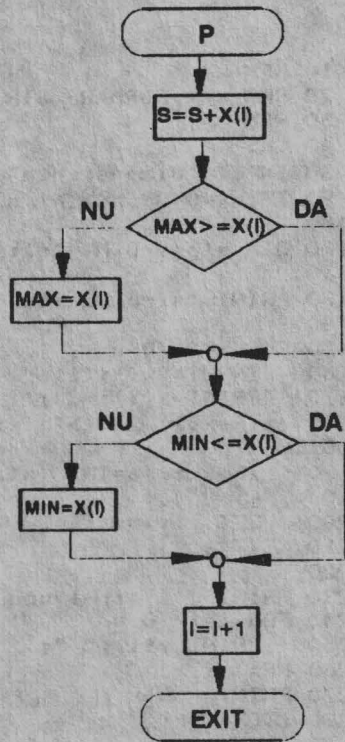
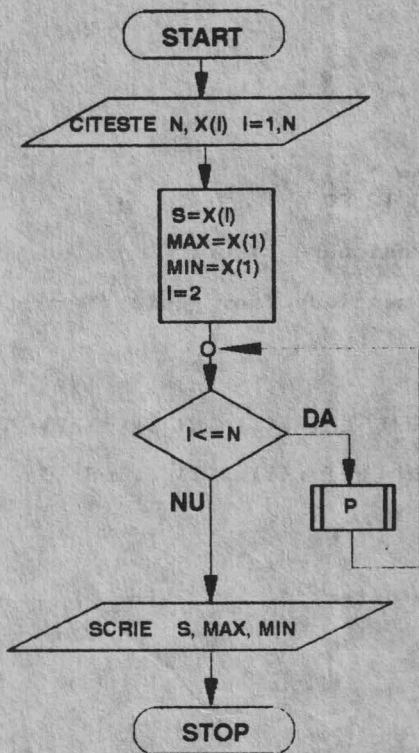
```

procedura P este:

```

 $s \leftarrow s + x_i$ 
dacă not ( $\max \geq x_i$ )
  atunci  $\max \leftarrow x_i$ 
sfîrșit-dacă
dacă not ( $\min \leq x_i$ )
  atunci  $\min \leftarrow x_i$ 
sfîrșit-dacă
sfîrșit-procedură

```



```

10 REM
20 REM *** MINIM, MAXIM SI SUMA ELEMENTELOR ***
30 REM
40 CLS
80 INPUT "dimensiunea sirului n= ";n
90 IF n<=0 OR n<>INT n THEN GO TO 80
100
110 DIM x(n): REM dimensionare sir
120
130 PRINT "sirul x de dimensiune ";n;" este:"
140
150 FOR i=1 TO n
160   INPUT "x(";i);"=";x(i)
170   PRINT x(i);" ";
180 NEXT i
190
200 LET s=x(1): LET max=x(1): LET min=x(1)
210
220 FOR i=2 TO n
230   LET s=s+x(i)
240   IF max<x(i) THEN LET max=x(i)
250   IF min>x(i) THEN LET min=x(i)
260 NEXT i
270
280 PRINT "min(x(i))=";min" 1<=i<=";n" ma
x(x(i))=";max" 1<=i<=";n" suma=";s
290 STOP
  
```

sirul x de dimensiune 13 este:

3 8 -7 0 -7 89 -45 0 33 123 -765 0 13

min(x(i))=-765
1<=i<=13

max(x(i))=123
1<=i<=13

suma=-555

5.5 Se calculează mai întâi suma tuturor elementelor șirului. Apoi se înlocuiește fiecare element x_i cu suma din care se scade valoarea lui x_i împărțită la $n-1$.

```
110 INPUT "dimensiunea sirului n= ";n
120 IF n<=0 OR n<>INT n THEN GO TO 110
130
140 DIM x(n)
150
160 LET s=0
170 PRINT "sirul initial x este:"
180
190 FOR i=1 TO n
200   INPUT "x(";i);")=";x(i)
210   PRINT x(i);" ";
220   LET s=s+x(i)
230 NEXT i
240
250 PRINT ""sirul nou format x este:"
260
270 FOR i=1 TO n
280   LET x(i)=(s-x(i))/(n-1)
290   PRINT x(i);" ";
300 NEXT i
310 PRINT
320 STOP
```

sirul initial x este:

2 6 1 4 5 3

sirul nou format x este:

3.8 3 4 3.4 3.2 3.6

sirul initial x este:

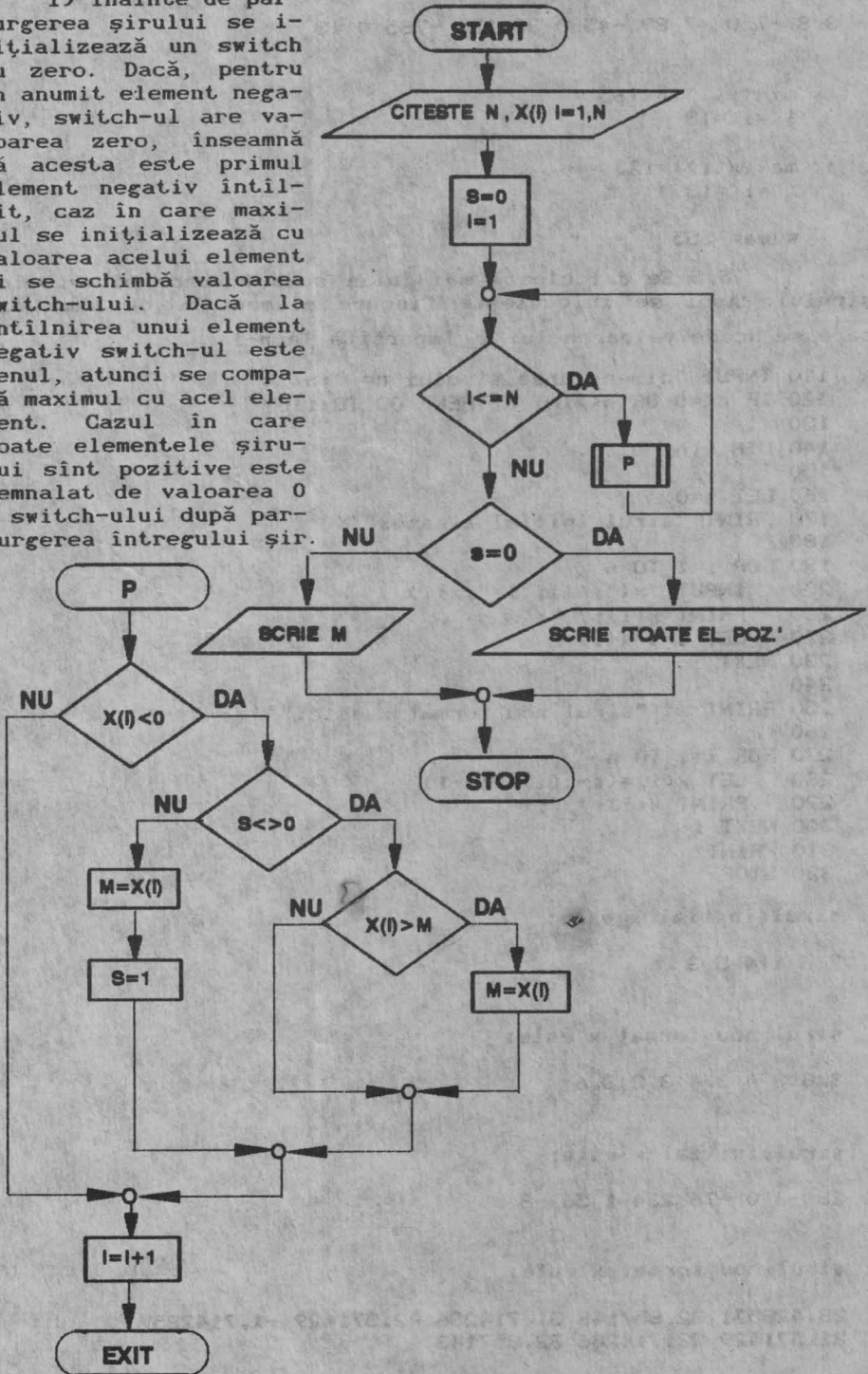
23 -8 0 -76 234 1 56 -8

sirul nou format x este:

28.428571 32.857143 31.714286 42.571429 -1.7142857
31.571429 23.714286 32.857143

5.6 Vom indica două soluții posibile.

1) Înainte de parcurgerea șirului se inițializează un switch cu zero. Dacă, pentru un anumit element negativ, switch-ul are valoarea zero, înseamnă că acesta este primul element negativ întâlnit, caz în care maximum se inițializează cu valoarea acelui element și se schimbă valoarea switch-ului. Dacă la întâlnirea unui element negativ switch-ul este nenul, atunci se compară maximum cu acel element. Cazul în care toate elementele șirului sînt pozitive este semnalat de valoarea 0 a switch-ului după parcurgerea întregului șir.




```

10 REM
20 REM *** MAXIMUL ELEMENTELOR NEGATIVE ***
30 REM
40 CLS
50 PRINT "*****"
60 PRINT "* MAXIMUL ELEMENTELOR NEGATIVE *"
70 PRINT "*****"
80 INPUT "dimensiunea sirului n= ";n
90 IF n<=0 OR n<>INT n THEN GO TO 80
100
110 DIM x(n); LET s=0
120
130 PRINT ""sirul initial x este:""
140
150 FOR i=1 TO n
160 INPUT "x(";i);")=";x(i)
170 PRINT x(i);" ";
180 IF x(i)>=0 THEN GO TO 220
190 IF s=0 THEN LET m=x(i); LET s=1; GO TO 220
200 IF x(i)<=m THEN GO TO 220
210 LET m=x(i)
220 NEXT i
230
240 IF s=0 THEN PRINT ""TOATE elementele pozitive!";
GO TO 260
250 PRINT "" max{x(i)|x(i)<0}=";m" 1<=i<=";n
260 STOP

```

```

*****
* MAXIMUL ELEMENTELOR NEGATIVE *
*****

```

sirul initial x este:

0 45 -45 -7 9 11 -3 0 55 -3 2 3 13

max{x(i)|x(i)<0}=-3
1<=i<=13

sirul initial x este:

4 6 1 8 3 99 11 24 13 10

TOATE elementele pozitive!

2) Se parcurge şirul până la întâlnirea primului element negativ cu a cărui valoare se inițializează maximul și al cărui indice se păstrează. Se parcurge apoi şirul începând de la acel element, comparându-se toate elementele negative cu maximul. Un test asupra variabilei care păstrează indicele primului element negativ (variabilă inițializată cu zero) indică dacă un asemenea element există în şir.

algoritmul 5.6.b. este:

```
citește n
pentru i=1,n execută
  citește  $x_i$ 
sfîrșit-pentru
j ← 0
pentru i=1,n execută
  cheamă P1
sfîrșit-pentru
dacă j=0 atunci scrie
  'nici un element negativ'
  altfel pentru i=j+1,n execută
    cheamă P2
    sfîrșit-pentru
    scrie m
sfîrșit-dacă
```

stop

```
80 INPUT "dimensiunea sirului n=" ;n
90 IF n<=0 OR n<>INT n THEN GO TO 80
100
110 DIM x(n): LET j=0
120
130 PRINT "" "sirul initial x este:" ""
140
150 FOR i=1 TO n
160 INPUT "x(" ;(i); ")=" ;x(i)
170 PRINT x(i); " ";
180 NEXT i
190
200 REM determin primul negativ apoi iesire fortata
210 FOR i=1 TO n
220 IF x(i)<0 THEN LET j=i: LET m=x(i): LET i=n
230 NEXT i
240
250 IF j=0 THEN GO TO 310: REM nici un negativ
260 FOR i=j+1 TO n
270 IF x(i)<0 THEN IF x(i)>m THEN LET m=x(i)
280 NEXT i
290 PRINT "" " max{x(i);x(i)<0}=" ;m" 1<=i<=n: G
0 TO 320
300
310 PRINT "" "TOATE elementele pozitive!"
320 STOP
```

sirul initial x este:

5 7 0 -8 -7 0 -2 11 -2 -3 13 56 113

max{x(i);x(i)<0}=-2
1<=i<=13

sirul initial x este:

5 8 99 0 1 23 5 0 67 1

TOATE elementele pozitive!

procedura P1 este:

```
dacă  $x_i < 0$ 
  atunci j ← i
  m ←  $x_i$ 
  i ← n
sfîrșit-dacă
sfîrșit-procedură
```

procedura P2 este:

```
dacă  $x_i < 0$ 
  atunci
    dacă  $x_i > m$ 
      atunci m ←  $x_i$ 
    sfîrșit-dacă
  sfîrșit-dacă
sfîrșit-procedură
```

5.7 La formarea unui nou șir al cărui număr de elemente nu se cunoaște se inițializează indicele noului șir cu zero și se mărește acest indice cu 1 înainte de a atribui o valoare următorului element al șirului. La încheierea formării noului șir indicele arată exact numărul său de elemente.

algoritmul 5.7. este:

citește a,b,n

pentru i=1,n execută

citește x_i

sfârșit-pentru

$k \leftarrow 0$

$s \leftarrow 0$

$l \leftarrow 0$

pentru i=1,n execută

cheamă P

sfârșit-pentru

dacă $k=0$ atunci scrie 'nu sint elemente in interval'

altfel $s \leftarrow \frac{s}{k}$

scrie s

sfârșit-dacă

dacă $l=0$ atunci scrie 'toate elementele sint in interval'

altfel

pentru j=1,l execută

scrie y_j

sfârșit-pentru

sfârșit-dacă

stop

procedura P este:

dacă $x_i \leq b$ atunci

dacă $x_i \geq a$ atunci $k \leftarrow k+1$

$s \leftarrow s+x_i$

altfel $l \leftarrow l+1$

$y_l \leftarrow x_i$

sfârșit-dacă

altfel $l \leftarrow l+1$

$y_l \leftarrow x_i$

sfârșit-dacă

sfârșit-procedură

80 INPUT "dimensiunea sirului n = ";n

90 IF n<=0 OR n<>INT n THEN GO TO 80

100

110 INPUT "intervalul [a,b] a=";a;" b=";b

120 IF a>b THEN GO TO 110

130 DIM x(n): DIM y(n): LET k=0: LET s=0: LET l=0

140

150 PRINT "sirul initial x este:"

160

170 FOR i=1 TO n

180 INPUT "x(";i);")=";x(i)

190 PRINT x(i);" ";

200' IF x(i)>b THEN LET l=l+1: LET y(l)=x(i): GO TO 230

210 IF x(i)<a THEN LET l=l+1: LET y(l)=x(i): GO TO 230

220 LET k=k+1: LET s=s+x(i)

230 NEXT i

```

240
250 IF k=0 THEN PRINT ""intervalul [";a;",";b;"] nu con
tine nici un element din sir": GO TO 290
260 LET s=s/k
270 PRINT ""media aritmetica a elementelor din interval
ul [";a;",";b;"] este:"""
280 PRINT " ma=";s
290
300 IF l=0 THEN PRINT ""toate elementele sirului sint in
""intervalul [";a;",";b;"]": GO TO 350
310 PRINT ""sirul nou format este:"""
320 FOR j=1 TO l
330 PRINT y(j);" ";
340 NEXT j
350 STOP

```

sirul initial x este:

5 3 7 9 8 5 6 3 8 9

media aritmetica a elementelor din intervalul [3,9] este:

ma=6.3

toate elementele sirului sint in intervalul [3,9]

sirul initial x este:

0 9 -4 -7 -12 1 2 -55 13 0 11

intervalul [5,8] nu contine nici un element din sir

sirul nou format este:

0 9 -4 -7 -12 1 2 -55 13 0 11

sirul initial x este:

2 7 -9 10 6 -2 1 9 -3 8 -1 0 11

media aritmetica a elementelor din intervalul [6,10] este:

ma=8

sirul nou format este:

2 -9 -2 1 -3 -1 0 11

5.8 Se formează un nou șir în felul următor: șirul inițial se parcurge într-o structură repetitivă de contor i ce variază între 1 și $n-1$ (deoarece în fiecare etapă sint necesare cite două elemente consecutive ale șirului). Pentru fiecare valoare a lui i se introduc în noul șir elementul x_i și media aritmetică a lui x_i și a succesivului său în șir. La încheierea structurii repetitive trebuie introdus în noul șir elementul x_n .

algoritmul 5.8. este:

```

citește n
pentru i=1, n execută
  citește  $x_i$ 
sfârșit-pentru
j ← 0
pentru i=1, n-1 execută
  j ← j+2
   $y_{j-1}$  ←  $x_i$ 
   $y_j$  ←  $\frac{x_i + x_{i+1}}{2}$ 
sfârșit-pentru
j ← j+1
 $y_j$  ←  $x_n$ 
pentru k=1, j execută
  scrie  $y_k$ 
sfârșit-pentru
stop

```

```

10 REM
20 REM *** FORMARE SIR PRIN INSERARE ***
30 REM
40 CLS
100 INPUT "dimensiunea sirului n=" ;n
110 IF n<=0 OR n<>INT n THEN GO TO 100
120
130 DIM x(n): DIM y(2*n-1): LET j=0
140
150 PRINT "sirul initial x este:"
160
170 FOR i=1 TO n
180   INPUT "x(";i);")=" ;x(i)
190   PRINT x(i);" ";
200 NEXT i
210
220 FOR i=1 TO n-1
230   LET j=j+2
240   LET y(j-1)=x(i)
250   LET y(j)=(x(i)+x(i+1))/2
260 NEXT i
270 LET j=j+1
280 LET y(j)=x(n)
290
300 PRINT ""sirul nou format y este:"
310 FOR k=1 TO j
320   PRINT y(k);" ";
330 NEXT k
340 PRINT
350 STOP

```

sirul initial x este:

4 8 -9 0 -4 11 -5 1 13

sirul nou format y este:

4 6 8 -0.5 -9 -4.5 0 -2 -4 3.5 11 3 -5 -2 1 7 13

5.9

```

120 INPUT "dimensiunea sirului n= ";n
130 IF n<=0 OR n<>INT n THEN GO TO 120
140 INPUT "intervalul a=";a;" b=";b
150 IF a>b OR a*b<=0 THEN GO TO 140
160
170 DIM x(n): LET k=0: LET j=0: LET mg=1: LET sp=0: LET l=0:
LET pn=1
180
190 PRINT "sirul initial x este:"
200
210 FOR i=1 TO n
220 INPUT "x(";i);")=";x(i)
230 PRINT x(i);" ";
240 IF x(i)>=a AND x(i)<=b THEN LET mg=mg*x(i): LET k=k+1
250 IF x(i)>=0 THEN LET j=j+1: LET sp=sp+x(i): GO TO 270
260 LET l=l+1: LET pn=pn*x(i)
270 NEXT i
280
290 PRINT "intervalul considerat [";a;" ";b;"]";
300 IF k<>0 THEN LET mg=mg^(1/k): PRINT "media geometrica
mg=";mg: GO TO 320
310 PRINT "nici un element nu apartine"intervalului [";a;
";b;"]": GO TO 320
320 PRINT "in sir sint ";j;" elemente pozitive"
330 IF j<>0 THEN PRINT "suma lor SP=";sp
340 PRINT "in sir sint ";l;" elemente negative"
350 IF l<>0 THEN PRINT "produsul lor PN=";pn
360 STOP

```

sirul initial x este:

6 0 -1 3 -7 8 10 -1 -2 12 4 7 13

intervalul considerat [1,7]

media geometrica mg=4.7381372

in sir sint 9 elemente pozitive
suma lor SP=63

in sir sint 4 elemente negative
produsul lor PN=14

sirul initial x este:

1 9 0 8 22 13 56 21 98 123 33 77 13

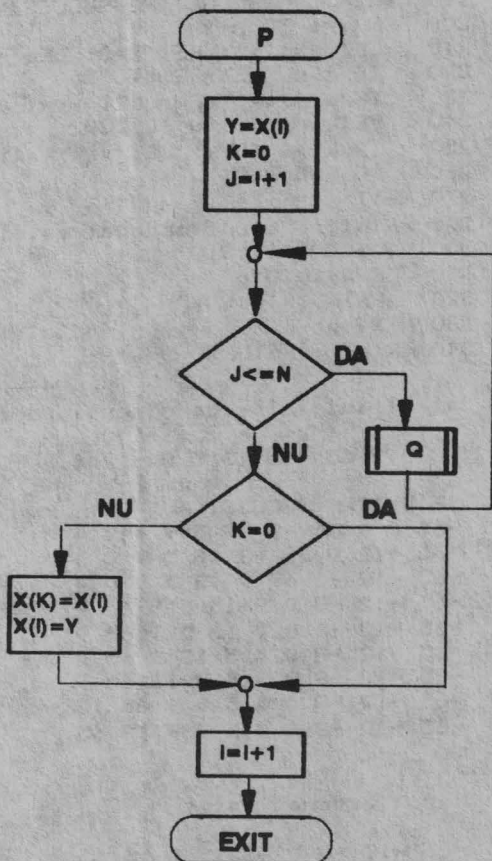
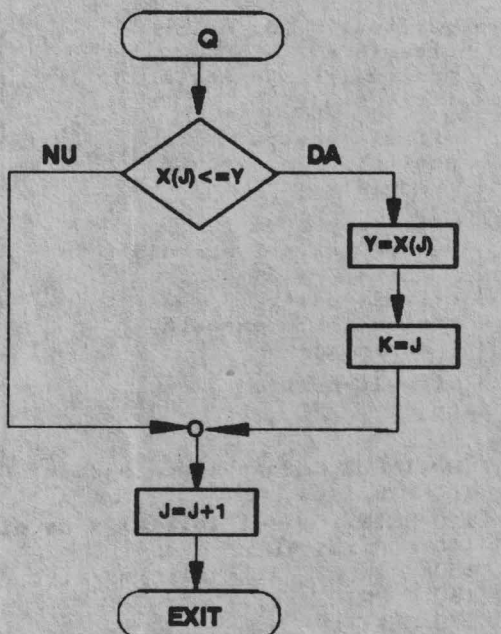
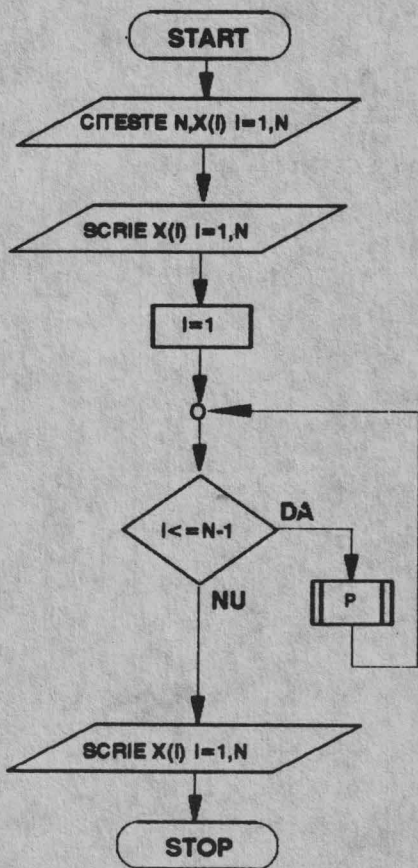
intervalul considerat [5,6]

nici un element nu apartine
intervalului [5,6]

in sir sint 13 elemente pozitive
suma lor SP=474

in sir sint 0 elemente negative

5.10 Se caută cel mai mic element din șir, se reține indicele său și se inversează acel element cu primul element al șirului. Apoi se caută cel mai mic element începând de la al doilea, acesta se inversează cu al doilea element etc.



```

algoritmul 5.10. este:
  citește n
  pentru i=1,n execută
    citește xi
  sfârșit-pentru
  pentru i=1,n execută
    scrie xi
  sfârșit-pentru
  pentru i=1,n-1 execută
    cheamă P
  sfârșit-pentru
  pentru i=1,n execută
    scrie xi
  sfârșit-pentru
stop

```

```

procedura P este:
  y ← xi
  k ← 0
  pentru j=i+1,n execută
    cheamă Q
  sfârșit-pentru
  dacă k≠0 atunci xk ← xi
                  xi ← y
  sfârșit-dacă
sfârșit-procedură

```

```

procedura Q este:
  dacă xj ≤ y atunci y ← xj
                  k ← j
  sfârșit-dacă
sfârșit-procedură

```

```

80 INPUT "dimensiunea sirului n = ";n: LET n=ABS (INT n)
100 DIM x(n)
120 PRINT "sirul initial x de dimensiune ";n;" este:"
140 FOR i=1 TO n
150   INPUT "x(";i);")=";x(i)
160   PRINT x(i);" ";
170 NEXT i
190 PRINT ""SE OBTINE SUCCESIV"
200 FOR i=1 TO n-1
210   LET y=x(i): LET k=0: LET j=i+1
220   IF j>n THEN GO TO 250
230   IF x(j)<y THEN LET y=x(j): LET k=j
240   LET j=j+1: GO TO 220
250   IF k<>0 THEN LET x(k)=x(i): LET x(i)=y
260   GO SUB 310
270 NEXT i
290 PRINT ""sirul ordonat este:"
300 GO SUB 310: STOP
310 .OR p=1 TO n
320   PRINT x(p);" ";
330 NEXT p
340 PRINT : RETURN

```

sirul initial x de dimensiune 10 este:

-8 0 9 65 -65 13 5 9 -12 4

SE OBTINE SUCCESIV

-65 0 9 65 -8 13 5 9 -12 4

-65 -12 9 65 -8 13 5 9 0 4

-65 -12 -8 65 9 13 5 9 0 4

-65 -12 -8 0 9 13 5 9 65 4

-65 -12 -8 0 4 13 5 9 65 9

-65 -12 -8 0 4 5 13 9 65 9

-65 -12 -8 0 4 5 9 13 65 9

-65 -12 -8 0 4 5 9 9 65 13

-65 -12 -8 0 4 5 9 9 13 65

sirul ordonat este:

-65 -12 -8 0 4 5 9 9 13 65

5.11 După calcularea maximului elementelor șirului, acesta se poate insera înaintea tuturor elementelor, între x_1 și x_2 , între x_2 și x_3 etc. sau după toate elementele șirului. Deoarece problema cere să nu se formeze un nou șir, se poate proceda astfel: se scrie maximul urmat de elementele șirului, apoi pentru fiecare k între 1 și n , se scriu primele k elemente ale șirului, apoi maximul și în continuare elementele începând de la al $k+1$ -lea element.

algoritmul 5.11. este:

```

citește n
[ pentru i=1,n execută
  citește  $x_i$ 
  sfârșit-pentru
  max ←  $x_1$ 
  pentru i=2,n execută
    cheamă P
  sfârșit-pentru
  scrie max
  pentru i=1,n execută
    scrie  $x_i$ 
  sfârșit-pentru
  pentru k=1,n execută
    cheamă Q
  sfârșit-pentru
stop

```

procedura P este:

```

dacă not (max >  $x_i$ )
  atunci max ←  $x_i$ 
sfârșit-dacă
sfârșit-procedură

```

procedura Q este:

```

pentru i=1,k execută
  scrie  $x_i$ 
sfârșit-pentru
scrie max
  pentru i=k+1,n execută
    scrie  $x_i$ 
  sfârșit-pentru
sfârșit-procedură

```

```

80 INPUT "dimensiunea sirului n = ";n: LET n=ABS (IN" n)
90 IF n=0 THEN GO TO 80
100
110 DIM x(n)
120
130 PRINT "sirul initial x este:"
140 FOR i=1 TO n
150 INPUT "x(";i);" )=";x(i)
160 PRINT x(i);" ";
170 NEXT i
180
190 LET max=x(1)
200 FOR i=2 TO n
210 IF x(i)>max THEN LET max=x(i)
220 NEXT i
230
240 PRINT ""prin inserarea elementului ";max
250 PRINT "se obtin sirurile:"
260 PRINT INVERSE 1;max; INVERSE 0;" ";
270 FOR i=1 TO n
280 PRINT x(i);" ";
290 NEXT i: PRINT
300 FOR k=1 TO n
310 FOR i=1 TO k
320 PRINT x(i);" ";
330 NEXT i
340 PRINT INVERSE 1;max; INVERSE 0;" ";
350 FOR i=k+1 TO n
360 PRINT x(i);" ";
370 NEXT i
380 PRINT
390 NEXT k
400 STOP

```

sirul initial x este:

3 7 9 -7 10 -4 1 13 -5 3

prin inserarea elementului 13
se obtin sirurile:

13 3 7 9 -7 10 -4 1 13 -5 3
3 13 7 9 -7 10 -4 1 13 -5 3
3 7 13 9 -7 10 -4 1 13 -5 3
3 7 9 13 -7 10 -4 1 13 -5 3
3 7 9 -7 13 10 -4 1 13 -5 3
3 7 9 -7 10 13 -4 1 13 -5 3
3 7 9 -7 10 -4 13 1 13 -5 3
3 7 9 -7 10 -4 1 13 13 -5 3
3 7 9 -7 10 -4 1 13 13 -5 3
3 7 9 -7 10 -4 1 13 -5 13 3
3 7 9 -7 10 -4 1 13 -5 3 13

5.12 a) Se folosesc doi indici. Indicele pentru elementele nenule, se inițializează cu zero și se mărește cu 1 ori de câte ori se întâlnește un element nenul (cea ce corespunde punerii elementelor nenule la începutul noului șir în ordinea găsirii lor). Celălalt indice se folosește pentru elementele nule, este inițializat cu $n+1$ și se micșorează cu 1 la întâlnirea fiecărui element nul (cea ce corespunde cu punerea zerourilor la sfârșitul noului șir, ceea ce nu este necesar în cazul folosirii limbajului BASIC deoarece prin dimensionarea unui șir numeric elementele acestuia sînt inițializate cu zero).

b) Indicele pentru elementele nenule are aceeași variație ca și în soluția dată la punctul a). Al doilea indice nu există, elementele nule se sar iar șirul se completează cu 0 pînă la dimensiunea inițială.

algoritmul 5.12.a. este:

```
citește n
pentru i=1,n execută
  citește  $x_i$ 
sfîrșit-pentru
pentru i=1,n execută
  scrie  $x_i$ 
sfîrșit-pentru
j ← 0
k ← n+1
pentru i=1,n execută
  dacă  $x_i=0$  atunci k ← k-1
   $y_k ← x_i$ 
  altfel j ← j+1
   $y_j ← x_i$ 
sfîrșit-dacă
sfîrșit-pentru
pentru i=1,n execută
  scrie  $y_i$ 
sfîrșit-pentru
stop
```

algoritmul 5.12.b. este:

```
citește n
pentru i=1,n execută
  citește  $x_i$ 
sfîrșit-pentru
pentru i=1,n execută
  scrie  $x_i$ 
sfîrșit-pentru
j ← 0
pentru i=1,n execută
  dacă  $x_i \neq 0$  atunci j ← j+1
   $x_j ← x_i$ 
sfîrșit-dacă
sfîrșit-pentru
pentru i=j+1,n execută
   $x_i ← 0$ 
sfîrșit-pentru
pentru i=1,n execută
  scrie  $x_i$ 
sfîrșit-pentru
stop
```

VARIANTA a

```

10 REM
20 REM *** MUTARE ZEROURI LA SFIRSIT ***
30 REM
80 INPUT "dimensiunea sirului n= ";n: LET n=ABS (INT n)
90 IF n=0 THEN GO TO 80
100
110 DIM x(n): DIM y(n)
120
130 PRINT "sirul initial x este:"
140 FOR i=1 TO n
150   INPUT "x(";i);")=";x(i)
160   PRINT x(i);" ";
170 NEXT i
180
190 LET j=0: LET k=n+1
200 FOR i=1 TO n
210   IF x(i)=0 THEN LET k=k-1: LET y(k)=x(i): GO TO 230
220   LET j=j+1: LET y(j)=x(i)
230 NEXT i
240
250 PRINT ""sirul obtinut este:"
260 FOR i=1 TO n
270   PRINT y(i);" ";
280 NEXT i: PRINT
290 STOP

```

sirul initial x este:

4 8 0 -8 11 0 0 -13 5 -78 0 -7 13

sirul obtinut este:

4 8 -8 11 -13 5 -78 -7 13 0 0 0 0

VARIANTA b

```

80 INPUT "dimensiune sir n=";n: LET n=ABS (INT n)
90 IF n=0 THEN GO TO 80
100
110 DIM x(n)
120
130 PRINT "sirul initial x este:"
140 FOR i=1 TO n
150   INPUT "x(";i);")=";x(i)
160   PRINT x(i);" ";
170 NEXT i
180
190 LET j=0
200 FOR i=1 TO n
210   IF x(i)<>0 THEN LET j=j+1: LET x(j)=x(i)
220 NEXT i
230
240 FOR i=j+1 TO n
250   LET x(i)=0
260 NEXT i
270 PRINT ""sirul obtinut este:"
280 FOR i=1 TO n
290   PRINT x(i);" ";
300 NEXT i: PRINT
310 STOP

```

sirul initial x este:

0 11 3 0 0 9 8 13 8 0 8 7 0

sirul obtinut este:

11 3 9 8 13 8 8 7 0 0 0 0 0

5.13

```
10 REM
20 REM *** ELIMINARE ZEROURI-INSERARE ELEMENT ***
30 REM
40 CLS
90 INPUT "dimensiune sir n=";n: LET n=ABS (INT n)
100 IF n=0 THEN GO TO 90
110
120 DIM x(n): DIM y(n): DIM z(2*n-1)
130
140 PRINT "sirul initial x este:"
150 FOR i=1 TO n
160   INPUT "x(";i);")=";x(i)
170   PRINT x(i);" ";
180 NEXT i
190
200 LET j=0
210 FOR i=1 TO n
220   IF x(i)<>0 THEN LET j=j+1: LET y(j)=x(i)
230 NEXT i
240
250 IF j=0 THEN PRINT "toate elementele nule": GO TO 400
260
270 LET k=0
280 FOR i=1 TO j-1
290   LET p=y(i)*y(i+1)
300   IF p<0 THEN LET k=k+1: LET z(k)=y(i): GO TO 340
310   LET z(k+1)=y(i)
320   LET z(k+2)=p
330   LET k=k+2
340 NEXT i
350 LET z(k+1)=y(j)
360 PRINT "sirul obtinut este:"
370 FOR i=1 TO k+1
380   PRINT z(i);" ";
390 NEXT i: PRINT
400 STOP
```

sirul initial x este:

3 6 0 8 0 1 -8 0 -6 0

sirul obtinut este:

3 18 6 48 8 8 1 -8 48 -6

algoritmul 5.14. este:

```

citește n
pentru i=1,n execută
  citește  $x_i$ 
sfârșit-pentru
pentru i=1,n execută
  scrie  $x_i$ 
sfârșit-pentru
j ← 0
k ← 0
pentru i=1,n execută
  dacă  $x_i \neq 0$  atunci j ← j+1
     $x_j \leftarrow x_i$ 
    dacă j ≥ 2
      atunci p ←  $x_{j-1} \cdot x_j$ 
        dacă p ≥ 0 atunci
           $z_{k+1} \leftarrow x_{j-1}$ 
           $z_{k+2} \leftarrow p$ 
          k ← k+2
        altfel k ← k+1
           $z_k \leftarrow x_{j-1}$ 
        sfârșit-dacă
      sfârșit-dacă
    sfârșit-dacă
  sfârșit-pentru
  dacă j=0 atunci scrie 'toate elementele nule'
  altfel  $z_{k+1} \leftarrow x_j$ 
    pentru i=1,k+1 execută
      scrie  $x_i$ 
    sfârșit-pentru
  sfârșit-dacă
stop

```

```

10 REM
20 REM *** ELIMINARE ZEROURI-INSERARE ELEMENT ***
30 REM
40 CLS
50 PRINT "*****"
60 PRINT "* ELIMINARE ZEROURI *"
70 PRINT "* INSERARE ELEMENT *"
80 PRINT "*****"
90 INPUT "dimensiune sir n=";n: LET n=ABS (INT n)
100 IF n=0 THEN GO TO 90
110
120 DIM x(n): DIM z(2*n-1)
130
140 PRINT "sirul initial x este:"
150 FOR i=1 TO n
160 INPUT "x(";(i);")=";x(i)
170 PRINT x(i);" ";
180 NEXT i
190

```

```

200 LET j=0: LET k=0
210 FOR i=1 TO n
220 IF x(i)=0 THEN GO TO 300
230 LET j=j+1: LET x(j)=x(i)
240 IF j<2 THEN GO TO 300
250 LET p=x(j-1)*x(j)
260 IF p<0 THEN LET k=k+1: LET z(k)=x(j-1): GO TO 300
270 LET z(k+1)=x(j-1)
280 LET z(k+2)=p
290 LET k=k+2
300 NEXT i
310
320 IF j=0 THEN PRINT "toate elementele nule": GO TO 390
330
340 LET z(k+1)=x(j)
350 PRINT "sirul obtinut este:"
360 FOR i=1 TO k+1
370 PRINT z(i); " ";
380 NEXT i: PRINT
390 STOP

```

```

*****
* ELIMINARE ZEROURI *
* INSERARE ELEMENT *
*****

```

sirul initial x este:

-7 -8 0 -1 9 0 0 4 0 10

sirul obtinut este:

-7 56 -8 8 -1 9 36 4 40 10

6.1 Se inițializează trei contori pentru numărarea elementelor pozitive, negative și nule (NP, NN respectiv NZ) după care se parcurge matricea, actualizind pentru fiecare element contorul corespunzător

```

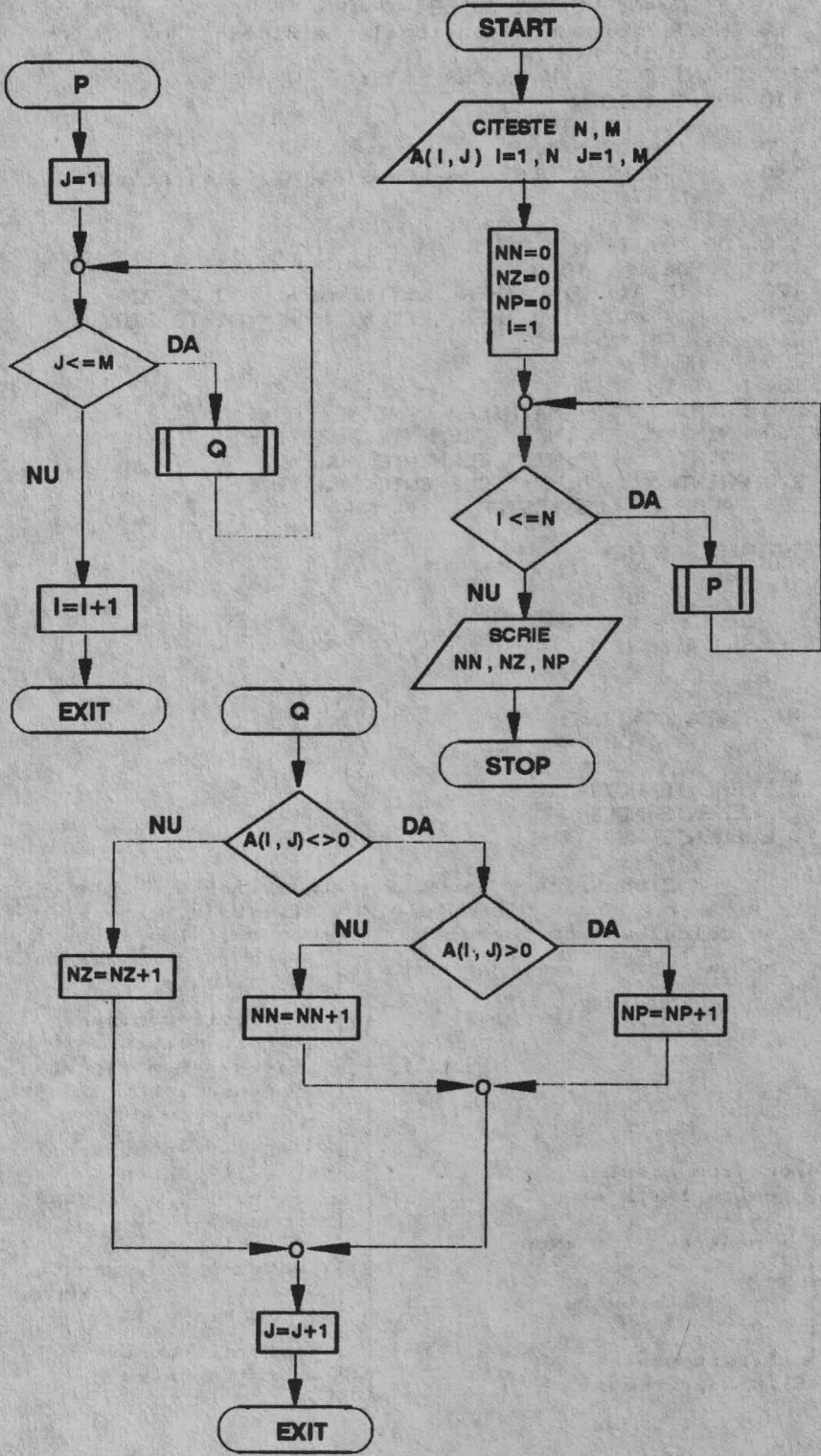
[ algoritmul 6.1 este:
  citește n,m
  pentru i=1,n execută
  [ pentru j=1,m execută
    citește ai,j
  ] sfârșit-pentru
  sfârșit-pentru
  nn ← 0
  np ← 0
  nz ← 0
  pentru i=1,n execută
  [ cheamă P
  ] sfârșit-pentru
  scrie nn,np,nz
] stop

```

```

[ procedura P este:
  pentru j=1,m execută
  [ cheamă Q
  ] sfârșit-pentru
  sfârșit-procedură
]
[ procedura Q este:
  dacă ai,j ≠ 0
  atunci [ dacă ai,j > 0
    atunci np ← np+1
    altfel nn ← nn+1
  ] sfârșit-dacă
  altfel nz ← nz+1
  sfârșit-dacă
] sfârșit-procedură

```



```

70 LET NN=0: LET NZ=0: LET NP=0
80 INPUT "Introdu dimensiunile matricei "N=";N,"M=";M
90 CLS : DIM A(N,M)
100 PRINT "" MATRICEA ESTE: "
110 FOR I=1 TO N
120   FOR J=1 TO M
130     INPUT A(I,J)
140     PRINT AT I+3,J*4-LEN STR$ A(I,J)+4;A(I,J)
150   NEXT J
160 NEXT I
170 FOR I=1 TO N
180   FOR J=1 TO M
190     IF A(I,J)>0 THEN LET NP=NP+1: GO TO 220
200     IF A(I,J)=0 THEN LET NZ=NZ+1: GO TO 220
210     LET NN=NN+1
220   NEXT J
230 NEXT I
240 PRINT "" MATRICEA CONTINE: ""
250 PRINT " ";NP;" ELEMENTE POZITIVE"
260 PRINT " ";NZ;" ELEMENTE NULE"
270 PRINT " ";NN;" ELEMENTE NEGATIVE"
280 PAUSE 0: CLS : STOP

```

MATRICEA ESTE:

1	-2	0	9
-6	12	4	0
-56	4	3	6

MATRICEA CONTINE:

7 ELEMENTE POZITIVE
2 ELEMENTE NULE
3 ELEMENTE NEGATIVE

6.2 Matricea C are n linii și k coloane. Elementele sale se calculează cu formula:

$$c_{i,l} = \sum_{j=1}^m a_{i,j} \cdot b_{j,l} \quad i=1,n; \\ l=1,k;$$

procedura P este:

- pentru $l=1,k$ execută
 - $s \leftarrow 0$
 - pentru $j=1,m$ execută
 - $s \leftarrow s + a_{i,j} \cdot b_{j,l}$
 - sfârșit-pentru
 - $c_{i,l} \leftarrow s$
- sfârșit-pentru
- sfârșit-procedură

algoritmul 6.2 este:

- citește n,m,k
- pentru $i=1,n$ execută
 - pentru $j=1,m$ execută
 - citește $a_{i,j}$
 - sfârșit-pentru
 - sfârșit-pentru
 - pentru $j=1,m$ execută
 - pentru $l=1,k$ execută
 - citește $b_{j,l}$
 - sfârșit-pentru
 - sfârșit-pentru
 - pentru $i=1,n$ execută
 - cheamă P
 - sfârșit-pentru
 - pentru $j=1,n$ execută
 - pentru $l=1,k$ execută
 - scrie $c_{j,l}$
 - sfârșit-pentru
 - sfârșit-pentru
 - stop


```

10 PRINT " *****"
20 PRINT " * PRODUSUL A 2 TABLOURI *"
30 PRINT " * BIDIMENSIONALE *"
40 PRINT " *****"
50 INPUT "Introdu dimensiunile matricelor"/"N=";N;"
M=";M;" K=";K
60 DIM A(N,M): DIM B(M,K): DIM C(N,K): CLS
70 PRINT "" TABLOURILE SINT:"
80 REM *** prima matrice ***
90 FOR I=1 TO N
100 FOR J=1 TO M
110 INPUT A(I,J)
120 PRINT AT I+3,J*4-LEN STR$ A(I,J);A(I,J)
130 NEXT J
140 NEXT I
150 REM *** a doua matrice ***
160 FOR I=1 TO M
170 FOR J=1 TO K
180 INPUT B(I,J)
190 PRINT AT I+N+4,J*4-LEN STR$ B(I,J);B(I,J)
200 NEXT J
210 NEXT I
220 REM *** calcul produs ***
230 FOR I=1 TO N
240 FOR L=1 TO K
250 LET S=0
260 FOR J=1 TO M
270 LET S=S+A(I,J)*B(J,L)
280 NEXT J
290 LET C(I,L)=S
300 NEXT L
310 NEXT I
320 CLS : PRINT "" PRODUSUL TABLOURILOR ESTE:"
330 FOR I=1 TO N
340 FOR J=1 TO K
350 PRINT AT I+3,J*4-LEN STR$ C(I,J)+3;C(I,J)
360 NEXT J
370 NEXT I
380 PAUSE 0: CLS : STOP

```

```

*****
* PRODUSUL A 2 TABLOURI *
* BIDIMENSIONALE *
*****

```

TABLOURILE SINT:

```

1  0  -2  -1
0  2  -2   1
3  0  -1   3

2  0
-3  1
4  0
-2  3

```

PRODUSUL TABLOURILOR ESTE:

```

-4  -3
-16 5
-4  9

```

6.3 Se parcurge matricea pe coloane. Pentru fiecare coloană se calculează atât suma elementelor sale cât și maximum lor, indicele de linie al maximumului reținându-se. Când s-a terminat parcurgerea unei coloane, elementul maxim se înlocuiește cu suma calculată, după care se trece la următoarea coloană.

```

algoritmul 6.3. este:
  citește n,m
  pentru i=1,n execută
    pentru j=1,m execută
      citește  $a_{i,j}$ 
    sfîrșit-pentru
  sfîrșit-pentru
  cheamă SCRIE
  cheamă P
  cheamă SCRIE
stop

```

```

procedura SCRIE este:
  pentru i=1,n execută
    pentru j=1,m execută
      scrie  $a_{i,j}$ 
    sfîrșit-pentru
  sfîrșit-pentru
sfîrșit-procedură

```

```

procedura P este:
  pentru j=1,m execută
    maxj ←  $a_{1,j}$ 
    ii ← 1
    sj ←  $a_{1,j}$ 
    pentru i=2,n execută
      sj ← sj+ $a_{i,j}$ 
      dacă  $a_{i,j} > \text{maxj}$ 
        atunci maxj ←  $a_{i,j}$ 
        ii ← j
    sfîrșit-dacă
    sfîrșit-pentru
     $a_{ii,j} \leftarrow \text{sj}$ 
  sfîrșit-pentru
sfîrșit-procedură

```

```

100 INPUT "linii n=";n;" coloane=";m: LET n=ABS (INT n):
LET m=ABS (INT m)
110 IF n=0 OR m=0 THEN GO TO 100
120 DIM a(n,m): LET lin=0: LET len=4: REM len=lungimea pe
care se scrie un element din matrice
130 PRINT "matricea A:"
140 FOR i=1 TO n
150 FOR j=1 TO m
160 INPUT "a(";(i);","; (j);")=";a(i,j)
170 GO SUB 400: PRINT a(i,j);
180 NEXT j
190 PRINT : LET lin=lin+1
200 NEXT i
210
220 FOR j=1 TO m
230 LET maxj=a(1,j): LET ii=1: LET sj=a(1,j)
240 FOR i=2 TO n
250 LET sj=sj+a(i,j)
260 IF a(i,j)>maxj THEN LET maxj=a(i,j): LET ii=i
270 NEXT i
280 LET a(ii,j)=sj
290 NEXT j
300
310 PRINT #0;AT 0,8;"- apasa o tasta -": PAUSE 0
320 PRINT "matricea A dupa inlocuire:"
330 FOR i=1 TO n
340 FOR j=1 TO m
350 GO SUB 400: PRINT a(i,j);
360 NEXT j
370 PRINT
380 NEXT i
390 STOP
400 FOR u=1 TO len-LEN (STR$ a(i,j)): PRINT " ";: NEXT u:
RETURN

```

matricea A:

5	4	0	-8	7
-76	8	9	12	13
-8	0	4	5	99

matricea A dupa inlocuire:

-79	4	0	-8	7
-76	12	13	9	13
-8	0	4	5	119

6.4 Se va folosi o structură repetitivă de pas 2 după indicele de linie i ce variază între 1 și $n-1$. Pentru fiecare valoare a indicelui i se vor inversa element cu element liniile i și $i+1$ (folosind o altă structură repetitivă, de data aceasta după indicele de coloană).

```
80 INPUT "linii n=";n;" coloane=";m: LET n=ABS (INT n):
LET m=ABS (INT m)
90 IF n=0 OR m=0 THEN GO TO 80
100 DIM a(n,m): LET len=4: REM len=lungimea pe care se scr
ie un element din matrice
110 PRINT "matricea A:"
120 FOR i=1 TO n
130   FOR j=1 TO m
140     INPUT "a(";i);", ";(j);")=";a(i,j)
150     GO SUB 350: PRINT a(i,j);
160   NEXT j
170   PRINT
180 NEXT i
190
200 FOR i=1 TO n-1 STEP 2
210   FOR j=1 TO m
220     LET y=a(i,j): LET a(i,j)=a(i+1,j): LET a(i+1,j)=y
230   NEXT j
240 NEXT i
250
260 PRINT #0;AT 0,8;"- apasa o tasta -": PAUSE 0
270 PRINT "matricea A dupa transformare:"
280 FOR i=1 TO n
290   FOR j=1 TO m
300     GO SUB 350: PRINT a(i,j);
310   NEXT j
320   PRINT
330 NEXT i
340 STOP
350 FOR u=1 TO len-LEN (STR$ a(i,j)): PRINT " ";: NEXT u:
RETURN
```

matricea A:

1	4	55	7	0
-89	7	0	-45	12
5	6	8	12	-98
1	2	3	4	5

matricea A dupa transformare:

-89	7	0	-45	12
1	4	55	7	0
1	2	3	4	5
5	6	8	12	-98

6.5 Se pot aduna concomitent la suma S (inițializată cu valoarea zero) elementele de același indice de coloană de pe prima și respectiv ultima linie a matricii și elementele cu același indice de linie de pe prima și respectiv ultima coloană, avînd grijă să nu adăugăm de două ori elementele a_{11} , a_{1m} , a_{n1} și a_{nm} , (n și m fiind numărul de linii, respectiv de coloane al matricii).

```

10 REM
20 REM *** SUMA ELEMENTELOR MARGINALE ***
30 REM
40 PAPER 0: INK 5: BORDER 0: BRIGHT 1: CLS
50 PRINT "*****"
60 PRINT "* SUMA ELEMENTELOR MARGINALE *"
70 PRINT "*****"
80 INPUT "linii n=";n;" coloane=";m: LET n=ABS (INT n):
LET m=ABS (INT m)
90 IF n=0 OR m=0 THEN GO TO 80
100 DIM a(n,m): LET len=4: REM len=lungimea pe care se scr
ie un element din matrice
110 PRINT "matricea A:"
120 FOR i=1 TO n
130 FOR j=1 TO m
140 INPUT "a(";(i);";";(j);")=";a(i,j)
150 GO SUB 310: PRINT a(i,j);
160 NEXT j
170 PRINT
180 NEXT i
190
200 LET s=0
210 FOR j=1 TO m
220 LET s=s+a(1,j)+a(n,j)
230 NEXT j
240 FOR i=2 TO n-1
250 LET s=s+a(i,1)+a(i,m)
260 NEXT i
270
280 PRINT "suma elementelor marginale:"
290 PRINT " s=";s
300 STOP
310 FOR u=1 TO len-LEN (STR$ a(i,j)): PRINT " "; NEXT u
320 RETURN

```

```

*****
* SUMA ELEMENTELOR MARGINALE *
*****

```

matricea A:

2	5	0	9
56	-8	0	-34
12	34	0	98
-56	-9	0	-8
6	5	1	4
0	9	8	7

suma elementelor marginale:

s=118

6.6

```

10 PRINT " *****"
20 PRINT " * GENERAREA UNUI SIR *"
30 PRINT " * CU ELEMENTELE *"
40 PRINT " * UNEI MATRICI *"
50 PRINT " *****"
60 PRINT "'SE PASTREAZA IN SIR ELEMENTELE"
70 PRINT " UNEI MATRICI PARCURSA:"
80 PRINT "' 1.PE LINIE'" 2.PE COLOANA"
90 PAUSE 0: IF INKEY$="1" THEN LET SW=1: GO TO 120
100 IF INKEY$="2" THEN LET SW=2: GO TO 120
110 GO TO 90
120 INPUT "Introdu dimensiunile matricei "'N=";N,"M=";M
130 CLS : DIM A(N,M): DIM X(N*M)
140 PRINT "' MATRICEA INITIALA ESTE:"
150 FOR I=1 TO N
160   FOR J=1 TO M
170     INPUT A(I,J)
180     PRINT AT I+4,J*4-LEN STR$ A(I,J)+4;A(I,J)
190   NEXT J
200 NEXT I
210 IF SW=2 THEN GO TO 300
220 REM *** parcurgere pe linie ***
230 LET L=0
240 FOR I=1 TO N
250   FOR J=1 TO M
260     LET L=L+1: LET X(L)=A(I,J)
270   NEXT J
280 NEXT I
290 GO TO 380
300 REM *** parcurgere pe coloana ***
310 LET L=0
320 FOR J=1 TO M
330   FOR I=1 TO N
340     LET L=L+1: LET X(L)=A(I,J)
350   NEXT I
360 NEXT J
370 REM *** afisarea sirului ***
380 CLS : PRINT "' SIRUL GENERAT ESTE:"
390 LET K=5: LET J=-4
400 FOR I=1 TO L
410   LET J=J+4
420   PRINT AT K,J-LEN STR$ X(I)+4;X(I)
430   IF I/8=INT (I/8) THEN PRINT : LET K=K+1: LET J=-4
440 NEXT I
450 PAUSE 0: CLS : GO TO 60

```

```

*****
* GENERAREA UNUI SIR *
* CU ELEMENTELE *
* UNEI MATRICI *
*****

```

SE PASTREAZA IN SIR ELEMENTELE
UNEI MATRICI PARCURSA:

1.PE LINIE

2.PE COLOANA

MATRICEA INITIALA ESTE:

1	2	3	4
5	6	7	8
9	10	11	12

SIRUL GENERAT ESTE:

1	2	3	4	5	6	7	8
9	10	11	12				

SE PASTREAZA IN SIR ELEMENTELE
UNEI MATRICI PARCURSA:

1. PE LINIE

2. PE COLOANA

MATRICEA INITIALA ESTE:

1	2	3	4
5	6	7	8
9	10	11	12

SIRUL GENERAT ESTE:

1	5	9	2	6	10	3	7
11	4	8	12				

6.7 Algoritmul va conține o structură repetitivă de pas 2 după indicele de coloană j ce variază între 1 și $m-1$ (m fiind numărul de coloane al matricii). In fiecare etapă se va calcula maximul elementelor de pe două coloane alăturate. Dacă m este impar, la ieșirea din FOR indicele va fi egal cu m , caz in care trebuie calculat maximul elementelor ultimei coloane. Dacă m este par, la ieșirea din FOR indicele de coloană va fi mai mare decit m , ceea ce indică incheierea algoritmului.

algoritmul 6.7. este:

```
citește n,m
pentru i=1,n execută
  pentru j=1,m execută
    citește ai,j
  sfîrșit-pentru
sfîrșit-pentru
l ← 0
pentru j=1,m-1 execută
  cheamă P
sfîrșit-pentru
dacă j=m atunci cheamă Q
sfîrșit-dacă
  pentru i=1,l execută
    scrie xi
  sfîrșit-pentru
stop
```

procedura P este:

```
max ← a1,j
pentru i=1,n execută
  cheamă P1
sfîrșit-pentru
l ← l+1
xl ← max
sfîrșit-procedură
```

procedura Q este:

```
max ← a1,j
pentru i=2,n execută
  cheamă Q1
sfîrșit-pentru
l ← l+1
xl ← max
sfîrșit-procedură
```

procedura P1 este:

```
dacă  $\max\langle a_{i,j} \rangle$   
  atunci  $\max \leftarrow a_{i,j}$   
sfârșit-dacă  
dacă  $\max\langle a_{i,j+1} \rangle$   
  atunci  $\max \leftarrow a_{i,j+1}$   
sfârșit-dacă  
sfârșit-procedură
```

procedura Q1 este:

```
dacă  $a_{i,m} > \max$   
  atunci  $\max \leftarrow a_{i,m}$   
sfârșit-dacă  
sfârșit-procedură
```

```
10 REM  
20 REM *** FORMARE SIR CU MAXIME ***  
30 INPUT "linii n=";n;" coloane=";m: LET n=ABS (INT n):  
LET m=ABS (INT m)  
90 IF n=0 OR m=0 THEN GO TO 80  
100 DIM a(n,m): DIM x(INT (m/2)+1): LET len=4  
110 PRINT "matricea A:"  
120 FOR i=1 TO n  
130   FOR j=1 TO m  
140     INPUT "a(";(i);",";(j);")=";a(i,j)  
150     GO SUB 410: PRINT a(i,j);  
160   NEXT j  
170   PRINT  
180 NEXT i  
190  
200 LET l=0  
210 FOR j=1 TO m-1 STEP 2  
220   LET max=a(1,j)  
230   FOR i=1 TO n  
240     IF max<a(i,j) THEN LET max=a(i,j)  
250     IF max<a(i,j+1) THEN LET max=a(i,j+1)  
260   NEXT i  
270   LET l=l+1: LET x(l)=max  
280 NEXT j  
290 IF j<>m THEN GO TO 360  
300 LET max=a(1,m)  
310 FOR i=2 TO n  
320   IF a(i,m)>max THEN LET max=a(i,m)  
330 NEXT i  
340 LET l=l+1: LET x(l)=max  
350  
360 PRINT "sirul format este:"  
370 FOR i=1 TO l  
380   PRINT x(i);" ";  
390 NEXT i: PRINT  
400 STOP  
410 FOR u=1 TO len-LEN (STR$ a(i,j)): PRINT " ";: NEXT u  
420 RETURN
```

matricea A:

4	7	0	-9	8
-12	3	5	88	0
87	-11	4	66	9
1	2	55	0	0
-8	-13	0	14	55

sirul format este:

87 88 55

6.8 Se inițializează cu zero un contor ce va număra elementele subunitare pozitive ale matricii, contor ce va servi atît ca indice de linie pentru matricea B cît și drept indice pentru șirul X . Dacă după parcurgerea întregii matrice contorul rămîne la valoarea inițială zero, matricea nu conține nici un element subunitar pozitiv.

```

algoritmul 6.8. este:
  citește n,m
  pentru i=1,n execută
  [ pentru j=1,m execută
    [ citește ai,j
    ] sfîrșit-pentru
  ] sfîrșit-pentru
  pentru i=1,n execută
  [ pentru j=1,m execută
    [ scrie ai,j
    ] sfîrșit-pentru
  ] sfîrșit-pentru
  k ← 0
  pentru i=1,n execută
  [ pentru j=1,m execută
    [ dacă (ai,j > 0) and (ai,j < 1) atunci k ← k+1
      [ bk,1 ← i
      [ bk,2 ← j
      [ xk ← ai,j
    ] sfîrșit-dacă
  ] sfîrșit-pentru
  ] sfîrșit-pentru
  dacă k=0 atunci scrie 'nici un element subunitar pozitiv'
  altfel [ pentru i=1,k execută
    [ scrie bi,1, bi,2
    ] sfîrșit-pentru
    [ pentru i=1,k execută
      [ scrie xi
      ] sfîrșit-pentru
    ] sfîrșit-pentru
  ] sfîrșit-dacă
stop

```

```

40 PAPER 0: INK 5: BORDER 0: BRIGHT 1: CLS
50 PRINT "*****"
60 PRINT "* FORMARE MATRICE SI SIR *"
70 PRINT "* ELEMENTE SUBUNITARE POZITIVE *"
80 PRINT "*****"
90 INPUT "linii n=";n;" coloane=";m: LET n=ABS (INT n):
LET m=ABS (INT m)
100 IF n=0 OR m=0 THEN GO TO 90
110 DIM a(n,m): DIM b(n*m,2): DIM x(n*m): LET len=6
120 PRINT "matricea A:"
130 FOR i=1 TO n
140 LET fata=3
150 FOR j=1 TO m
160 INPUT "a(";(i);";";(j);")=";a(i,j)
170 GO SUB 390: PRINT a(i,j);
180 NEXT j
190 PRINT
200 NEXT i

```



```

210
220 LET k=0
230 FOR i=1 TO n
240   FOR j=1 TO m
250     IF a(i,j)>0 AND a(i,j)<1 THEN LET k=k+1: LET b(k,
1)=i: LET b(k,2)=j: LET x(k)=a(i,j)
260   NEXT j
270 NEXT i
280 IF k=0 THEN PRINT "NICI UN element subunitar >0":
60 TO 370
290 PRINT "matricea indicilor:"
300 FOR i=1 TO k
310   PRINT "   ";b(i,1);"   ";b(i,2)
320 NEXT i
330 PRINT "sirul format este:"
340 FOR i=1 TO K
350   PRINT x(i);" ";
360 NEXT i: PRINT
370 STOP
380
390 FOR u=1 TO len-fata-LEN (STR$ (INT a(i,j)))
400   PRINT " ";
410 NEXT u
420 LET fata=LEN (STR$ (a(i,j)-INT (a(i,j))))
430 RETURN

```

```

*****
*   FORMARE MATRICE SI SIR   *
* ELEMENTE SUBUNITARE POZITIVE *
*****

```

matricea A:

0.8	1	6	0	-76	0.3
0.1	0.2	9	0	0.4	-8
0	1	66	-8	9	-2
0.89	2	3	5	0	-8
-7	0	0.66	2	1	5
-0.4	0	9	7	0.5	1
0	7	-7	0.9	2	4

matricea indicilor:

1	1
1	6
2	1
2	2
2	5
4	1
5	3
6	5
7	4

sirul format este:

0.8 0.3 0.1 0.2 0.4 0.89 0.66 0.5 0.9

6.9 a) Algoritmul este, în principiu, același ca la ordonarea elementelor unui șir (indiferent prin ce metodă), însă în loc să comparăm elementele unui șir comparăm elementele ultimei coloane a matricii și în loc să inversăm două elemente ale unui șir, inversăm element cu element două linii ale matricii.

algoritmul 6.9.a. este:

```

citește n,m
pentru i=1,n execută
  pentru j=1,m execută
    citește ai,j
  sfîrșit-pentru
sfîrșit-pentru
pentru i=1,n execută
  pentru j=1,m execută
    scrie ai,j
  sfîrșit-pentru
sfîrșit-pentru
sw ← 1
cît-timp sw=1 execută
  sw ← 0
  pentru i=1,n-1 execută
    dacă ai,m > ai+1,m atunci
      pentru j=1,m execută
        y ← ai,j
        ai,j ← ai+1,j
        ai+1,j ← y
      sfîrșit-pentru
      sw ← 1
    sfîrșit-dacă
  sfîrșit-pentru
sfîrșit-cît-timp
pentru i=1,n execută
  pentru j=1,m execută
    scrie ai,j
  sfîrșit-pentru
sfîrșit-pentru
stop

```

```

50 PRINT "*****"
60 PRINT "*" REARANJARE LINII CU "*"
70 PRINT "*" ORDONARE ULTIMA COLOANA "*"
80 PRINT "*****"
90 INPUT "linii n=";n;" coloane=";m: LET n=ABS (INT n):
LET m=ABS (INT m)
100 IF n=0 OR m=0 THEN GO TO 90
110 DIM a(n,m): LET len=4
120 PRINT "matricea A:"
130 FOR i=1 TO n
140 FOR j=1 TO m
150 INPUT "a(";(i);",";";(j);")=";a(i,j)
160 GO SUB 420: PRINT a(i,j);
170 NEXT j
180 PRINT
190 NEXT i
200
210 LET sw=1

```

```

220 IF sw=0 THEN GO TO 320
230 LET sw=0
240 FOR i=1 TO n-1
250   IF a(i,m)<=a(i+1,m) THEN GO TO 300
260   FOR j=1 TO m
270     LET y=a(i,j): LET a(i,j)=a(i+1,j): LET a(i+1,j)=y
280   NEXT j
290   LET sw=1
300 NEXT i
310 GO TO 220
320
330 PRINT ""matricea obtinuta:""
340 FOR i=1 TO n
350   FOR j=1 TO m
360     GO SUB 420: PRINT a(i,j);
370   NEXT j
380   PRINT
390 NEXT i: PRINT
400 STOP
410
420 FOR u=1 TO len-LEN (STR$ (INT a(i,j)))
430   PRINT " ";
440 NEXT u
450 RETURN

```

```

*****
* REARANJARE LINII CU *
* ORDONARE ULTIMA COLOANA *
*****

```

matricea A:

4	6	0	8
-8	9	-11	23
-8	7	0	1
8	123	6	-89
0	3	55	12
89	-67	33	6
6	0	9	13

matricea obtinuta:

8	123	6	-89
-8	7	0	1
89	-67	33	6
4	6	0	8
0	3	55	12
6	0	9	13
-8	9	-11	23

b) Se formează două șiruri: unul cu elementele ultimei coloane a matricei și altul cu indicii de linie ai acestora. Se ordonează crescător elementele primului șir, însă, concomitent cu inversarea a două elemente se inversează și elementele corespunzătoare din șirul de indici. Când primul șir este ordonat, al doilea șir indică în ce ordine trebuie puse liniile matricei pentru a obține o matrice cu proprietatea cerută în enunț.

algoritmul 6.9.b. este:

```
citește n,m
pentru i=1,n execută
  pentru j=1,m execută
    citește ai,j
  sfîrșit-pentru
sfîrșit-pentru
pentru i=1,n execută
  xi ← ai,m
  yi ← i
sfîrșit-pentru
cheamă ORD
pentru i=1,n execută
  cheamă P
sfîrșit-pentru
pentru i=1,n execută
  pentru j=1,m execută
    scrie bi,j
  sfîrșit-pentru
sfîrșit-pentru
stop
```

procedura P este:

```
k ← yi
pentru j=1,m execută
  bi,j ← ak,j
sfîrșit-pentru
sfîrșit-procedură
```

procedura ORD este:

```
execută
sw ← 0
pentru i=1,n-1 execută
  cheamă Q
sfîrșit-pentru
încm-cînd sw=0
sfîrșit-procedură
```

procedura Q este:

```
dacă xi > xi+1
  atunci sw ← 1
  z ← xi
  xi ← xi+1
  xi+1 ← z
  z ← yi
  yi ← yi+1
  yi+1 ← z
sfîrșit-dacă
sfîrșit-procedură
```

```
10 REM
20 REM *** REARANJARE LINII CUORDONARE COLOANA M ***
30 REM
40 PAPER 0: INK 5: BORDER 0: BRIGHT 1: CLS
50 PRINT "*****"
60 PRINT "* REARANJARE LINII CU *"
70 PRINT "* ORDONARE ULTIMA COLOANA *"
80 PRINT "* CU FORMAREA UNEI NOI MATRICI *"
90 PRINT "*****"
100 INPUT "linii n=";n;" coloane=";m
110 LET n=ABS (INT n): LET m=ABS (INT m)
120 IF n=0 OR m=0 THEN GO TO 100
130 DIM a(n,m): DIM b(n,m): DIM x(n): DIM y(n): LET len=4
140 PRINT "matricea A:"
150 FOR i=1 TO n
160 FOR j=1 TO m
170 INPUT "a(";i);",";j);")=";a(i,j)
180 LET w=a(i,j): GO SUB 500: PRINT a(i,j);
190 NEXT j
200 PRINT
210 NEXT i
220
230 FOR i=1 TO n
240 LET x(i)=a(i,m): LET y(i)=i
250 NEXT i
260 LET sw=0
270 FOR i=1 TO n-1
280 IF x(i)<=x(i+1) THEN GO TO 320
290 LET sw=1
```



```

300 LET z=x(i): LET x(i)=x(i+1): LET x(i+1)=z
310 LET z=y(i): LET y(i)=y(i+1): LET y(i+1)=z
320 NEXT i
330 IF sw=1 THEN GO TO 260
340 FOR i=1 TO n
350 LET k=y(i)
360 FOR j=1 TO m
370 LET b(i,j)=a(k,j)
380 NEXT j
390 NEXT i
400
410 PRINT "matricea obtinuta:"
420 FOR i=1 TO n
430 FOR j=1 TO m
440 LET w=b(i,j): GO SUB 500: PRINT b(i,j);
450 NEXT j
460 PRINT
470 NEXT i: PRINT
480 STOP
490
500 FOR u=1 TO len-LEN (STR$ (INT w))
510 PRINT " ";
520 NEXT u
530 RETURN

```

```

*****
* REARANJARE LINII CU *
* ORDONARE ULTIMA COLOANA *
* CU FORMAREA UNEI NOI MATRICI *
*****

```

matricea A:

9	0	9	9	8	7	6
5	4	4	4	5	6	78
56	678	0	8	8	6	67
21	123	4	6	7	6	7
646	5	7	7	8	8	9
6	6	45	6	89	-87	-9
0	-6	0	788	21	-5	-87
4	356	-76	1	243	8	9
87	5	78	12	3	56	67
89	88	54	33	654	133	45

matricea obtinuta:

0	-6	0	788	21	-5	-87
6	6	45	6	89	-87	-9
9	0	9	9	8	7	6
21	123	4	6	7	6	7
646	5	7	7	8	8	9
4	356	-76	1	243	8	9
89	88	54	33	654	133	45
56	678	0	8	8	6	67
87	5	78	12	3	56	67
5	4	4	4	5	6	78

6.10 Matricea T are $m+k$ linii și $n+l$ coloane.

```

algoritmul 6.10. este:
citește m,n,k,l
  pentru i=1,m execută
    pentru j=1,n execută
      citește ai,j
    sfîrșit-pentru
  sfîrșit-pentru
  pentru i=1,m execută
    pentru j=1,l execută
      citește bi,j
    sfîrșit-pentru
  sfîrșit-pentru
  pentru i=1,k execută
    pentru j=1,n execută
      citește ci,j
    sfîrșit-pentru
  sfîrșit-pentru
  pentru i=1,k execută
    pentru j=1,l execută
      citește di,j
    sfîrșit-pentru
  sfîrșit-pentru
  pentru i=1,m execută
    cheamă P
  sfîrșit-pentru
  pentru i=1,k execută
    cheamă Q
  sfîrșit-pentru
  pentru i=1,m+k execută
    pentru j=1,n+l execută
      scrie ti,j
    sfîrșit-pentru
  sfîrșit-pentru
stop
  
```

```

procedura P este:
  pentru j=1,n execută
    ti,j ← ai,j
  sfîrșit-pentru
  pentru j=1,l execută
    ti,j+n ← bi,j
  sfîrșit-pentru
sfîrșit-procedură
  
```

```

procedura Q este:
  pentru j=1,n execută
    ti+m,j ← ci,j
  sfîrșit-pentru
  pentru j=1,l execută
    ti+m,j+n ← di,j
  sfîrșit-pentru
sfîrșit-procedură
  
```

```

10 PRINT " *****"
20 PRINT " *   DIN 4 MATRICE   *"
30 PRINT " * FORMAM O ALTA MATRICE *"
40 PRINT " *****"
50 INPUT "Dimensiunile matricei A" / "M=";M,"N=";N
60 DIM A(M,N): CLS
70 PRINT "" MATRICEA A ESTE:"
80 FOR I=1 TO M
90   FOR J=1 TO N
100    INPUT A(I,J)
110    PRINT AT I+4,J*4-LEN STR$ A(I,J)+4;A(I,J)
120   NEXT J
130 NEXT I
140 INPUT "Cite coloane are matricea B ?", "L=";L
150 DIM B(M,L): CLS
160 PRINT "" MATRICEA B ESTE:"
170 FOR I=1 TO M
180   FOR J=1 TO L
190    INPUT B(I,J)
200    PRINT AT I+4,J*4-LEN STR$ B(I,J)+4;B(I,J)
210   NEXT J
220 NEXT I
  
```

```

230 INPUT "Cite linii are matricea C ?" "K=";K
240 DIM C(K,N): CLS
250 PRINT ""          MATRICEA C ESTE:"
260 FOR I=1 TO K
270     FOR J=1 TO N
280         INPUT C(I,J)
290         PRINT AT I+4,J*4-LEN STR$ C(I,J)+4;C(I,J)
300     NEXT J
310 NEXT I
320 DIM D(K,L): DIM T(M+K,N+L): CLS
330 PRINT ""          MATRICEA D ESTE:"
340 FOR I=1 TO K
350     FOR J=1 TO L
360         INPUT D(I,J)
370         PRINT AT I+4,J*4-LEN STR$ D(I,J)+4;D(I,J)
380     NEXT J
390 NEXT I
400 CLS : PRINT ""    MATRICEA FORMATA ESTE:"
410 FOR I=1 TO M
420     FOR J=1 TO N
430         LET T(I,J)=A(I,J)
440     NEXT J
450     FOR J=1 TO L
460         LET T(I,J+N)=B(I,J)
470     NEXT J
480 NEXT I
490 FOR I=1 TO K
500     FOR J=1 TO N
510         LET T(I+M,J)=C(I,J)
520     NEXT J
530     FOR J=1 TO L
540         LET T(I+M,J+N)=D(I,J)
550     NEXT J
560 NEXT I
570 FOR I=1 TO M+K
580     FOR J=1 TO N+L
590         PRINT AT I+4,J*4-LEN STR$ T(I,J);T(I,J)
600     NEXT J
610 NEXT I
620 PAUSE 0: CLS : STOP

```

```

*****
*   DIN 4 MATRICE   *
* FORMAM O ALTA MATRICE *
*****

```

MATRICEA A ESTE:

```

1  1  1
1  1  1

```

MATRICEA B ESTE:

```

2  2  2  2
2  2  2  2

```

MATRICEA C ESTE:

```
0 0 0
0 0 0
0 0 0
```

MATRICEA D ESTE:

```
9 9 9 9
9 9 9 9
9 9 9 9
```

MATRICEA FORMATA ESTE:

```
1 1 1 2 2 2 2
1 1 1 2 2 2 2
0 0 0 9 9 9 9
0 0 0 9 9 9 9
0 0 0 9 9 9 9
```

6.11 Matricea T are $2n+m$ linii și $m+k+1$ coloane.

```
10 REM
20 REM *** CONCATRNARE MATRICI ***
30 REM
40 PAPER 0: INK 5: BORDER 0: BRIGHT 1: CLS
50 PRINT "*****"
60 PRINT "*          CONCATENARE MATRICI          *"
70 PRINT "*****"
80 INPUT "n=";n;" m=";m;" k=";k
90 LET n=ABS (INT n): LET m=ABS (INT m): LET k=ABS (INT k)
100 IF n=0 OR m=0 OR k=0 THEN GO TO 30
110 DIM a(2*n,m): DIM b(n,k): DIM c(2*n)
120 DIM d(m,k+1): DIM t(2*n+m,m+k+1): LET len=4
130 PRINT "matricea A:"
140 FOR i=1 TO 2*n
150   FOR j=1 TO m
160     INPUT "a(";i);", ";(j);")=";a(i,j)
170     LET w=a(i,j): GO SUB 710: PRINT a(i,j);
180   NEXT j
190   PRINT
200 NEXT i
210 PRINT "matricea B:"
220 FOR i=1 TO n
230   FOR j=1 TO k
240     INPUT "b(";i);", ";(j);")=";b(i,j)
250     LET w=b(i,j): GO SUB 710: PRINT b(i,j);
260   NEXT j
270   PRINT
280 NEXT i
290 PRINT "sirul C:"
300 FOR i=1 TO 2*n
310   INPUT "c(";i);")=";c(i)
320   LET w=c(i): GO SUB 710: PRINT c(i);
330 NEXT i
340 PRINT "matricea D"
```



```

350 FOR i=1 TO m
360   FOR j=1 TO k+1
370     INPUT "d(";i);", ";(j);")=";d(i,j)
380     LET w=d(i,j): GO SUB 710: PRINT d(i,j);
390   NEXT j
400   PRINT
410 NEXT i
420 FOR i=1 TO n
430   FOR j=1 TO m
440     LET t(i,j)=a(i,j)
450     LET t(n+i,j)=a(n+i,j)
460   NEXT j
470   IF j>m+k THEN GO TO 510
480   LET t(i,j)=b(i,j-m)
490   LET t(n+i,j)=0
500   LET j=j+1: GO TO 470
510   LET t(i,j)=c(i): LET t(n+i,j)=c(i+n)
520 NEXT i
530 FOR i=2*n+1 TO 2*n+m
540   FOR j=1 TO m
550     IF j=i-2*n THEN LET t(i,j)=1: GO TO 570
560     LET t(i,j)=0
570   NEXT j
580   IF j>m+k+1 THEN GO TO 610
590   LET t(i,j)=d(i-2*n,j-m)
600   LET j=j+1: GO TO 580
610 NEXT i
620
630 PRINT "matricea obtinuta:"
640 FOR i=1 TO 2*n+m
650   FOR j=1 TO m+k+1
660     GO SUB 710: PRINT t(i,j);
670   NEXT j
680   PRINT
690 NEXT i: PRINT
700 STOP
710 FOR u=1 TO len-LEN (STR$ (INT w)): PRINT " "; NEXT u
720 RETURN

```

```

*****
*      CONCATENARE MATRICI      *
*****

```

matricea A:

1	2	3	4
5	6	7	8
9	8	7	6
5	4	3	2
1	2	3	4
5	6	7	8

matricea B:

9	9	9	9	9
9	9	9	9	9
9	9	9	9	9

sirul C:

5	5	5	5	5	5
---	---	---	---	---	---

matricea D

1	1	1	1	1	1
1	1	1	1	1	1
1	1	1	1	1	1
1	1	1	1	1	1

matricea obtinuta:

1	2	3	4	9	9	9	9	9	5
5	6	7	8	9	9	9	9	9	5
9	8	7	6	9	9	9	9	9	5
5	4	3	2	0	0	0	0	0	5
1	2	3	4	0	0	0	0	0	5
5	6	7	8	0	0	0	0	0	5
1	0	0	0	1	1	1	1	1	1
0	1	0	0	1	1	1	1	1	1
0	0	1	0	1	1	1	1	1	1
0	0	0	1	1	1	1	1	1	1

6.12

```
10 PRINT " *****"
20 PRINT " * GENERAREA UNEI *"
30 PRINT " * MATRICI PATRATE *"
40 PRINT " * DIN ALTA MATRICE *"
50 PRINT " *****"
60 INPUT "Introdu dimensiunea matricei N=";N
70 DIM A(N,N)
80 PRINT "" MATRICEA INITIALA ESTE:""
90 FOR I=1 TO N
100 FOR J=1 TO N
110 INPUT A(I,J)
120 PRINT AT I+9,J*4-LEN STR$ A(I,J)+4;A(I,J)
130 NEXT J
140 NEXT I
150 FOR I=1 TO N
160 LET A(I,I)=0: LET A(I,N-I+1)=0
170 NEXT I
180 CLS : PRINT "" MATRICEA OBTINUTA ESTE:"
190 FOR I=1 TO N
200 FOR J=1 TO N
210 PRINT AT I+4,J*4-LEN STR$ A(I,J)+4;A(I,J)
220 NEXT J
230 NEXT I
240 PAUSE 0: CLS : STOP
```

```
*****
* GENERAREA UNEI *
* MATRICI PATRATE *
* DIN ALTA MATRICE *
*****
```

MATRICEA INITIALA ESTE:

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16

MATRICEA OBTINUTA ESTE:

0	2	3	0
5	0	0	8
9	0	0	12
0	14	15	0

6.13 Se parcurge doar triunghiul superior al matricei (format din elementele situate deasupra diagonalei principale), deoarece elementele de pe diagonală rămân neschimbate iar elementele de dedesubtul diagonalei principale vor fi egale cu cele de deasupra, matricea ce se formează fiind simetrică.

algoritmul 6.13. este:

```

citește n,m
  pentru i=1,n execută
    pentru j=1,m execută
      citește ai,j
    sfîrșit-pentru
  sfîrșit-pentru
  pentru i=1,n-1 execută
    cheamă P
  sfîrșit-pentru
  bn,n ← an,n
  pentru i=1,n execută
    pentru j=1,n execută
      scrie bi,j
    sfîrșit-pentru
  sfîrșit-pentru
stop
  
```

procedura P este:

```

bi,i ← ai,i
  pentru j=i+1,n execută
    bi,j ←  $\frac{a_{i,j} + a_{j,i}}{2}$ 
  sfîrșit-pentru
  sfîrșit-procedură
  
```

```

10 PRINT " *****"
20 PRINT " * GENERAREA UNEI *"
30 PRINT " * MATRICI SIMETRICE *"
40 PRINT " * DINTR-O MATRICE *"
50 PRINT " * PATRATA DATA *"
60 PRINT " *****"
70 INPUT "Introdu dimensiunea matricei N=";N
80 DIM A(N,N): DIM B(N,N)
90 PRINT "" MATRICEA INITIALA ESTE:"""
100 FOR I=1 TO N
110 FOR J=1 TO N
120 INPUT A(I,J)
130 PRINT AT I+1,J*4-LEN STR$ A(I,J)+3;A(I,J)
140 NEXT J
150 NEXT I
160 FOR I=1 TO N
170 FOR J=1 TO N
180 LET B(I,J)=(A(I,J)+A(J,I))/2
190 LET B(J,I)=B(I,J)
200 NEXT J
210 NEXT I
220 CLS : PRINT "" MATRICEA SIMETRICA GENERATA:""
230 FOR I=1 TO N
240 FOR J=1 TO N
250 PRINT AT I+4,J*4-LEN STR$ B(I,J)+4;B(I,J)
260 NEXT J
270 NEXT I
280 PAUSE 0: CLS : STOP
  
```

```

*****
*   GENERAREA UNEI *
* MATRICI SIMETRICE *
* DINTR-O MATRICE *
*   PATRATA DATA *
*****

```

MATRICEA INITIALA ESTE:

1	2	3	4
6	5	2	2
5	4	1	-1
-2	0	1	6

MATRICEA SIMETRICA GENERATA:

1	4	4	1
4	5	3	1
4	3	1	0
1	1	0	6

6.14 Este suficientă o singură parcurgere a matricii T , deoarece pot fi inițializate simultan patru elemente ale matricii A ce sînt egale cu elementul t_{ij} și anume cele de indici i și j , $n+i$ și $n+j$, j și $n+i$, $n+j$ și i .

```

algoritmul 6.14. este:
citește n
pentru i=1,n execută
  pentru j=1,n execută
    citește  $t_{i,j}$ 
    sfîrșit-pentru
  sfîrșit-pentru
pentru i=1,n execută
  cheamă P
sfîrșit-pentru
 $m \leftarrow 2 \cdot n$ 
pentru i=1,m execută
  pentru j=1,m execută
    scrie  $a_{i,j}$ 
  sfîrșit-pentru
sfîrșit-pentru
stop

```

```

procedura P este:
pentru j=1,n execută
   $a_{i,j} \leftarrow t_{i,j}$ 
   $a_{n+i,n+j} \leftarrow t_{i,j}$ 
   $a_{j,n+i} \leftarrow t_{i,j}$ 
   $a_{j+n,i} \leftarrow t_{i,j}$ 
sfîrșit-pentru
sfîrșit-procedură

```



```

10 PRINT " *****"
20 PRINT " * DIN MATRICE PATRATA *"
30 PRINT " * FORMAM O ALTA MATRICE *"
40 PRINT " *****"
50 INPUT "Introdu dimensiunea matricei N=";N
60 DIM T(N,N): DIM A(2*N,2*N)
70 CLS : PRINT "" MATRICEA ESTE:"
80 FOR I=1 TO N
90   FOR J=1 TO N
100     INPUT T(I,J)
110     PRINT AT I+5,J*4-LEN STR$ T(I,J);T(I,J)
120   NEXT J
130 NEXT I
140 FOR I=1 TO N
150   FOR J=1 TO N
160     LET A(I,J)=T(I,J): LET A(N+I,N+J)=T(I,J)
170     LET A(J,N+I)=T(I,J): LET A(N+J,I)=T(I,J)
180   NEXT J
190 NEXT I
200 CLS : PRINT "" MATRICEA OBTINUTA ESTE:"
210 FOR I=1 TO 2*N
220   FOR J=1 TO 2*N
230     PRINT AT I+5,J*4-LEN STR$ A(I,J);A(I,J)
240   NEXT J
250 NEXT I
260 PAUSE 0: CLS : STOP

```

```

*****
* DIN MATRICE PATRATA *
* FORMAM O ALTA MATRICE *
*****

```

MATRICEA ESTE:

1	2	3
4	5	6
7	8	9

MATRICEA OBTINUTA ESTE:

1	2	3	1	4	7
4	5	6	2	5	8
7	8	9	3	6	9
1	4	7	1	2	3
2	5	8	4	5	6
3	6	9	7	8	9

6.15 a) Stiind că pe prima paralelă deasupra diagonalei principale sînt $n-1$ elemente, este suficientă o singură structură repetitivă.

b) Dacă se pun în șir cite un element de deasupra diagonalei principale urmat de unul de dedesubtul ei, este necesară o singură structură repetitivă (se ține cont de faptul că simetricul unui element de pe prima paralelă deasupra diagonalei principale este un element situat pe prima paralelă de dedesubtul acesteia).

```

algoritmul 6.15. este:
  citește n
  pentru i=1,n execută
  [
    pentru j=1,n execută
    [
      citește ai,j
    ]
    sfârșit-pentru
  ]
  sfârșit-pentru
  execută
  [
    citește v
    dacă v='a' atunci cheamă PA
    altfel cheamă PB
  ]
  sfârșit-dacă
  ieșim-cînd (v≠'a') and (v≠'b')
stop

```

```

procedura PA este:
  l ← 0
  pentru i=1,n-1 execută
  [
    l ← l+1
    xl ← ai,l+1
  ]
  sfârșit-pentru
  pentru i=1,n-1 execută
  [
    l ← l+1
    xl ← ai+1,i
  ]
  sfârșit-pentru
  pentru i=1,l execută
  [
    scrie xi
  ]
  sfârșit-pentru
sfârșit-procedură

```

```

procedura PB este:
  l ← 0
  pentru i=1,n-1 execută
  [
    l ← l+2
    xl-1 ← ai,i+1
    xl ← ai+1,i
  ]
  sfârșit-pentru
  pentru i=1,l execută
  [
    scrie xi
  ]
  sfârșit-pentru
sfârșit-procedură

```

```

50 PRINT ""      1) IN ORDINEA INDICATA"
60 PRINT "      2) IN ORICE ORDINE"
70 PAUSE 0
80 IF INKEY$ <> "1" AND INKEY$ <> "2" THEN GO TO 80
90 IF INKEY$ = "1" THEN LET SW=1: GO TO 110
100 LET SW=2
110 CLS : INPUT "Introdu dimensiunea matricei N="; N
120 DIM A(N,N): DIM X(N*2)
130 PRINT ""      MATRICEA ESTE:"
140 FOR I=1 TO N
150   FOR J=1 TO N
160     INPUT A(I,J)
170     PRINT AT I+4, J*4-LEN STR$ A(I,J)+4; A(I,J)
180   NEXT J
190 NEXT I
200 IF SW=1 THEN GO SUB 260: GO TO 220
210 GO SUB 350
220 PRINT ""      SIRUL OBTINUT ESTE:""" X(=";
230 FOR I=1 TO L: PRINT " "; X(I);: NEXT I
240 PRINT " )"
250 PAUSE 0: CLS : GO TO 50
260 REM *** caz1 ***
270 LET L=0
280 FOR I=1 TO N-1
290   LET L=L+1: LET X(L)=A(I,I+1)
300   LET X(L+N-1)=A(I+1,I)
310 NEXT I
320 LET L=L+N-1
330 RETURN
340 REM *** caz2 ***
350 LET L=0

```

```

360 FOR I=1 TO N-1
370   LET L=L+2: LET X(L-1)=A(I,I+1)
380   LET X(L)=A(I+1,I)
390 NEXT I
400 RETURN

```

a) IN ORDINEA INDICATA

b) IN ORICE ORDINE

MATRICEA ESTE:

```

  1  2  3  4
  5  6  7  8
  9 10 11 12
 13 14 15 16

```

SIRUL OBTINUT ESTE:

X=(2 7 12 5 10 15)

MATRICEA ESTE:

```

  1  2  3  4
  5  6  7  8
  9 10 11 12
 13 14 15 16

```

SIRUL OBTINUT ESTE:

X=(2 5 7 10 12 15)

6.16 Cum $a_{ii} = -a_{ii}$ pentru orice indice i rezultă că orice matrice antisimetrică are elementele diagonalei principale nule, ceea ce înseamnă că șirul va avea

$$1 + 2 + \dots + n-1 = \frac{n \cdot (n-1)}{2}$$

elemente. Reciproc, având elementele șirului și presupunind că s-a lucrat pe linie, reținându-se elementele situate deasupra diagonalei principale, se calculează mai întâi dimensiunea n a matricei în funcție de dimensiunea m a șirului, rezolvând ecua-

ția în $n:m = \frac{n \cdot (n-1)}{2}$ care are ca unică soluție pozitivă numărul:

$$n = \frac{1 + \sqrt{1 + 8 \cdot m}}{2}$$

număr ce trebuie să fie natural pentru ca problema să aibă sens. Datorită modului de reprezentare internă a datelor numerice în interpretorul BASIC, nu se testează egalitatea $n = \lceil n \rceil$ ci inegalitatea $n - \lceil n \rceil < \epsilon$, unde ϵ este un număr pozitiv dat, suficient de mic.

procedura P1 este:

- pentru $j=i+1, n$ execută
 - $l \leftarrow l+1$
 - $a_{l,j} \leftarrow x_l$
 - $a_{j,l} \leftarrow -x_l$
- sfârșit-pentru
- sfârșit-procedură

```

algoritmul 6.16.b. este:
citește m
pentru i=1,m execută
    citește xi
sfârșit-pentru

d ← √(1+8·m)
x ← (-1+d) / 2

dacă x=[x]
    atunci cheamă P
    altfel scrie
        'problema e imposibilă'
sfârșit-dacă
stop

```

```

procedura P este:
n ← x
l ← 0
pentru i=1,n-1 execută
    ai,i = 0
    cheamă P1
sfârșit-pentru
an,n ← 0

pentru i=1,n execută
    pentru j=1,n execută
        scrie ai,j
    sfârșit-pentru
sfârșit-pentru
sfârșit-procedură

```

```

70 PAUSE 0: CLS : PRINT "" 1.PASTRAREA ELEMENTELOR UNEI"
80 PRINT "MATRICI ANTISIMETRICE INTR-UN"
90 PRINT "SIR DE LUNGIME MINIMA"
100 PRINT "" 2.FORMAREA UNEI MATRICI AN-"
110 PRINT "TISIMETRICE CU ELEMENTELE UNUI"/"SIR"
120 PAUSE 0: IF INKEY$<>"1" AND INKEY$<>"2" THEN GO TO 120
130 CLS
140 IF INKEY$="2" THEN GO TO 400
150 INPUT "Introdu dimensiunea matricei N=";N
160 DIM A(N,N): DIM X(N*(N-1)/2)
170 CLS : PRINT "" MATRICEA INITIALA ESTE:"
180 FOR I=1 TO N
190     FOR J=1 TO N
200         INPUT A(I,J)
210         PRINT AT I+4,J*4-LEN STR$ A(I,J)+4;A(I,J)
220     NEXT J
230 NEXT I
240 LET SW=0
250 FOR I=1 TO N-1
260     FOR J=I+1 TO N
270         IF A(I,J)<>-A(J,I) THEN LET SW=1
280     NEXT J
290 NEXT I
300 IF SW=1 THEN PRINT "MATRICEA NU ESTE ANTISIMETRICA": PAUS
E0: GO TO 160
310 LET L=0
320 FOR I=1 TO N-1
330     FOR J=I+1 TO N
340         LET L=L+1: LET X(L)=A(I,J)
350     NEXT J
360 NEXT I
370 PRINT "" SIRUL GENERAT ESTE:""" "
380 FOR J=1 TO L: PRINT X(J);" ";: NEXT J
390 GO TO 70
400 REM matrice din sir
410 INPUT "Introdu dimensiunea sirului M=";M
420 LET N=(1+SQR (1+8*M))/2
430 IF ABS (N-INT N)>0.003 THEN GO TO 410
440 PRINT #0;"N=";N: DIM X(M): DIM A(N,N)
450 PRINT "" SIRUL DAT ESTE:""" ( ";
460 FOR I=1 TO M
470     INPUT X(I): PRINT X(I);" ";
480 NEXT I
140

```



```

490 PRINT ")": LET L=1
500 FOR I=1 TO N-1
510   FOR J=I+1 TO N
520     LET A(I,J)=X(L): LET A(J,I)=-X(L): LET L=L+1
530   NEXT J
540 NEXT I
550 PRINT "" MATRICEA ESTE:""
560 FOR I=1 TO N
570   FOR J=1 TO N
580     PRINT AT I+10,J*4-LEN STR$ A(I,J);A(I,J)
590   NEXT J
600 NEXT I
610 GO TO 70

```

1.PASTRAREA ELEMENTELOR UNEI
MATRICI ANTISIMETRICE INTR-UN
SIR DE LUNGIME MINIMA

2.FORMAREA UNEI MATRICI AN-
TISIMETRICE CU ELEMENTELE UNUISIR

MATRICEA INITIALA ESTE:

0	1	2	3
-1	0	3	4
-2	-3	0	5
-3	-4	-5	0

SIRUL GENERAT ESTE:

1 2 3 3 4 5

1.PASTRAREA ELEMENTELOR UNEI
MATRICI ANTISIMETRICE INTR-UN
SIR DE LUNGIME MINIMA

2.FORMAREA UNEI MATRICI AN-
TISIMETRICE CU ELEMENTELE UNUISIR

SIRUL DAT ESTE:

(1 2 3 4 5 6
)

MATRICEA ESTE:

0	1	2	3
-1	0	4	5
-2	-4	0	6
-3	-5	-6	0

6.17 Soluția acestei probleme se poate obține ușor modificând programul 6.16

6.18 Se ține cont de faptul că elementele situate deasupra diagonalei secundare sînt simetricele pe linie ale elementelor situate deasupra diagonalei principale.

algoritmul 6.18. este:

```

citește n
pentru i=1,n execută
  pentru j=1,n execută
    citește ai,j
  sfîrșit-pentru
sfîrșit-pentru
l ← 0
pentru i=1,n execută
  cheamă P
sfîrșit-pentru
pentru i=1,l execută
  scrie xi
sfîrșit-pentru
stop

```

procedura P este:

```

pentru j=i,n execută
  dacă ai,n-j+1 > 0
    atunci l ← l+1
    xl ← ai,n-j+1
sfîrșit-dacă
sfîrșit-pentru
sfîrșit-procedură

```

```

10 PRINT " *****"
20 PRINT " * GENERAREA UNUI SIR *"
30 PRINT " * CU UNELE ELEMENTE ALE *"
40 PRINT " * UNEI MATRICI PATRATE *"
50 PRINT " *****"
60 INPUT "Introdu dimensiunea matricei N=";N
70 DIM A(N,N): DIM X(N*(N-1)/2)
80 PRINT "" MATRICEA INITIALA ESTE:"
90 FOR I=1 TO N
100 FOR J=1 TO N
110 INPUT A(I,J)
120 PRINT AT I+10,J*4-LEN STR$ A(I,J)+3;A(I,J)
130 NEXT J
140 NEXT I
150 LET L=0
160 FOR I=1 TO N-1
170 FOR J=I+1 TO N
180 LET J1=N+1-J
190 IF A(I,J1)>0 THEN LET L=L+1: LET X(L)=A(I,J1)
200 NEXT J
210 NEXT I
220 PRINT "" SIRUL GENERAT ESTE:""" ";
230 FOR I=1 TO L
240 PRINT X(I);" ";
250 NEXT I
260 PAUSE 0: CLS : STOP

```

```

*****
* GENERAREA UNUI SIR *
* CU UNELE ELEMENTE ALE *
* UNEI MATRICI PATRATE *
*****

```

MATRICEA INITIALA ESTE:

1	-3	0	4
-2	1	4	0
3	-2	1	4
-5	2	4	3

SIRUL GENERAT ESTE:

1 1 3

6.19

```

10 PRINT " *****"
20 PRINT " * DIN MATRICEA PATRATA *"
30 PRINT " * FORMAM UN SIR *"
40 PRINT " *****"
50 INPUT "Introdu dimensiunea matricei N=";N
60 PRINT "" MATRICEA ESTE:"""
70 DIM A(N,N)
80 FOR I=1 TO N
90 FOR J=1 TO N
100 INPUT A(I,J)
110 PRINT AT I+9,J*4-LEN STR$ A(I,J)+4;A(I,J)
120 NEXT J
130 NEXT I
140 DIM X(N)
150 FOR I=1 TO N
160 LET S1=0: LET S2=0
170 FOR J=1 TO I
180 LET S1=S1+A(I,J): LET S2=S2+A(J,I)
190 NEXT J
200 IF S1<=S2 THEN LET X(I)=-1: GO TO 220
210 LET X(I)=1
220 NEXT I
230 PRINT "" SIRUL FORMAT ESTE:""" X=( ";
240 FOR I=1 TO N
250 PRINT X(I);" ";
260 NEXT I
270 PRINT ")"
280 PAUSE 0: CLS : STOP

```

```

*****
* DIN MATRICEA PATRATA *
* FORMAM UN SIR *
*****

```

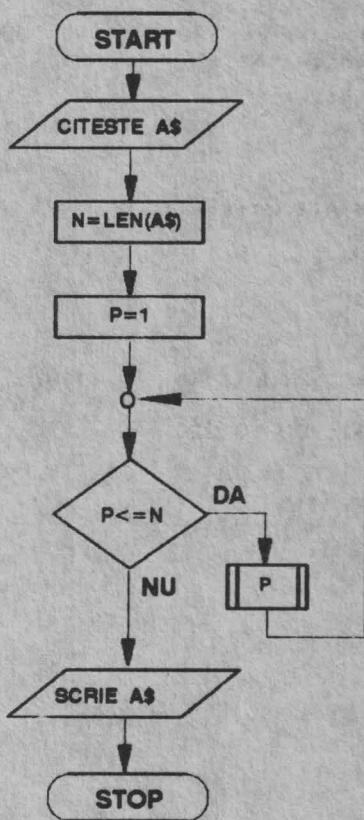
MATRICEA ESTE:

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16

SIRUL FORMAT ESTE:

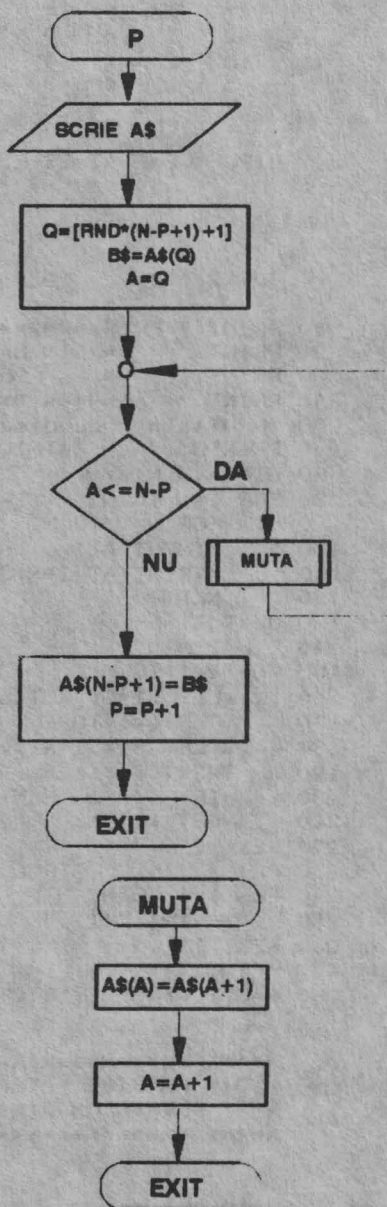
X=(-1 1 1 1)

7.1 Se va genera un număr aleator cuprins între 1 și $l = \text{LEN}(\text{șir})$; elementul respectiv este trecut la sfârșitul șirului, cu decalarea celorlalte cu un pas în față, apoi cu șirul de lungime $l-1$ se repetă procedeul. Această metodă poate fi folosită cu succes la amestecarea literelor unui joc de SCRABBLE pe calculator, la amestecarea cărților la un joc de cărți etc.



algoritmul 7.1. este:
 citește a\$
 $n \leftarrow \text{LEN}(a\$)$
 pentru $p=1, n$ execută
 cheamă P
 sfârșit-pentru
 scrie a\$
 stop

procedura P este:
 scrie a\$
 $q \leftarrow \text{RND} \cdot (n-p+1)+1$
 $b\$ \leftarrow a\(q)
 pentru $a=q, n-p$ execută
 $a\$(a) \leftarrow a\$(a+1)$
 sfârșit-pentru
 $a\$(n-p+1) \leftarrow b\$$
 sfârșit-procedură




```

10 REM
20 REM *** DISTRIBUTIE ALEATOARE ***
30 REM
40 PAPER 0: INK 7: BORDER 0: BRIGHT 1: CLS
50 PRINT "*****"
60 PRINT "*      DISTRIBUTIE ALEATOARE      *"
70 PRINT "*****"
80 LET p$="  Dindu-se o variabila sir, sase scrie un pro
gram care sa dis-tribue in mod aleator elemente-le acestui
sir in el insusi."
90 FOR i=1 TO LEN p$: PRINT p$(i);: BEEP 0.005,15: NEXT i
100 PRINT '': FOR i=1 TO 32: PRINT "-": NEXT i: PRINT ''
110 INPUT "a$-";a$
120 PRINT "Sirul initial:"
130 PRINT '
140 INVERSE 1: PRINT a$'': INVERSE 0
150 RANDOMIZE
160 LET n=LEN a$
170 FOR p=1 TO n
180 LET q=INT (RND*(n-p+1)+1)
190 LET b$=a$(q)
200 FOR a=q TO n-p
210 LET a$(a)=a$(a+1)
220 NEXT a
230 LET a$(n-p+1)=b$
240 PRINT "p=";: GO SUB 310: PRINT p;"  ";a$
250 NEXT p
260 PRINT ' "Sirul redistribuit:"'
270 PRINT '': INVERSE 1
280 PRINT a$: INVERSE 0
290 STOP
300
310 FOR k=1 TO 2-LEN (STR$ p): PRINT " ";: NEXT k: RETURN

```

```

*****
*      DISTRIBUTIE ALEATOARE      *
*****

```

Sirul initial:

siraleator

```

p= 1  siraleatro
p= 2  iraleatrso
p= 3  iraleartso
p= 4  iraleartso
p= 5  ialearrtso
p= 6  ialaerttso
p= 7  alaierttso
p= 8  aalierttso
p= 9  aalierttso
p=10  aalierttso

```

Sirul redistribuit:

aalierttso

7.2 Soluția este extrem de simplă, ea bazându-se pe proprietățile operațiilor cu șiruri de caractere: primul caracter se mută la sfârșitul șirului, celelalte decalându-se cu cite o poziție în față în șir. Apoi șirul se rescrie în același lbc pe ecran. Mesajul poate fi memorat în variabilă prin citire cu INPUT sau prin atribuire directă cu LET. Am folosit pentru testarea tastei apășate funcția INKEY\$.

```

10 REM
20 REM *** ROTIRE MESAJ ***
30 REM
40 PAPER 0: INK 7: BORDER 0: BRIGHT 1: CLS
50 PRINT "*****"
60 PRINT "*           ROTIRE MESAJ           *"
70 PRINT "*****"
80 LET p$=" Dindu-se un mesaj memorat in-tr-o variabila
sir , sa se scrieun program care sa "roteasca" peecran ace
st mesaj (caracterele care ""ies"" prin stinga sa ""rein-tr
e"" prin dreapta, pina la apasarea unei anumite taste."
90 FOR i=1 TO LEN p$: PRINT p$(i); BEEP 0.005,15: NEXT i
100 PRINT "": FOR i=1 TO 32: PRINT "-";: NEXT i: PRINT " "
110 INPUT "Mesajul:";a$: LET a$=" "+a$+" "
120 INPUT "tasta pentru oprire:";t$: LET t$=t$(1)
130 INPUT "Mesajul se scrie la x=";x;" y=";y
140 LET x=ABS (INT x): LET y=ABS (INT y)
150 IF x<0 OR x>21 OR y+LEN a$>31 OR y<0 THEN GO TO 130
160 INVERSE 1
170 PRINT AT x,y;a$: PAUSE 6
180 LET a$=a$(2 TO )+a$(1)
190 IF INKEY$<>t$ THEN GO TO 170
200 INVERSE 0

```

7.3 Se inițializează a\$ cu șirul vid la care se adună (se concatenează) pe rind caracterele tastate, dacă acestea sînt cifre și dacă lungimea șirului deja format nu depășește numărul de cifre. Sînt analizate separat cazurile CHR\$(12) (ștergerea unui caracter) și CHR\$(13) (sfîrșitul introducerii): se poate șterge un caracter și se poate incheia introducerea dacă a fost introdus cel puțin un caracter. Pentru ștergerea pe ecran am folosit și caracterul special CHR\$(8) - BACK SPACE.

```

[ algoritmul 7.3. este:
  citește n
  a$ ← ''
  citește b$
  [ cît-timp (b$≠CHR$(13)) or (LEN(a$)=0) execută
    cheamă P
  ] sfîrșit-cît-timp
  scrie a$
  număr ← VAL(a$)
  scrie număr
] stop

[ procedura P este:
  [ dacă '0'≤b$≤'9' atunci [ dacă LEN(a$)<n
    [ atunci a$ ← a$+b$
    ] sfîrșit-dacă
  ] altfel 0
  ] sfîrșit-dacă
] sfîrșit-procedură

```

procedura Q este:

```
    dacă b$=CHR$(12) atunci      [  dacă LEN(a$)>0
                                  [  atunci a$ ← a$(TO LEN(a$)-1)
                                  sfîrșit-dacă
    altfel                        [  dacă b$='-'
                                  atunci [  dacă LEN(a$)=0
                                          [  atunci a$ ← a$+b$
                                          sfîrșit-dacă
                                  sfîrșit-dacă
    sfîrșit-dacă
sfîrșit-procedură
```

```
40 PAPER 0: INK 7: BORDER 0: BRIGHT 1: CLS
50 PRINT "*****"
60 PRINT "*   INTRODUCERE NUMAR INTREG   *"
70 PRINT "*****"
80 PRINT " Sa se scrie un program care sa" / "permita introd
ucerea de la tasta" / "tura a unui intreg, din maxim n" / "carac
tere, cu interzicerea tasta" / "rii altor caractere"
90 PRINT "/: FOR i=1 TO 32: PRINT "-";: NEXT i: PRINT "/
100 INPUT "Numar caractere:";n: LET n=INT n
110 IF n<1 THEN GO TO 100
120 LET a$=""
130 PRINT AT 21,0;"Numarul este:";
140 PAUSE 0: LET b$=INKEY$
150 IF CODE b$<CODE "0" OR CODE b$>CODE "9" THEN GO TO 170
160 PRINT b$ AND LEN a$<n;: LET a$=a$+(b$ AND LEN a$<n)
170 IF b$=CHR$ 13 AND LEN a$<>0 THEN GO TO 210
180 IF b$=CHR$ 12 AND LEN a$>0 THEN LET a$=a$( TO LEN a$-1)
: PRINT CHR$ 8;" ";CHR$ 8;
190 IF CODE b$=CODE "-" AND LEN a$=0 THEN LET a$=a$+b$: PRI
NT b$;
200 GO TO 140
210 PRINT AT 13,0;"Sirul format:";a$
220 IF LEN a$=1 AND a$="-" THEN LET numar=0: GO TO 240
230 LET numar=VAL a$
240 PRINT "Numarul introdus:";numar
250 STOP
```

```
*****
*   INTRODUCERE NUMAR INTREG   *
*****
```

Sa se scrie un program care sa
permita introducerea de la tasta
tura a unui intreg, din maxim n
caractere, cu interzicerea tasta
rii altor caractere

357 8

Numarul introdus:358

-56 89 3

Numarul introdus:-583

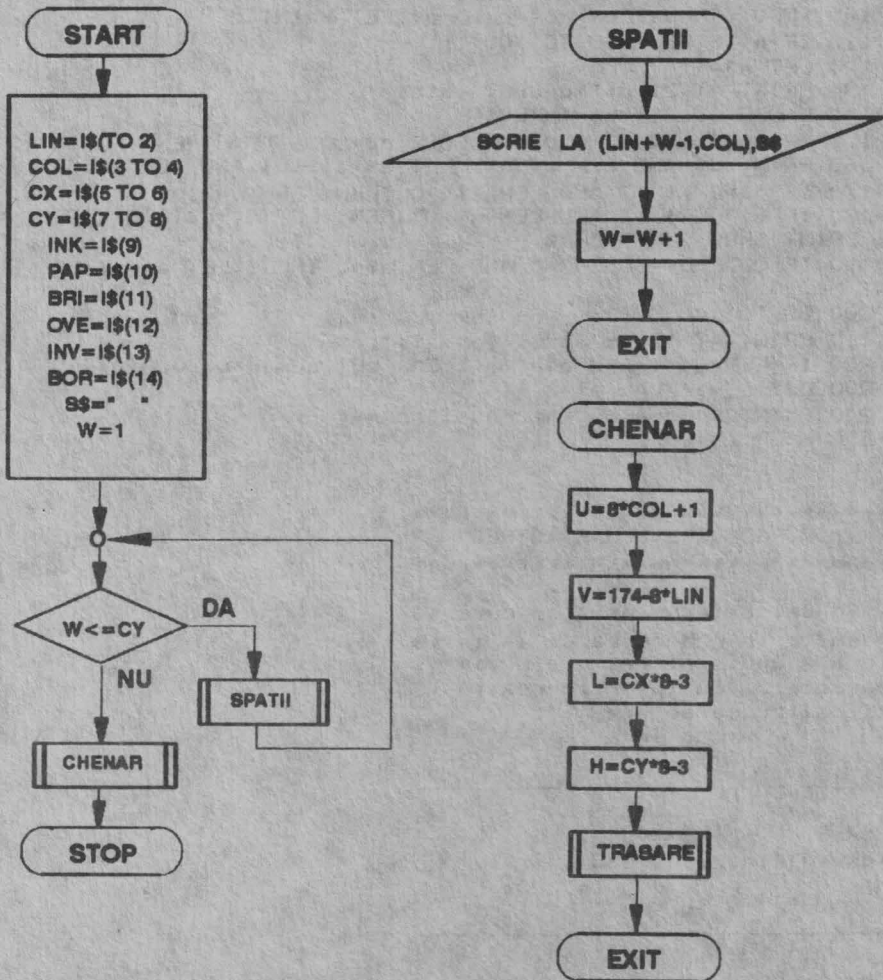
7.4 Se construiește un vector în felul următor:

$i\$ = "xyyccllipboIB"$, unde:

xx - colțul stînga sus (numărul liniei unde începe fereastra)
 yy - colțul stînga sus (numărul coloanei unde începe fereastra)
 cc - lățime fereastră (număr de caractere) $00 < cc < 31$
 ll - înălțime fereastră (număr de caractere) $00 < ll < 21$
 i - culoare cerneală (INK) $0 < i < 9$
 p - culoare hirtie (PAPER) $0 < p < 9$
 b - strălucire (BRIGHT) $0 < b < 1$ sau $b=8$
 o - suprapunere (OVER) $0 < o < 1$
 I - invers-video (INVERS) $0 < I < 1$
 B - border (BORDER) $0 < B < 1$

Acest vector este decodificat și se "desenează" o fereastră folosind parametrii aflați, apoi se trasează marginile ferestrei. În trasarea chenarului se ține cont că lin , col , cx , cy reprezintă număr de caractere iar un caracter are 8×8 pixeli; (u, v) - colțul stînga sus; l - lungime chenar; h - înălțime chenar.

Obs.: Subprogramul *Trasare* din schemă logică și pseudocod realizează desenarea ferestrei.



algoritmul 7.4. este:

```
citește i$
lin ← i$(TO 2)
col ← i$(3 TO 4)
cx ← i$(5 TO 6)
cy ← i$(7 TO 8)
ink ← i$(9)
pap ← i$(10)
bri ← i$(11)
ove ← i$(12)
inv ← i$(13)
bor ← i$(14)
s$ ← ''
```

pentru i=1,cx execută

```
s$ ← s$+''
```

pentru w=1,cy execută

```
scrie AT(lin+w-1,col),s$
```

sfârșit-pentru

sfârșit-pentru

cheamă CHENAR

stop

procedura CHENAR este:

```
u ← 8·col+1
```

```
v ← 174-8·lin
```

```
l ← cx·8-3
```

```
h ← cy·8-3
```

cheamă TRASARE

sfârșit-procedură

```
10 REM
20 REM *** WINDOWS ***
30 REM
40 BORDER 1: PAPER 1: INK 7: BRIGHT 1: CLS
50 PRINT
60 PRINT " *****"
70 PRINT " *           W I N D O W S           *"
80 PRINT " *****"
90 LET i$="00003222701101": GO SUB 390
100 LET i$="06042414160001": GO SUB 390: PAUSE 0
110 PRINT AT lin+1,col+1;"i$=06042414160001"
120 DATA "lin","col","cx","cy"
130 RESTORE 120
140 FOR i=1 TO 4
150 READ a$
160 PRINT AT lin+2+i,col+4;a$;"=";VAL i$(2*i-1 TO 2*i)
170 NEXT i
180 DATA "ink","pap","bri","ove","inv","bor"
190 RESTORE 180
200 FOR i=1 TO 6
210 READ a$
220 PRINT AT lin+6+i,col+4;a$;"=";VAL i$(8+i)
230 NEXT i
240 PAUSE 0
250 LET i$="06042414711101": GO SUB 390: PAUSE 0
260 LET i$="10081106521001": GO SUB 390
270 PRINT AT lin+2,col+2;"Alta";AT lin+3,col+1;"fereastră"
280 PAUSE 0
290 FOR i=0 TO 7
300 LET i$="0"+STR$ (i+1)+"0"+STR$ (i+1)+"2414"+STR$ (9)+
STR$ i+"1001": GO SUB 390
310 NEXT i
320 PRINT AT lin+6,col+8;"WINDOWS...": PAUSE 0
330 PRINT AT lin+8,col+1;"Liceul"
340 PRINT AT lin+9,col+4;"""GRIGORE MOISIL""
350 PRINT AT lin+11,col+4;"T I M I S O A R A"
360 BORDER 1: PAPER 1: INK 7: BRIGHT 1
370 STOP
```

```

380
390 REM *** W I N D O W ***
400 LET lin=VAL i$( TO 2): LET col=VAL i$(3 TO 4)
410 LET cx=VAL i$(5 TO 6): LET cy=VAL i$(7 TO 8)
420 LET ink=VAL i$(9): LET pap=VAL i$(10)
430 LET bri=VAL i$(11): LET ove=VAL i$(12)
440 LET inv=VAL i$(13): LET bor=VAL i$(14)
450 DIM s$(cx)
460 BORDER bor: PAPER pap: INK ink
470 BRIGHT bri: OVER ove: INVERSE inv
480 FOR w=1 TO cy: PRINT AT lin+w-1,col:s$: NEXT w
490 LET u=8*col+1: LET v=174-8*lin
500 LET l=8*cx-3: LET h=8*cy-3
510 OVER 0
520 PLOT u,v: DRAW 1,0: DRAW 0,-h: DRAW -1,0: DRAW 0,h
530 RETURN

```

7.5 Cele n nume vor fi memorate într-un tablou bidimensional n(n,18)$ iar ocupațiile corespunzătoare într-un tablou bidimensional o(n,10)$. După introducerea numărului de persoane se intră în prelucrarea tablourilor prin intermediul unui meniu.

Obs.: Procedura *Meniu* din pseudocod realizează afișarea meniului.

algoritmul 7.5. este:

```

citește n
cheamă MENU
[
  execută
  [
    citește opt
    ieșim-cînd (opt 1)or(opt=2)or(opt=3)or(opt=4)or(opt=5)
    cît-timp opt≠5 execută
  ]
  cheamă P
  sfîrșit-cît-timp
]
stop

```

procedura *P* este:

```

dacă opt=1
  atunci cheamă INTROD
  altfel
    [
      dacă opt=2
        atunci cheamă LIST
        altfel
          [
            dacă opt=3
              atunci cheamă LISTOCUP
              altfel cheamă LISTNR
            sfîrșit-dacă
          ]
      sfîrșit-dacă
    ]
  sfîrșit-dacă
cheamă MENU
[
  execută
  [
    citește opt
    ieșim-cînd (opt=1)or(opt=2)or(opt=3)or(opt=4)or(opt=5)
  ]
]
sfîrșit-procedură

```

procedura *INTROD* este:

```

[
  execută
  [
    citește nrord
    ieșim-cînd 1≤nrord≤n
    citește n$(nrord)
    citește o$(nrord)
  ]
]
sfîrșit-procedură

```

```

procedura LIST este:
  l ← 0
  pentru i=1,n execută
  [ cheamă LIST1
  [ sfârșit-pentru
  [ dacă l=0 atunci scrie 'nu a fost introdusă nici o persoană'
  [ sfârșit-dacă
  sfârșit-procedură

```

```

procedura LIST1 este:
  dacă n$(i)≠'' atunci l ← l+1
  scrie l,n$(i),o$(i)
  sfârșit-dacă
sfârșit-procedură

```

```

procedura LISTOCUP este:
  citește m$
  l ← 0
  pentru i=1,n execută
  [ cheamă LISTOCUP1
  [ sfârșit-pentru
  [ dacă l=0 atunci scrie 'nu există nimeni cu ocupația',m$
  [ sfârșit-dacă
  sfârșit-procedură

```

```

procedura LISTOCUP1 este:
  dacă n$(i)≠'' atunci
  [ dacă o$(i)=m$
  [ atunci l ← l+1
  [ scrie l,n$(i),o$(i)
  [ sfârșit-dacă
  sfârșit-dacă
sfârșit-procedură

```

```

procedura LISTNR este:
  l ← 0
  pentru i=1,n execută
  [ cheamă LISTNR1
  [ sfârșit-pentru
  [ dacă l=0 atunci scrie 'tablourile sînt pline'
  [ altfel scrie 'sînt disponibile',l,'poziții'
  [ sfârșit-dacă
  sfârșit-procedură

```

```

procedura LISTNR1 este:
  dacă n$(i)='' atunci l ← l+1
  scrie i
  sfârșit-dacă
sfârșit-procedură

```

```

70 PRINT " *****"
80 PRINT " *          SIMULARE FISIER          *"
90 PRINT " *****"
100 LET i$="00003222701101": GO SUB 1380
110 LET i$="05013016160001": GO SUB 1380
120 PRINT AT lin+1,col+1;" Sa se scrie un program ca-";AT
lin+2,col+1;"re memoreaza N nume de per-";AT lin+3,col+1;"s
oane si ocupatiile lor si";AT lin+4,col+1;"care poate afis
a la cerere:"
130 PRINT AT lin+6,col+1;" - lista persoanelor si";AT lin
+7,col+6;"ocupatiilor lor";AT lin+8,col+1;" - lista persoa
nelor cu";AT lin+9,col+6;"o anumita ocupatie"

```

```

140 PRINT AT lin+10,col+1;" -- lista numerelor de";AT lin+
11,col+6;"ordine disponibile la";AT lin+12,col+6;"un moment
dat"
150 LET i$="10032609521101": GO SUB 1380: PAUSE 200
160 INPUT "Numar maxim persoane:";n
170 LET n=INT n: IF n<1 OR n>99 THEN GO TO 160
175 DIM n$(n,18): DIM o$(n,10)
176 LET meniu=1000: LET apasa=900: LET curata=950
180 GO SUB meniu
185 IF opt=5 THEN STOP
190 GO TO 200*opt
200 REM *** introducere o persoana ***
210 DIM s$(18): DIM a$(384): PRINT AT 10,0;a$
220 INPUT "Numar de ordine:";nrord
230 IF nrord<1 OR nrord>n THEN GO TO 220
240 PRINT AT 10,0;"Numar de ordine:";nrord
250 INPUT "Numele:";n$(nrord)
260 IF n$(nrord)=s$ THEN GO TO 250
270 PRINT "Numele:";n$(nrord)
280 INPUT "Ocupatia:";o$(nrord)
290 IF o$(nrord)=s$( TO 10) THEN GO TO 280
300 PRINT "Ocupatia:";o$(nrord)
310 GO SUB apasa: GO SUB meniu: GO TO 185
400 REM *** listare nume si ocupatii ***
410 CLS : DEF FN x$(1)=(" " AND 1<10)+(STR$ 1+". ")
415 DIM s$(18)
420 PRINT " Listare totala"
430 LET l=0
440 FOR i=1 TO n
450 IF n$(i)=s$ THEN GO TO 480
460 LET l=l+1: GO SUB curata
470 PRINT FN x$(1);n$(i);o$(i)
480 NEXT i
490 IF l=0 THEN PRINT "'Nu a fost introdus nici un nume!"

500 GO SUB apasa: GO SUB meniu: GO TO 185
600 REM *** listare persoane cu o ocupatie ***
610 CLS : DIM m$(10): INPUT "Ocupatia:";m$
620 PRINT "Persoane cu ocupatia:";
624 FOR i=1 TO 10
625 IF m$(i)<>" " THEN PRINT FLASH 1;m$(i);
626 NEXT i: PRINT
630 LET l=0
640 FOR i=1 TO n
650 IF n$(i)=s$ OR o$(i)<>m$ THEN GO TO 680
660 LET l=l+1: GO SUB curata
670 PRINT FN x$(1);n$(i)
680 NEXT i
690 IF l=0 THEN PRINT "'Nu exista nici o persoana cu" "oc
upatia:";m$
700 GO SUB apasa: GO SUB meniu: GO TO 185
800 REM *** listare pozitii libere ***
810 CLS
820 PRINT " Numere de ordine libere"
830 LET l=0
840 FOR i=1 TO n
850 IF n$(i)=s$ THEN LET l=l+1: PRINT i;" ";
860 NEXT i
870 IF l=0 THEN PRINT "'tablourile sint pline": GO TO 890

880 PRINT "'sint disponibile ";l;" pozitii"

```



```

390 GO SUB apasa: GO SUB meniu: GO TO 185
900 PRINT AT 21,10;"Apasa o tasta": PAUSE 0
910 RETURN
950 IF 1/18=INT (1/18) THEN DIM a$(576): PRINT AT 2,0;a$:
PRINT AT 2,0;
960 RETURN
1000 REM *** M E N I U ***
1005 PAPER 0: BRIGHT 1: INK 7: BORDER 0: CLS
1010 PRINT AT 5,11;"M E N I U"
1020 PRINT AT 8,3;"1 - introducere o persoana"
1030 PRINT AT 10,3;"2 - lista persoane/ocupatii"
1040 PRINT AT 12,3;"3 - lista persoane cu o";AT 13,24;"ocupa
tie"
1060 PRINT AT 14,3;"4 - lista numere de ordine";AT 15,21;"di
sponibile"
1070 PRINT AT 16,3;"5 - STOP "
1080 PRINT AT 21,0;"Alege optiunea"
1090 PAUSE 0: LET t$=INKEY$
1100 IF t$<"1" OR t$>"5" THEN GO TO 1090
1110 LET opt=VAL t$
1120 RETURN
1380 REM *** W I N D O W ***
1390 LET lin=VAL i$( TO 2): LET col=VAL i$(3 TO 4)
1400 LET cx=VAL i$(5 TO 6): LET cy=VAL i$(7 TO 8)
1410 LET ink=VAL i$(9): LET pap=VAL i$(10)
1420 LET bri=VAL i$(11): LET ove=VAL i$(12)
1430 LET inv=VAL i$(13): LET bor=VAL i$(14)
1440 DIM s$(cx)
1450 BORDER bor: PAPER pap: INK ink
1455 BRIGHT bri: OVER ove: INVERSE inv
1460 FOR w=1 TO cy: PRINT AT lin+w-1,col;s$: NEXT w
1470 LET u=8*col+1: LET v=174-8*lin
1480 LET l=8*cx-3: LET h=8*cy-3
1490 OVER 0
1500 PLOT u,v: DRAW 1,0: DRAW 0,-h: DRAW -1,0: DRAW 0,h
1510 RETURN

```

```

*****
*          SIMULARE FISIER          *
*****

```

Numele:Corici Corina
Ocupatia:profesor

Numele:Manz Dorin
Ocupatia:profesor

Numele:Stan Ioana
Ocupatia:profesor

Numele:Dragoi Lucia
Ocupatia:operator

Numele:Simulescu Adriana
Ocupatia:profesor

Numele:Pantea Sorin
Ocupatia:elev

Numele:Serban Marinel
Ocupatia:profesor

Listare totala

1. Corici Corina profesor
 2. Manz Dorin profesor
 3. Stan Ioana profesor
 4. Dragoi Lucia operator
 5. Simulescu Adriana profesor
 6. Pantea Sorin elev
 7. Serban Marinel profesor
- Persoane cu ocupatia:profesor

1. Corici Corina
 2. Manz Dorin
 3. Stan Ioana
 4. Simulescu Adriana
 5. Serban Marinel
- Numere de ordine libere

1 2 3 4 5 6 8 9 12 15 16 17 19 20 21 23 24 25

sint disponibile 18 pozitii

Numele:Serban Rodica
Ocupatia:analist

Numele:Sipetan Camelia
Ocupatia:economist

Numele:Manz Mirela
Ocupatia:analist

Listare totala

1. Serban Rodica analist
2. Corici Corina profesor
3. Sipetan Camelia economist
4. Manz Dorin profesor
5. Stan Ioana profesor
6. Dragoi Lucia operator
7. Simulescu Adriana profesor
8. Pantea Sorin elev
9. Serban Marinel profesor
10. Manz Mirela analist

Persoane cu ocupatia:analist

1. Serban Rodica
2. Manz Mirela

Numere de ordine libere

1 2 4 5 6 9 12 15 16 17 19 20 21 23 25

sint disponibile 15 pozitii

7.6 In tabelul ASCII toate caracterele speciale din textul problemei(" ", ".", ",", ";", ":", ";", "(", ")", "!", "?") au codul mai mic decît 65, deci dacă a(i) < "A"$ (codul lui A este 65) în mod sigur am găsit un cuvînt. Dacă în plus acest caracter este ".", "!" sau "?" înseamnă că s-a încheiat o frază. Dacă a(i) > 65$ atunci intervin două cazuri:

a) litere mari - coduri 65-90
b) litere mici - coduri 97-122

In ambele cazuri, printr-o simplă scădere se determină numărul de ordine al literei (considerînd A - 1, B - 2, ..., Z - 26). In fine, numărul total de litere se poate calcula fie însumînd $l(i)$, $i=1, 26$ fie mai simplu: $n-cuv$ (numărul total de caractere din care se scade numărul cuvintelor, adică al caracterelor speciale).

algoritmul 7.6. este:

```

citește a$
n ← LEN(a$)
[ pentru i=1,26 execută
  li ← 0
sfîrșit-pentru
cuv ← 0
fra ← 0
[ pentru i=1,n execută
  cheamă P
sfîrșit-pentru
scrie a$
[ pentru i=1,26 execută
  pi ← 100 *  $\frac{l_i}{n-cuv}$ 
  scrie chr$(i+64), li, pi
sfîrșit-pentru
scrie cuv, fra
stop

```

procedura P este:

```

dacă a$(i) < 'A'
  atunci cuv ← cuv+1
  [ dacă (a$(i)='.' or (a$(i)='!' or (a$(i)='?'))
    atunci fra ← fra+1
  sfîrșit-dacă
  altfel [ dacă a$(i) ≤ 'z' atunci c ← code(a$(i))-64
    altfel c ← code(a$(i))-96
  sfîrșit-dacă
  lc ← lc+1
sfîrșit-dacă
sfîrșit-procedură

```

```

10 REM
20 REM *** ANALIZA TEXT ***
30 REM
40 BORDER 1: PAPER 1: INK 7: BRIGHT 1: CLS
50 PRINT
60 PRINT " *****"
70 PRINT " * ANALIZA TEXT * "
80 PRINT " *****"
90 LET i$="00003222701101": GO SUB 530
100 LET i$="05042615160001": GO SUB 530
110 PRINT AT lin+1,col+1;"Dindu-se un text in lim";AT li
n+2,col+1;"ba romana, corect, memo-";AT lin+3,col+1;"rat i
ntr-o variabila sir"
120 PRINT AT lin+4,col+1;"sa se scrie un program";AT li
n+5,col+1;"care:";AT lin+6,col+1;" - sa numere cite litere
"
130 PRINT AT lin+7,col+1;"de ficare fel se afla in";AT li
n+8,col+1;"text stabilind si procen";AT lin+9,col+1;"tabel
e corespunzatoare"
140 PRINT AT lin+10,col+1;" - sa numere cuvintele";AT l
in+11,col+1;"din text"

```

```

150 PRINT AT lin+12,col+1;" - sa numere cite fraze";AT lin+13,col+1;"sint in text"
160 PAUSE 0
170 LET i$="00003222701001": GO SUB 530
180 PRINT AT 10,10;"Introdu textul";AT 14,2;"textul NU va fi verificat!!!"
190 INPUT LINE a$
200 LET n=LEN a$: IF n<1 THEN GO TO 190
210 DIM l(26): DIM p(26)
220 LET cuv=0: LET fra=0
230 FOR i=1 TO n
240 IF a$(i)<"A" THEN LET cuv=cuv+1: GO TO 280
250 IF "A"<=a$(i) AND a$(i)<="Z" THEN LET c=CODE (a$(i))-64: GO TO 270
260 LET c=CODE (a$(i))-96
270 LET l(c)=l(c)+1: GO TO 290
280 IF a$(i)="." OR a$(i)="!" OR a$(i)="?" THEN LET fra=fra+1
290 NEXT i
300 CLS : PRINT "Textul:": PRINT
310 FOR i=1 TO LEN a$
320 PRINT a$(i);
330 IF a$(i)="." OR a$(i)="?" OR a$(i)="!" OR a$(i)="/" THEN PRINT
340 NEXT i
350 PRINT "litera aparitii procentaj (%)"
360 FOR i=1 TO 26
370 LET p(i)=100*l(i)/(n-cuv)
380 PRINT " ";CHR$(i+64);"(";CHR$(i+96);") ";
390 GO SUB 490: PRINT l(i);" "; GO SUB 460
400 PRINT (INT (p(i)*100))/100
410 NEXT i
420 PRINT "In text sint ";cuv;" cuvinte"
430 PRINT "Textul are ";fra;" fraze"
440 STOP
450
460 FOR j=1 TO 3-LEN (STR$ (INT (p(i))))
470 PRINT " ";
480 NEXT j: RETURN
490 FOR j=1 TO 3-LEN (STR$ (l(i)))
500 PRINT " ";
510 NEXT j: RETURN
520
530 REM *** W I N D O W ***
540 LET lin=VAL i$( TO 2): LET col=VAL i$(3 TO 4)
550 LET cx=VAL i$(5 TO 6): LET cy=VAL i$(7 TO 8)
560 LET ink=VAL i$(9): LET pap=VAL i$(10)
570 LET bri=VAL i$(11): LET ove=VAL i$(12)
580 LET inv=VAL i$(13): LET bor=VAL i$(14)
590 DIM s$(cx)
600 BORDER bor: PAPER pap: INK ink
610 BRIGHT bri: OVER ove: INVERSE inv
620 FOR w=1 TO cy: PRINT AT lin+w-1,col;s$: NEXT w
630 LET u=8*col+1: LET v=174-8*lin
640 LET l=8*cx-3: LET h=8*cy-3
650 OVER 0
660 PLOT u,v: DRAW 1,0: DRAW 0,-h: DRAW -1,0: DRAW 0,h
670 RETURN

```

 * ANALIZA TEXT *

Textul:

A fost odata ca-n povesti/
 A fost ca niciodata/
 Din rude mari imparatesti/
 O prea frumoasa fata.
 /
 Si era una la parinti/
 Si mindra-n toate cele/
 Cum e fecioara intre sfinti/
 Si luna intre stele.

litera aparitii procentaj (%)

A(a)	25	16.77
B(b)	0	0
C(c)	6	4.02
D(d)	5	3.35
E(e)	14	9.39
F(f)	6	4.02
G(g)	0	0
H(h)	0	0
I(i)	18	12.08
J(j)	0	0
K(k)	0	0
L(l)	4	2.68
M(m)	5	3.35
N(n)	11	7.38
O(o)	9	6.04
P(p)	4	2.68
Q(q)	0	0
R(r)	11	7.38
S(s)	10	6.71
T(t)	15	10.06
U(u)	5	3.35
V(v)	1	0.67
W(w)	0	0
X(x)	0	0
Y(y)	0	0
Z(z)	0	0

In text sint 38 cuvinte
 Textul are 2 fraze

7.7 Vom folosi un tablou cu două dimensiuni $n \times (n, 44)$ cu zonele: —
 $n(i, 1 \text{ TO } 28)$ - nume
 $n(i, 29 \text{ TO } 32)$ -media la matematică
 $n(i, 33 \text{ TO } 36)$ -media la fizică
 $n(i, 37 \text{ TO } 40)$ -media la română
 $n(i, 41 \text{ TO } 44)$ -media generală

Prelucrarea se realizează printr-o tehnică de tip meniuri. Singura problemă delicată este sortarea parametrizată (left-primul caracter al zonei cheie de sortare; right-ultimul caracter al zonei cheie de sortare). Parametrul left poate lua valorile 1,29,33,37,41 iar parametrul right corespunzător valorile 28,32,36,40,44. In soluție s-a folosit un algoritm de sor-

tare de tip SHELL, cu înjumătățirea intervalului pentru alegerea incrementilor. Bineînțeles, sortarea fiind unică, ea se face fie crescător fie descrescător, deci ordonarea alfabetică este inversă sortării după medii.

```

30 REM
40 REM *** SIMULARE FISIER ***
50 REM
55 BORDER 1: PAPER 1: INK 7: BRIGHT 1: CLS
60 GO SUB 600
110 LET i$="05003217160001": GO SUB 450
120 PRINT AT lin+1,col+1;" Programul permite:";AT lin+3,col+
3;" .memorarea a n nume de elevi";AT lin+4,col+4;" a mediilor 1
a"
130 PRINT AT lin+5,col+10;"matematica";AT lin+6,col+10;"fizi
ca";AT lin+7,col+10;"limba romana"
140 PRINT AT lin+8,col+4;"si a mediei generale";AT lin+9,col
+3;" .sortarea tabloului";AT lin+10,col+10;"alfabetic"
150 PRINT AT lin+11,col+10;"dupa una din medii"
160 PRINT AT lin+12,col+3;" .afisarea tabloului sortat";AT li
n+13,col+3;" .modificarea datelor unui";AT lin+14,col+4;"elev"

170 PAUSE 0
180 DEF FN h$(x)=((" "+STR$(100*x)) AND x<10)+("1000" AND x
=10)
190 DEF FN w$(1,x)=" "+n$(1,x TO x+1)+"."+n$(1,x+2 TO x+3)
200 LET meniu=7000: LET meniusort=6000
205 DIM s$(28): LET m$=""
210 INPUT "Numar maxim elevi:";n
215 LET l=1: DIM n$(ABS (INT n),44)
220 GO SUB meniu
230 IF opt=5 THEN STOP
240 GO SUB 1000*opt: GO SUB meniu: GO TO 230
400 REM *** scriere cu format ***
410 FOR j=1 TO 3-LEN (STR$ (INT (p(i))))
411 PRINT " ";
412 NEXT j: RETURN
420 FOR j=1 TO 3-LEN (STR$ (l(i)))
421 PRINT " ";
422 NEXT j: RETURN
450 REM *** W I N D O W ***
460 LET lin=VAL i$(3 TO 2): LET col=VAL i$(3 TO 4)
470 LET cx=VAL i$(5 TO 6): LET cy=VAL i$(7 TO 8)
480 LET ink=VAL i$(9): LET pap=VAL i$(10)
490 LET bri=VAL i$(11): LET ove=VAL i$(12)
500 LET inv=VAL i$(13): LET bor=VAL i$(14)
510 DIM s$(cx)
520 BORDER bor: PAPER pap: INK ink
525 BRIGHT bri: OVER ove: INVERSE inv
530 FOR w=1 TO cy: PRINT AT lin+w-1,col;s$: NEXT w
540 LET u=8*col+1: LET v=174-8*lin
550 LET lenght=8*cx-3: LET high=8*cy-3
560 OVER 0
570 PLOT u,v: DRAW lenght,0: DRAW 0,-high
575 DRAW -lenght,0: DRAW 0,high
580 RETURN
600 CLS : PRINT

```

```

610 PRINT " *****"
620 PRINT " *          SIMULARE FISIER          * "
630 PRINT " *****"
640 LET i$="00003205021101": GO SUB 450
650 RETURN
1000 REM *** introducere date ***
1005 LET i$="05003217160001": GO SUB 450
1010 IF l>n THEN PRINT AT lin+2,col+2;"TABLOU PLIN!!!": GO T
0 1040
1015 INPUT "Numele: ";n$(1, TO 28)
1020 IF l<n AND n$(1,1)<>" " THEN GO SUB 1500: LET m$="": G
0 TO 1010
1040 PAUSE 100: RETURN
1500 LET i$="05003217160001": GO SUB 450
1505 OVER 1: PRINT AT lin+1,0;
1510 PRINT " Nume: ";n$(1, TO 26)
1515 INPUT "nota matematica: ";mat
1520 IF mat<1 OR mat>10 THEN GO TO 1515
1525 PRINT " matematica: ",mat
1530 INPUT "nota fizica: ";fiz
1535 IF fiz<1 OR fiz>10 THEN GO TO 1530
1540 PRINT " fizica: ",fiz
1545 INPUT "nota romana: ";rom
1550 IF rom<1 OR rom>10 THEN GO TO 1545
1555 PRINT " romana: ",rom
1560 LET med=(mat+fiz+rom)/3
1565 PRINT " media: ";med
1570 LET n$(1,29 TO 32)=FN h$(mat)
1575 LET n$(1,33 TO 36)=FN h$(fiz)
1580 LET n$(1,37 TO 40)=FN h$(rom)
1585 LET n$(1,41 TO 44)=FN h$(med)
1590 LET l=l+1
1595 OVER 0
1600 RETURN
2000 REM *** SORTARI ***
2010 IF l=0 THEN PRINT AT 21,0; FLASH 1;"TABLOU GOL": PAUSE
0: RETURN
2015 GO SUB meniusort
2020 IF opts=6 THEN PAUSE 100: RETURN
2030 IF opts=1 THEN LET left=1: LET right=28: LET m$="ALFABE
T": GO TO 2080
2040 IF opts=2 THEN LET left=29: LET right=32: LET m$="MATEM
ATICA": GO TO 2080
2050 IF opts=3 THEN LET left=33: LET right=36: LET m$="FIZIC
A": GO TO 2080
2060 IF opts=4 THEN LET left=37: LET right=40: LET m$="ROMAN
A": GO TO 2080
2070 LET left=41: LET right=44: LET m$="MEDIE": GO TO 2080
2080 LET m=l-1
2090 LET m=INT (m/2)
2100 IF m=0 THEN GO TO 2210
2110 LET j=1: LET t=l-1-m
2120 LET p=j
2130 LET v=p+m
2140 IF n$(p,left TO right)>=n$(v,left TO right) THEN GO TO
2180
2150 LET u$=n$(p): LET n$(p)=n$(v): LET n$(v)=u$
2160 LET p=p-m
2170 IF p>0 THEN GO TO 2130
2180 LET j=j+1
2190 IF j>t THEN GO TO 2090

```

```

2200 GO TO 2120
2210 IF opts<>1 THEN GO TO 2265
2220 FOR i=1 TO INT ((1-1)/2)
2230 LET u$=n$(i)
2240 LET n$(i)=n$(1-i)
2250 LET n$(1-i)=u$
2260 NEXT i
2265 PRINT AT 20,1; FLASH 1;"Gata sortarea dupa ";m$
2266 PAUSE 0
2270 GO TO 2010
3000 REM *** LISTARE ***
3010 DEF FN p$(x)=((" "+STR$ x) AND x<10)+(STR$ x AND x>=10)+
" "
3015 IF l=1 THEN PRINT AT 20,10; INVERSE 1;"Tablou vid": PAU
SE 0; RETURN
3020 CLS : PRINT " Lista ordonata dupa "; INVERSE 1;m$
3030 FOR i=1 TO l-1
3040 IF i/11=INT (i/11) THEN DIM a$(640): PAUSE 0: PRINT A
T 2,0;a$: PRINT AT 2,0;
3050 PRINT FN p$(i);n$(i, TO 27)
3060 PRINT FN w$(i,29);FN w$(i,33);FN w$(i,37);FN w$(i,41)
3070 NEXT i
3080 PAUSE 0; GO SUB 600; RETURN
4000 REM *** MODIFICARE DATE UN ELEV ***
4010 LET i$="05003217160001": GO SUB 450
4020 INPUT "Numar ordine:";nrord
4030 IF nrord<1 OR nrord>l-1 THEN RETURN
4040 LET nrord=INT nrord: OVER 1: PRINT AT lin+1,col+1;
4050 PRINT "Numar ordine:";nrord
4060 PRINT " Nume:";n$(nrord, TO 27)
4070 PRINT " Matematica:";FN w$(nrord,29)
4080 PRINT " Fizica:";FN w$(nrord,33)
4090 PRINT " Romana:";FN w$(nrord,37)
4100 INPUT "Numele:";n$(nrord, TO 28)
4110 LET l1=l: LET l=nrord: GO SUB 1510: LET l=l1
4120 PAUSE 0; OVER 0: LET m$="": RETURN
6000 REM *** MENIUSORT ***
6010 LET i$="05003217160001": GO SUB 450
6020 PRINT AT 5,12;"MENIU SORT"
6030 PRINT AT 8,5;"1 - sortare alfabetica"
6040 PRINT AT 10,5;"2 - sortare matematica"
6050 PRINT AT 12,5;"3 - sortare fizica"
6060 PRINT AT 14,5;"4 - sortare romana"
6070 PRINT AT 16,5;"5 - sortare medie"
6080 PRINT AT 18,5;"6 - revenire MENIU"
6090 PRINT AT 21,8; INVERSE 1;"Alege optiunea:"
6100 PAUSE 0: LET t$=INKEY$
6110 IF t$<"1" OR t$>"6" THEN GO TO 6100
6120 LET opts=VAL t$
6125 PRINT AT 8+(opts-1)*2,5; FLASH 1;t$
6130 RETURN
7000 REM *** M E N I U ***
7010 LET i$="05003217160001": GO SUB 450
7020 PRINT AT lin+2,col+12;"M E N I U"
7030 PRINT AT lin+5,col+7;"1 - introducere date"
7040 PRINT AT lin+7,col+7;"2 - sortari"
7050 PRINT AT lin+9,col+7;"3 - listare"
7060 PRINT AT lin+11,col+7;"4 - modificare"
7070 PRINT AT lin+13,col+7;"5 - STOP "
7080 PRINT AT 21,8; INVERSE 1;"Alege optiunea:"
7090 PAUSE 0: LET t$=INKEY$

```


7100 IF t\$<"1" OR t\$>"5" THEN GO TO 7090
7110 LET opt=VAL t\$
7120 RETURN

* SIMULARE FISIER *

Nume:Serban Mihaela

matematica:9.5
fizica:8.88
romana:10

media:9.46

Nume:Sperneac Paul

matematica:10
fizica:9.66
romana:7

media:8.8866667

Nume:Vinereanu Tudor

matematica:10
fizica:10
romana:9.12

media:9.7066667

Nume:Pacuraru Mihai

matematica:10
fizica:9
romana:10

media:9.6666667

Nume:Lupas Dan

matematica:8.88
fizica:7.9
romana:9.5

media:8.76

Nume:Neagoe Ana-Maria

matematica:9.67
fizica:9.84
romana:9.42

media:9.6433333

Lista ordonata dupa

1. Serban Mihaela	9.50	8.88	10.00	9.46
2. Sperneac Paul	10.00	9.66	7.00	8.88
3. Vinereanu Tudor	10.00	10.00	9.12	9.70
4. Pacuraru Mihai	10.00	9.00	10.00	9.66
5. Lupas Dan	8.88	7.90	9.50	8.76
6. Neagoe Ana-Maria	9.67	9.84	9.42	9.64

Lista ordonata dupa ALFABET

1. Lupas Dan	8.88	7.90	9.50	8.76
2. Neagoe Ana-Maria	9.67	9.84	9.42	9.64
3. Pacuraru Mihai	10.00	9.00	10.00	9.66
4. Serban Mihaela	9.50	8.88	10.00	9.46
5. Sperneac Paul	10.00	9.66	7.00	8.88
6. Vinereanu Tudor	10.00	10.00	9.12	9.70

Lista ordonata dupa MEDIE

1. Vinereanu Tudor	10.00	10.00	9.12	9.70
2. Pacuraru Mihai	10.00	9.00	10.00	9.66
3. Neagoe Ana-Maria	9.67	9.84	9.42	9.64
4. Serban Mihaela	9.50	8.88	10.00	9.46
5. Sperneac Paul	10.00	9.66	7.00	8.88
6. Lupas Dan	8.88	7.90	9.50	8.76

Lista ordonata dupa ROMANA

1. Serban Mihaela	9.50	8.88	10.00	9.46
2. Pacuraru Mihai	10.00	9.00	10.00	9.66
3. Lupas Dan	8.88	7.90	9.50	8.76
4. Neagoe Ana-Maria	9.67	9.84	9.42	9.64
5. Vinereanu Tudor	10.00	10.00	9.12	9.70
6. Sperneac Paul	10.00	9.66	7.00	8.88

Lista ordonata dupa MATEMATICA

1. Pacuraru Mihai	10.00	9.00	10.00	9.66
2. Sperneac Paul	10.00	9.66	7.00	8.88
3. Vinereanu Tudor	10.00	10.00	9.12	9.70
4. Neagoe Ana-Maria	9.67	9.84	9.42	9.64
5. Serban Mihaela	9.50	8.88	10.00	9.46
6. Lupas Dan	8.88	7.90	9.50	8.76

Lista ordonata dupa FIZICA

1. Vinereanu Tudor	10.00	10.00	9.12	9.70
2. Neagoe Ana-Maria	9.67	9.84	9.42	9.64
3. Sperneac Paul	10.00	9.66	7.00	8.88
4. Pacuraru Mihai	10.00	9.00	10.00	9.66
5. Serban Mihaela	9.50	8.88	10.00	9.46
6. Lupas Dan	8.88	7.90	9.50	8.76

Numar ordine:5

Nume:Serban Mihaela

Matematica: 9.50

Fizica: 8.88

Romana: 10.00

Nume:Serban Mihaela

matematica:9.63

fizica:8.95

romana:10

media:9.5266667

Lista ordonata dupa MEDIE

1. Vinereanu Tudor	10.00	10.00	9.12	9.70
2. Pacurararu Mihai	10.00	9.00	10.00	9.66
3. Neagoe Ana-Maria	9.67	9.84	9.42	9.64
4. Serban Mihaela	9.63	8.95	10.00	9.52
5. Sperneac Paul	10.00	9.66	7.00	8.88
6. Lupas Dan	8.80	7.90	9.50	8.76

8.1.1 Numerele se consideră ca șiruri de cifre binare de lungime n . Primul șir considerat este cel cu toate elementele nule. După tipărire se face adunarea cu 1 în felul următor: se parcurge șirul de la ultimul element către primul; atita vreme cât se întâlnesc elemente egale cu 1, acestora li se atribuie valoarea 0, iar la întâlnirea primului element nul acesta ia valoarea 1, elementele dinaintea lui rămânând neschimbate. Un contor indică momentul în care au fost obținute toate numerele binare cerute. În soluția propusă la scrierea fiecărui șir s-au sărit zerourile ne semnificative.

```

20 PRINT "      * AFISARE NUMERE BINARE *"
40 INPUT "Introduceti numarul N "; N
50 PRINT "" NUMARUL N=";N: PAUSE 0
60 DIM X(N): CLS
70 PRINT "" NUMERELE BINARE DE LA"
80 PRINT "      0 LA ";2*N-1;" SINT:"
90 PRINT : PRINT "      0"
100
110 REM Generarea succesivului
120
130 FOR K=1 TO 2*N-1
140   LET L=N
150   IF L=0 THEN GO TO 180
160   IF X(L)=1 THEN LET X(L)=0: LET L=L-1: GO TO 150
170   LET X(L)=1
180   LET SW=0: PRINT : PRINT "      ";
190   FOR I=1 TO N
200     IF SW=0 THEN GO TO 220
210     PRINT X(I);" ";: GO TO 240
220     IF X(I)=0 THEN GO TO 240
230     PRINT X(I);" ";: LET SW=1
240   NEXT I
250   PRINT
260 NEXT K
270 PAUSE 0: CLS
280 STOP

```

* AFISARE NUMERE BINARE *

NUMARUL N=4

NUMERELE BINARE DE LA
0 LA 15 SINT:

0

1

1 0

1 1

1 0 0

1 0 1

1 1 0

1 1 1

* AFISARE NUMERE BINARE *

NUMARUL N=4

NUMERELE BINARE DE LA
0 LA 15 SINT:

0
1
1 0
1 1
1 0 0
1 0 1
1 1 0
1 1 1
1 0 0 0
1 0 0 1
1 0 1 0
1 0 1 1
1 1 0 0
1 1 0 1
1 1 1 0
1 1 1 1

8.1.3 Şirul Y ce conţine reprezentarea numarului în baza b se parcurge de la ultimul element începînd. Dacă la sfîrşit sînt elemente egale cu $b-1$ (adică cu ultimul element din şirul cifrelor bazei b), acestea se înlocuiesc cu zero (primul element în şirul cifrelor). Primul element diferit de $b-1$ trebuie căutat în şirul X al cifrelor şi înlocuit cu următorul element din X (înlocuire ce corespunde mării cu 1 a cifrei respective). Dacă toate elementele şirului Y sînt egale cu $b-1$, atunci se scrie "1" înainte de a tipări şirul (care are, în acest caz, toate elementele nule cînd s-a încheiat parcurgerea sa).

```
10 CLS : LET z1=1: LET z2=30: GO SUB 400
20 PRINT " * ADUNARE CU 1 INTR-O BAZA B *": GO SUB 400
30 INPUT "Introduceti baza b "; b
40 PRINT " BAZA=";B: DIM X$(B)
50 PRINT " CIFRELE BAZEI ";B;" SINT:"
60 PRINT " ;"
```

```

70 FOR i=1 TO b
80   INPUT x$(i): PRINT x$(i);" ";
90   IF i=14 THEN PRINT : PRINT "   ";
100 NEXT i
110 PRINT
120 INPUT "Dati nr de cifre al numarului n ";n: DIM y$(n)
130 PRINT #0; "Introduceti cifrele numarului n ": PAUSE 50
140 PRINT "/".   NUMARUL ESTE:";
150 FOR j=1 TO n
160   INPUT y$(j): PRINT y$(j);
170 NEXT j
180 PRINT
190
200 REM Calcularea succesorelui
210
220 LET k=0
230 FOR j=n TO 1 STEP -1
240   IF y$(j)=x$(b) THEN LET y$(j)=x$(1): GO TO 270
250   LET k=1: LET q#=y$(j): GO SUB 350
260   LET y$(j)=x$(i+1): LET j=1
270 NEXT j
280 PRINT "/ "   SUCCESORUL LUI ESTE:";
290 IF k=0 THEN PRINT "1";
300 FOR j=1 TO n
310   PRINT y$(j);
320 NEXT j
330 PAUSE 0: CLS : STOP
340
350 LET i=1
360 IF q#=x$(i) THEN GO TO 380
370 LET i=i+1: GO TO 360
380 RETURN
390
400 FOR v=1 TO z1: PRINT " ";: NEXT v
410 FOR v=1 TO z2: PRINT "*";: NEXT v: PRINT
420 RETURN

```

```

*****
* ADUNARE CU 1 INTR-O BAZA B *
*****

```

BAZA=3

CIFRELE BAZEI 3 SINT:

0 1 2

NUMARUL ESTE:20212

SUCCESORUL LUI ESTE:20220

8.1.4 O variabilă u , inițializată cu zero, va conține eventuala cifră de transport. Șirurile Y și Z , ce conțin cifrele celor două numere, se parcurg de la ultimul element către primul. Pentru a afla valoarea corespunzătoare a unui element al acestor șiruri se procedează ca în problema precedentă.

```

30 INPUT "Introduceti baza b ";b
40 DIM x$(b): PRINT "" BAZA = ";b
50 PRINT "" CIFRELE BAZEI ";B;" SINT:"""
60 PRINT ""
70 FOR i=1 TO b
80 INPUT x$(i): PRINT x$(i);" ";: IF i=14 THEN PRINT
90 NEXT i
100 INPUT "Numerele se dau ca siruri de cifre de lungime ";n
110 DIM y$(n): DIM z$(n): DIM u$(n)
120 PRINT "" NUMERELE SINT:"""
130 FOR i=1 TO n
140 INPUT y$(i): PRINT y$(i);
150 NEXT i
160 PRINT ""
170 FOR i=1 TO n
180 INPUT z$(i): PRINT z$(i);
190 NEXT i
200
210 REM Se calculeaza suma numerelor date
220 LET u=0
230 FOR j=n TO 1 STEP -1
240 LET q%=y$(j): GO SUB 390: LET s=i-1
250 LET q%=z$(j): GO SUB 390: LET s=s+i-1+u
260 IF s>=b THEN GO TO 280
270 LET u=0: GO TO 290
280 LET u=1: LET s=s-b
290 LET u$(j)=x$(s+1)
300 NEXT j
310 PRINT "" SUMA ESTE:"""
320 IF u=1 THEN PRINT "1";
330 FOR j=1 TO n
340 PRINT u$(j);
350 NEXT j
360 PAUSE 0: CLS
370 GO TO 30
380
390 LET i=1
400 IF q%=x$(i) THEN RETURN
410 LET i=i+1: GO TO 400
420 RETURN

```

BAZA = 2

CIFRELE BAZEI 2 SINT:

0 1

NUMERELE SINT:

1101

1001

SUMA ESTE:

10110

8.1.5 Trebuie, în primul rând, determinat care dintre cele două șiruri corespunde numărului mai mare. Dacă acesta este al doilea șir, se tipărește caracterul "-" și se inversează elementele celor două șiruri. Variabila u, inițializată cu zero, va avea valoarea 1 dacă este necesar împrumutul.

algoritmul 8.1.5 este:

```

citește n,b
pentru i=1,n execută
  citește x$(i)
sfârșit-pentru
pentru i=1,n execută
  citește y$(i)
sfârșit-pentru
pentru i=1,n execută
  citește z$(i)
sfârșit-pentru
pentru i=1,n execută
  cheamă P
sfârșit-pentru
u ← 0
pentru j=n,1,-1 execută
  cheamă Q
sfârșit-pentru
dacă s=1 atunci scrie '-'
sfârșit-dacă
pentru j=1,n execută
  scrie u$(j)
sfârșit-pentru
stop

```

procedura P este:

```

q$ ← y$(j)
cheamă P1
l1 ← i-1
q$ ← z$(j)
cheamă P1
l2 ← i-1
dacă l1≠l2
  atunci
    dacă l1<l2
      atunci s ← 1
      altfel s ← 0
    sfârșit-dacă
    j ← n
  sfârșit-dacă
sfârșit-procedură

```

procedura P1 este:

```

i ← 1
cît-timp x$(i)≠q$ execută
  i ← i+1
sfârșit-cît-timp
sfârșit-procedură

```

procedura Q este:

```

q$ ← y$(j)
cheamă P1
d1 ← i-1
q$ ← z$(j)
cheamă P1
d2 ← i-1
dacă s≠1 atunci d ← d1-d2-u
  altfel d ← d2-d1-u
sfârșit-dacă
dacă d≥0 atunci u ← 0
  altfel u ← 1
  d ← b+d
sfârșit-dacă
u$(j) ← x$(d+1)
sfârșit-procedură

```

```

10 CLS : LET z1=4: LET z2=24: GO SUB 590
20 PRINT " * DIFERENTA A 2 NUMERE *"
30 PRINT " * IN BAZA B *": GO SUB 590
40 INPUT "Introduceti baza b " :b
50 PRINT "" BAZA=";B: DIM x$(b)
60 PRINT "" CIFRELE BAZEI ";b;" SINT:"" ";
70 FOR i=1 TO b
80 INPUT x$(i): PRINT x$(i);" ";
90 NEXT i
100 INPUT ""Numerele se dau ca siruri de lungime n=";n
110 DIM y$(n): DIM z$(n): DIM u$(n)
120 PRINT "" NUMERELE SINT:"" ";

```

```

130 FOR i=1 TO n
140   INPUT y$(i): PRINT y$(i);
150 NEXT i
160 PRINT "  ";
170 FOR i=1 TO n
180   INPUT z$(i): PRINT z$(i);
190 NEXT i
200 PRINT
210
220 REM Calcularea diferentei
230
240 FOR j=1 TO n
250   LET q#=y$(j): GO SUB 530: LET l1=i-1
260   LET q#=z$(j): GO SUB 530: LET l2=i-1
270   IF l1=l2 THEN GO TO 310
280   IF l1<l2 THEN LET s=1: GO TO 300
290   LET s=0
300   LET j=n
310 NEXT j
320 LET u=0
330 FOR j=n TO 1 STEP -1
340   LET q#=y$(j): GO SUB 530: LET d1=i-1
350   LET q#=z$(j): GO SUB 530: LET d2=i-1
360   IF s=0 THEN LET d=d1-d2-u: GO TO 380
370   LET d=d2-d1-u
380   IF d>=0 THEN LET u=0: GO TO 400
390   LET u=1: LET d=d+b
400   LET u$(j)=x$(d+1)
410 NEXT j
420 PRINT : PRINT "  DIFERENTA ESTE=";
430 IF s=1 THEN PRINT "-";
440 LET j=1
450 IF u$(j)<>x$(1) THEN GO TO 470
460 LET j=j+1: GO TO 450
470 FOR l=j TO n
480   PRINT u$(l);
490 NEXT l
500 PAUSE 0: CLS
510 STOP
520
530 REM Cautarea unei cifre in sirul cifrelor bazei
540 LET i=1
550 IF q#=x$(i) THEN GO TO 570
560 LET i=i+1: GO TO 550
570 RETURN
580
590 FOR v=1 TO z1: PRINT " ";: NEXT v
600 FOR v=1 TO z2: PRINT "*";: NEXT v: PRINT
610 RETURN

```

```

*****
* DIFERENTA A 2 NUMERE *
*       IN BAZA B       *
*****

```

BAZA=2

CIFRELE BAZEI 2 SINT:
0 1

NUMERELE SINT:

10011
11001

DIFERENTA ESTE=-110

BAZA=16

CIFRELE BAZEI 16 SINT:

0 1 2 3 4 5 6 7 8 9 A B C D E F

NUMERELE SINT:

A2C51
9FEFE

DIFERENTA ESTE=2D53

8.2.1 Valorile aproximative obținute pentru radical nu trebuie reținute, ci putem folosi doar două variabile: x care păstrează valoarea obținută în etapa anterioară (inițializată cu 1) și y în care se calculează valoarea obținută în etapa în curs, adică: $y=(a/x+x)/2$.

Algoritmul se încheie atunci când modulul diferenței dintre x și y este mai mic decât precizia cerută.

```
50 INPUT "Introduceti numarul a= ";a
60 PRINT "' ' "NUMARUL a=";a
70 LET T=SQR (a)
80 PRINT "' "RADICALUL CALCULAT CU AJUTORUL "
90 PRINT "FUNCTIEI SQR=";T
100 PAUSE 0: CLS
110 PRINT "'CALCULAM VALOAREA APROXIMATIVA"
120 PRINT "FOLOSIND FORMULA:"
130 PRINT "'x(n)=(a/x(n-1)+x(n-1))/2"
140 PRINT "'CONSIDERIND x(1)=1"
150 INPUT "Introduceti precizia eps= "; EPS
160 PRINT "' "PRECIZIA CERUTA ESTE=";EPS
170 LET X=1
180 PRINT "' "VALORILE APROXIMATIVE DUPA FIE-"
190 PRINT "CARE ITERATIE SINT:""'
200 LET Y=(a/X+X)/2
210 IF ABS (Y-X)<EPS THEN GO TO 230
220 LET X=Y: PRINT X: GO TO 200
230 LET Z3=INT (100*Y): LET Y=Z3/100
240 LET Z2=Z2+1
250 PRINT "' ' VALOAREA FINALA=";Y
260 PAUSE 0: CLS : STOP
```

NUMARUL a=120

RADICALUL CALCULAT CU AJUTORUL
FUNCTIEI SQR=10.954451

CALCULAM VALOAREA APROXIMATIVA
FOLOSIND FORMULA:

$x(n)=(a/x(n-1)+x(n-1))/2$

CONSIDERIND $x(1)=1$

PRECIZIA CERUTA ESTE=.01

VALORILE APROXIMATIVE DUPA FIECARE ITERATIE SINT:

60.5
31.241736
17.541376
12.191172
11.01718
10.95463

VALOAREA FINALA=10.95

8.2.2 Considerind un număr de trei cifre $n=n_1n_2n_3$ și o cifră zecimală nenulă, se obține următorul număr de forma cerută
 $m=an_1n_2n_3a$, deci $m=10n+a+10000a=10n+10001a$

Pentru a obține cel mai mic număr cu proprietatea cerută, se variază cifra a de la 1 la 9 și, pentru fiecare valoare a cifrei a , numărul n de la 0 la 999. Pentru fiecare cifră a și fiecare număr n , se generează numărul m conform egalității anterioare și se testează dacă el îndeplinește condiția cerută. La întâlnirea primului număr m cu proprietatea cerută, algoritmul se încheie.

```
10 CLS : LET z1=4: LET z2=24: GO SUB 240
20 PRINT " * GENERAREA UNUI NUMAR *"
30 GO SUB 240
40 PRINT "" PROGRAMUL GENEREAZA CEL MAI";
50 PRINT "MIC NUMAR DE 5 CIFRE CARE INCEPE"
60 PRINT "SI SE TERMINA CU ACEEASI CIFRA"
70 PRINT "SI DA RESTUL R IN IMPARTIREA CUNUMARUL P."
80 INPUT "INTRODUCETI R=";R;" P=";P
90 IF R>=P THEN GO TO 80
100 PRINT "" P=";P;" R=";R
110 REM Se genereaza toate nr de 5 cifre cu prima cifra
120 REM egala cu ultima si apoi se testeaza estul
130 REM IMPARTIRII LOR LA P.
140 FOR A=1 TO 9
150 FOR N=0 TO 999
160 LET M=10*N+10001*A: LET C=INT (M/P)
170 LET R1=M-P*C
180 IF R1<>R THEN GO TO 200
190 PRINT : PRINT "NUMARUL CAUTAT ESTE:";M: GO TO 220
200 NEXT N
210 NEXT A
220 PAUSE 0: CLS : STOP
230
240 FOR V=1 TO Z1: PRINT " ";: NEXT V
250 FOR V=1 TO Z2: PRINT " *";: NEXT V
260 PRINT : RETURN
```



```
*****
* GENERAREA UNUI NUMAR *
*****
```

PROGRAMUL GENEREAZA CEL MAIMIC NUMAR DE 5 CIFRE CARE
INCEPE SI SE TERMINA CU ACEEASI CIFRA
SI DA RESTUL R IN IMPARTIREA CUNUMARUL P.

P=732 R=25

NUMARUL CAUTAT ESTE:13201

8.2.3 Se caută prima cifră a numărului care este mai mare decât următoarea și se elimină acea cifră. Dacă cifrele sînt în ordine crescătoare, atunci se elimină ultima cifră.

```
10 LET Z1=4: LET Z2=21: GO SUB 290
20 PRINT " * NUMAR MINIM *"
30 PRINT " * ELIMININD O CIFRA *"
40 GO SUB 290
50 PRINT " " NUMAR MINIM CIFRA ELIM"
60 PRINT " : DIM A(100)
70 INPUT "INTRODU NR N ";N
80 LET VV=N: GO SUB 260
90 LET K=1
100 LET A(K)=N-10*INT (N/10): LET N=INT (N/10): LET K=K+1
110 IF N>=10 THEN GO TO 100
120 LET A(K)=N: LET M=1
130 FOR I=K TO 2 STEP -1
140 IF A(I)<=A(I-1) THEN GO TO 160
150 LET M=I: LET I=2
160 NEXT I
170 LET Q=0: LET R=1
180 FOR I=1 TO K
190 IF I=M THEN GO TO 210
200 LET Q=Q+A(I)*10^(R-1): LET R=R+1
210 NEXT I
220 LET VV=Q: GO SUB 260
230 LET VV=A(M): GO SUB 260
240 PRINT : GO TO 70
250
260 FOR Z=1 TO 8-LEN STR$ (VV): PRINT " ";: NEXT Z
270 PRINT VV;: RETURN
280
290 FOR V=1 TO Z1: PRINT " ";: NEXT V
300 FOR V=1 TO Z2: PRINT " *";
310 NEXT V: PRINT : RETURN
```

```
*****
* NUMAR MINIM *
* ELIMININD O CIFRA *
*****
```

NUMAR MINIM CIFRA ELIM

13	1	3
576	56	7
2536	236	5
133657	13357	6
256789	25678	9
37509	3509	7
8965	865	9
531	31	5
21	1	2

8.2.4 $n=100a+10b+c+100b+10c+a+100c+10a+b=111(a+b+c)$, a, b, c fiind cifre zecimale, rezultă că $a+b+c$ este cuprins între 0 și 27. Însă, n trebuind să aibă patru cifre, $a+b+c$ se consideră cel puțin 10. Notînd $d=a+b+c$, obținem condiția ca d să ia valori între 10 și 27. Deci, pentru fiecare valoare a lui d cuprinsă între limitele de mai sus, se formează numărul $n=111d$ și apoi se caută cifrele a, b, c care permit scrierea cerută.

```

10 CLS : LET z1=4: LET z2=21: GO SUB 130
20 PRINT " * NUMERELE DE FORMA *"
30 PRINT " * ABC+BCA+CAB *"
40 GO SUB 130
50 PRINT " " NUMERELE DE 4 CIFRE CE PER-";
60 PRINT " MIT Scrierea CERUTA:"
70 FOR D=10 TO 27
80 GO SUB 170
90 LET N=111*D
100 PRINT " "; N; "="; A; B; C; "+"; B; C; A; "+"; C; A; B
110 NEXT D
120 PAUSE 0: CLS : STOP
130 FOR V=1 TO Z1: PRINT " ";: NEXT V
140 FOR V=1 TO Z2: PRINT " *";: NEXT V: PRINT
150 RETURN
160
170 REM Formarea numerelor posibile pentru o valoare a lui d
180 FOR A=1 TO 9
190 FOR B=1 TO 9
200 FOR C=1 TO 9
210 IF A+B+C=D THEN GO TO 250
220 NEXT C
230 NEXT B
240 NEXT A
250 RETURN

```

```

*****
* NUMERELE DE FORMA *
* ABC+BCA+CAB *
*****

```

NUMERELE DE 4 CIFRE CE PER-
MIT Scrierea CERUTA:

1110=118+181+811

1221=119+191+911

1332=129+291+912

$$1443=139+391+913$$

$$1554=149+491+914$$

$$1665=159+591+915$$

$$1776=169+691+916$$

$$1887=179+791+917$$

$$1998=189+891+918$$

$$2109=199+991+919$$

$$2220=299+992+929$$

$$2331=399+993+939$$

$$2442=499+994+949$$

$$2553=599+995+959$$

$$2664=699+996+969$$

$$2775=799+997+979$$

$$2886=899+998+989$$

$$2997=999+999+999$$

8.2.5 $k=x+(x+1)+\dots+(x+y)=x(y+1)+(1+2+\dots+y)=x(y+1)+y(y+1)/2=(y+1)(2x+y)/2$ de unde $y+1 \leq 2k$, adică $-y \geq 1-2k$. Din inegalitatea $2x+y \geq 0$ rezultă $2x \geq -y \geq 1-2k$, deci $x \geq (-2k+1)/2$, adică cea mai mică valoare pe care o poate lua x este $-k+1$ iar cea mai mare este k . Pentru fiecare valoare a lui x cuprinsă între aceste limite se calculează, într-o structură repetitivă, suma numerelor succesive ce începe cu x și se formează un șir ce conține aceste numere. Dacă suma acestora depășește pe k fără a-i atinge valoarea, se mărește cu 1 valoarea lui x și se reia calculul. Dacă suma atinge valoarea k , se tipărește șirul ce conține cifrele succesive și după aceea se mărește x pentru a se relua calculele.

```
10 LET Z1=4: LET Z2=25: GO SUB 280
20 PRINT " * NUMAR SCRIS CA SUMA *"
30 PRINT " * DE NUMERE CONSECUTIVE *": GO SUB 280
40 DIM A(100)
50 INPUT "INTRODUCETI NUMARUL "; K
60 PRINT " " "NUMARUL = "; K
70 IF K=0 THEN PRINT " " "O INFINITATE DE SOLUTII": GO TO 50
80 PRINT " " "SOLUTIILE SINT:"
90 FOR X=-K TO K
100 LET S=0: LET I=1: LET N=X
110 IF S>=K THEN GO TO 140
120 LET S=S+N: LET A(I)=N: LET N=N+1
130 LET I=I+1: GO TO 110
140 IF S<>K THEN GO TO 250
150 PRINT : PRINT K; " = "; : LET L=1+LEN STR$(K)
160 PRINT A(1); : LET L=L+LEN STR$(A(1))
```

```

170 FOR J=2 TO I-1
180 PRINT "+";
190 LET L=L+LEN STR$(A(J))+1.
200 IF A(J)>=0 THEN PRINT A(J);: GO TO 220
210 PRINT "(";A(J);")";: LET L=L+2
220 IF L>=28 THEN PRINT : LET L=0
230 NEXT J
240 PRINT
250 NEXT X
260 GO TO 50
270
280 FOR V=1 TO Z1: PRINT " ";: NEXT V
290 FOR V=1 TO Z2: PRINT "*";: NEXT V
300 PRINT
310 RETURN

```

```

*****
* NUMAR SCRIS CA SUMA *
* DE NUMERE CONSECUTIVE *
*****

```

NUMARUL = 0

0 INFINITATE DE SOLUTII

NUMARUL = 25

SOLUTIILE SINT:

$25 = -24 + (-23) + (-22) + (-21) + (-20)$
 $+ (-19) + (-18) + (-17) + (-16) + (-15)$
 $+ (-14) + (-13) + (-12) + (-11) + (-10)$
 $+ (-9) + (-8) + (-7) + (-6) + (-5) + (-4)$
 $+ (-3) + (-2) + (-1) + 0 + 1 + 2 + 3 + 4 + 5 + 6$
 $+ 7 + 8 + 9 + 10 + 11 + 12 + 13 + 14 + 15 + 16 + 17$
 $+ 18 + 19 + 20 + 21 + 22 + 23 + 24 + 25$

$25 = -11 + (-10) + (-9) + (-8) + (-7) + (-6)$
 $+ (-5) + (-4) + (-3) + (-2) + (-1) + 0 + 1$
 $+ 2 + 3 + 4 + 5 + 6 + 7 + 8 + 9 + 10 + 11 + 12 + 13$

$25 = -2 + (-1) + 0 + 1 + 2 + 3 + 4 + 5 + 6 + 7$

$25 = 3 + 4 + 5 + 6 + 7$

$25 = 12 + 13$

$25 = 25$

8.2.6 Se folosește o procedură ce calculează cel mai mare divizor comun a două numere naturale și se ține cont de egalitatea:

$$\text{cmmmc}(n, m) = \frac{n \cdot m}{\text{cmmdc}(n, m)}$$


```

50 PRINT "/// " PROGRAMUL VERIFICA, PENTRU 2 " " NUMERE ";
60 PRINT " NATURALE DATE EGALI-" " TATEA;"
70 PRINT " " CMMDC(N+M,CMMC(N,M))=" " CMMDC(N,M)"
80 INPUT "INTRODUCETI NR N=";"N;" M=";"M
90 PRINT "/// " NUMERELE CONSIDERATE SINT"
100 PRINT " " N=";"N;" M=";"M
110 LET X=N: LET Y=M: GO SUB 190: LET D=X
120 LET X=N+M: LET Y=(N*M)/D: LET Z=Y
130 GO SUB 190
140 PRINT " " CMMDC(";"M;" ";"N;"")=";"D
150 PRINT " " CMMDC(";"N+M;" ";"Z;"")=";"X
160 GO TO 80
170
180 REM Calcularea c.m.m.d.c.
190 IF X=Y THEN GO TO 220
200 IF X>Y THEN LET X=X-Y: GO TO 190
210 LET Y=Y-X: GO TO 190
220 RETURN

```

PROGRAMUL VERIFICA, PENTRU 2
NUMERE NATURALE DATE EGALI-
TATEA:

$CMMDC(N+M, CMMC(N, M)) =$
 $CMMDC(N, M)$

NUMERELE CONSIDERATE SINT

$N=4 \quad M=5$

$CMMDC(5, 4)=1$
 $CMMDC(9, 20)=1$

NUMERELE CONSIDERATE SINT

$N=20 \quad M=12$

$CMMDC(12, 20)=4$
 $CMMDC(32, 60)=4$

NUMERELE CONSIDERATE SINT

$N=28 \quad M=15$

$CMMDC(15, 28)=1$
 $CMMDC(43, 420)=1$

8.2.7 Pentru calcularea lui k , acesta se inițializează cu 1 și apoi se caută factorii lui n . Pentru fiecare factor găsit se mărește cu 1 valoarea lui k , apoi se împarte n la factorul respectiv de câte ori se divide cu el. În felul acesta toți factorii luați în considerare sînt primi.

algoritmul 8.2.7 este:

citește n

$k \leftarrow 0$

$m \leftarrow n$

$i \leftarrow 2$

cît-timp $i \leq m$ execută

 cheamă P

sfîrșit-cît-timp

$l \leftarrow 0$

pentru $i=1, n-1$ execută

 cheamă Q

sfîrșit-pentru

 scrie $l, 2^{k-1}$

stop

procedura Q este:

$p \leftarrow i \cdot (i+1)$

dacă $p/n = [p/n]$

 atunci $l \leftarrow l+1$

 scrie $i, i+1$

sfîrșit-dacă

sfîrșit-procedură

procedura P este:

dacă $m/i = [m/i]$ atunci $k \leftarrow k+1$

$m \leftarrow m/i$

cît-timp $m/i = [m/i]$ execută

$m \leftarrow m/i$

sfîrșit-cît-timp

sfîrșit-dacă

sfîrșit-procedură

```

10 CLS : LET z1=4: LET z2=25: GO SUB 290
20 PRINT " * VERIFICAM O EGALITATE *"
30 PRINT " * DIN TEORIA NUMERELOR *"
40 GO SUB 290
50 INPUT "INTRODUCETI NR N "; N
60 PRINT " " NUMARUL ";N;" ADMITE URMATORII"
70 PRINT " FACTORI PRIMI:"/" " ";
80 LET K=0: LET M=N: LET I=2
90 IF I>M THEN GO TO 150
100 IF M/I<>INT (M/I) THEN GO TO 140
110 LET K=K+1: LET M=M/I: PRINT I;" ";
120 IF M/I<>INT (M/I) THEN GO TO 140
130 LET M=M/I: GO TO 120
140 LET I=I+1: GO TO 90
150 PRINT " " PERECHILE DE NUMERE NATURALE CONSECUTIVE";
160 PRINT " AL CAROR PRODUS SE DIVIDE LA N SINT:"
170 PRINT
180 LET L=0
190 FOR I=1 TO N-1
200 IF I*(I+1)/N<>INT (I*(I+1)/N) THEN GO TO 230
210 PRINT " "; LET L=L+1: LET VV=I: GO SUB 330
220 LET VV=I+1: GO SUB 330: PRINT
230 NEXT I
240 PRINT " " SINT ";L;" PERECHI"
250 PRINT " " NUMARUL ";N;" ARE ";K;" FACTORI"/" " PRIMI."
260 PRINT " 2^";K;"-1=";2^K-1
270 GO TO 50
280
290 FOR V=1 TO Z1: PRINT " "; NEXT V
300 FOR V=1 TO Z2: PRINT "*""; NEXT V

```

```
310 PRINT : RETURN
320
330 FOR W=1 TO 5-LEN STR$ VV: PRINT " ";: NEXT W
340 PRINT VV;: RETURN
```

```
*****
* VERIFICAM O EGALITATE *
* DIN TEORIA NUMERELOR *
*****
```

NUMARUL 39 ADMITE URMATORII
FACTORI PRIMI:

3 13

PERECHILE DE NUMERE NATURALE
CONSECUTIVE AL CAROR PRODUS
SE DIVIDE LA N SINT:

12	13
26	27
38	39

SINT 3 PERECHI

NUMARUL 39 ARE 2 FACTORI
PRIMI.

$2^2-1=3$

NUMARUL 572 ADMITE URMATORII
FACTORI PRIMI:

2 11 13

PERECHILE DE NUMERE NATURALE
CONSECUTIVE AL CAROR PRODUS
SE DIVIDE LA N SINT:

143	144
208	209
220	221
351	352
363	364
428	429
571	572

SINT 7 PERECHI

NUMARUL 572 ARE 3 FACTORI
PRIMI.

$2^3-1=7$

8.2.8

algoritmul 8.2.8 este:

```

citește n
pentru i=1,n execută
  citește xi
sfârșit-pentru
j ← 0
pentru i=1,n execută
  cheamă P
sfârșit-pentru
dacă j=0 atunci scrie 'nici un element'
  altfel
    dacă j=n atunci scrie 'toate elementele'
    altfel
      pentru l=1,j execută
        scrie yl
      sfârșit-pentru
    sfârșit-dacă
  sfârșit-dacă
stop

```

procedura P este:

```

m ← xi
sw ← 0
l ← 2
cît-timp l ≤ [m/2] execută
  cheamă P1
sfârșit-cît-timp
dacă sw=0
  atunci j ← j+1
  yj ← xi
sfârșit-dacă
sfârșit-procedură

```

procedura P1 este:

```

k ← 0
cît-timp m/l=[m/l] execută
  k ← k+1
  m ← m/l
sfârșit-cît-timp
dacă k≠0
  atunci
    dacă k/2=[k/2]
      atunci sw ← 1
      i ← [m/2]
    sfârșit-dacă
  sfârșit-dacă
sfârșit-procedură

```

```

10 CLS : LET z1=2: LET z2=28: GO SUB 400
20 PRINT " * CAUTAM ELEMENTELE UNUI *"
30 PRINT " * SIR CARE SINT INTREGI *"
40 PRINT " * LIBERI DE PATRATE *"
50 GO SUB 400
60 INPUT "Introduceti dimensiunea sirului "; N
70 DIM X(N): DIM Y(N)
80 PRINT "' ' " SIRUL ESTE: "' ' " ";
90 FOR I=1 TO N
100 INPUT X(I): PRINT X(I); " ";
110 NEXT I
120 PRINT : LET J=0
130 FOR I=1 TO N
140 LET M=X(I): LET SW=0: LET L=2
150 IF L>INT (M/2) THEN GO TO 170
160 GO SUB 330: GO TO 150
170 IF SW=0 THEN LET J=J+1: LET Y(J)=X(I)
180 NEXT I
190 IF J=N THEN GO TO 290
200 IF J=0 THEN GO TO 270
210 PRINT "' ' " INTREGI LIBERI DE PATRATE"
220 PRINT " SINT: "' ' " ";
230 FOR L=1 TO J
240 PRINT Y(L); " ";
250 NEXT L
260 PRINT : GO TO 60

```



```

270 PRINT "// " SIRUL NU CONTINE INTREGI "
280 PRINT " LIBERI DE PATRATE": GO TO 60
290 PRINT "// " SIRUL CONTINE DOAR INTREGI "
300 PRINT " LIBERI DE PATRATE"
310 GO TO 60
320
330 LET K=0
340 IF M/L<>INT (M/L) THEN GO TO 360
350 LET K=K+1: LET M=M/L: GO TO 340
360 IF K=0 THEN GO TO 380
370 IF K/2=INT (K/2) THEN LET SW=1: LET L=INT (M/2)
380 LET L=L+1: RETURN
390
400 FOR V=1 TO Z1: PRINT " ";: NEXT V
410 FOR V=1 TO Z2: PRINT "*": NEXT V
420 PRINT : RETURN

```

```

*****
* CAUTAM ELEMENTELE UNUI *
* SIR CARE SINT INTREGI *
* LIBERI DE PATRATE *
*****

```

SIRUL ESTE:

12 13 25 27 81

INTREGI LIBERI DE PATRATE
SINT:

13 27

SIRUL ESTE:

12 9 81 20 50

SIRUL NU CONTINE INTREGI
LIBERI DE PATRATE

SIRUL ESTE:

13 27

SIRUL CONTINE DOAR INTREGI
LIBERI DE PATRATE

8.2.9 Un număr întreg se numește liber de puteri dacă în descompunerea sa în factori primi nici un factor nu apare la putere.

```

20 CLS : LET z1=2: LET z2=28: GO SUB 400
30 PRINT " * CAUTAM ELEMENTELE UNUI *"
40 PRINT " * SIR CARE SINT INTREGI *"
50 PRINT " * LIBERI DE PUTERI *"
60 GO SUB 400
70 INPUT "Introduceti dimensiunea sirului "; N
80 DIM X(N): DIM Y(N)
90 PRINT "' ' " SIRUL ESTE: "' ' " ";
100 FOR I=1 TO N
110 INPUT X(I): PRINT X(I); " ";
120 NEXT I
130 PRINT : LET J=0
140 FOR I=1 TO N
150 LET M=X(I): LET SW=0: LET L=2
160 IF L>INT (M/2) THEN GO TO 180
170 GO SUB 340: GO TO 160
180 IF SW=0 THEN LET J=J+1: LET Y(J)=X(I)
190 NEXT I
200 IF J=N THEN GO TO 300
210 IF J=0 THEN GO TO 280
220 PRINT "' ' " INTREGI LIBERI DE PUTERI"
230 PRINT " SINT: "' ' " ";
240 FOR L=1 TO J
250 PRINT Y(L); " ";
260 NEXT L
270 PRINT : GO TO 70
280 PRINT "' ' " SIRUL NU CONTINE INTREGI "
290 PRINT " LIBERI DE PUTERI": GO TO 70
300 PRINT "' ' " SIRUL CONTINE DOAR INTREGI "
310 PRINT " LIBERI DE PUTERI"
320 GO TO 70
330
340 IF M/L<>INT (M/L) THEN GO TO 370
350 LET M=M/L
360 IF M/L=INT (M/L) THEN LET SW=1: LET L=INT (M/L)
370 LET L=L+1
380 RETURN
390
400 FOR V=1 TO Z1: PRINT " ";; NEXT V
410 FOR V=1 TO Z2: PRINT " *";; NEXT V
420 PRINT : RETURN

```

```

*****
* CAUTAM ELEMENTELE UNUI *
* SIR CARE SINT INTREGI *
* LIBERI DE PUTERI *
*****

```

SIRUL ESTE:

12 13 25 27 81

INTREGI LIBERI DE PUTERI

SINT:

13

SIRUL ESTE:

12 9 81 50 20

SIRUL NU CONTINE INTREGI
LIBERI DE PUTERI

SIRUL ESTE:

13 15

SIRUL CONTINE DOAR INTREGI
LIBERI DE PUTERI

8.2.10 Pentru fiecare număr $n \in (1, m)$ se calculează suma tuturor divizorilor săi pozitivi. Dacă această sumă este egală cu n , atunci el se depune într-un șir. De asemenea se formează un șir cu cifrele lui n și se verifică dacă elementele egal depărtate de extremități sînt egale două cite două, caz în care numărul n este palindrom și trebuie depus într-un alt șir.

algoritmul 8.2.10 este:

```
citește m1
i ← 0
j ← 0
pentru n=2, m1-1 execută
  cheamă P
  cheamă Q
sfîrșit-pentru
dacă i=0 atunci scrie 'nici un element perfect'
    altfel [ pentru l=1, i execută
              scrie  $x_l$ 
              sfîrșit-pentru
sfîrșit-dacă
dacă j=0 atunci scrie 'nici un palindrom'
    altfel [ pentru l=1, j execută
              scrie  $y_l$ 
              sfîrșit-pentru
sfîrșit-dacă
stop
```

procedura P este:

```
s1 ← 0
pentru l=1, n-1 execută
  cheamă P1
sfîrșit-pentru
dacă s1=n atunci i ← i+1
               $x_l$  ← n
sfîrșit-dacă
sfîrșit-procedură
```

procedura P1 este:

```
[ dacă  $n/l = \lfloor n/l \rfloor$  atunci s1 ← s1+1
  sfîrșit-dacă
sfîrșit-procedură
```

procedura Q este:

$l \leftarrow 0$

$m \leftarrow n$

$c \leftarrow \lfloor m/10 \rfloor$

$r \leftarrow m - 10 \cdot c$

cît-timp $c \neq 0$ execută

$l \leftarrow l + 1$

$a_l \leftarrow r$

$m \leftarrow c$

$c \leftarrow \lfloor m/10 \rfloor$

$r \leftarrow m - 10 \cdot c$

sfîrșit-cît-timp

$l \leftarrow l + 1$

$a_l \leftarrow r$

$s_2 \leftarrow 0$

$k \leftarrow 1$

cît-timp $k \leq \lfloor l/2 \rfloor$ execută

dacă $a_k \neq a_{l+1-k}$ atunci $s_2 \leftarrow 1$

$k \leftarrow \lfloor l/2 \rfloor$

altfel $k \leftarrow k + 1$

sfîrșit-dacă

sfîrșit-cît-timp

dacă $s_2 = 0$ atunci $j \leftarrow j + 1$

$y_j \leftarrow n$

sfîrșit-dacă

sfîrșit-procedură

```
60 INPUT "Introduceti M "; M1
70 PRINT " " INTERVALUL CONSIDERAT ESTE: " " (1, " ; M1 ; " )
80 DIM X(M1): DIM Y(M1): DIM A(10)
90 LET I=0: LET J=0
100 FOR N=2 TO M1-1
110 GO SUB 310: GO SUB 390
120 NEXT N
130 IF J=0 THEN GO TO 190
140 PRINT " " PALINDROAME SINT: " " " ";
150 FOR L=1 TO J
160 LET VV=Y(L): GO SUB 540
170 NEXT L
180 PRINT : GO TO 210
190 PRINT " " IN INTERVALUL CONSIDERAT NU"
200 PRINT " " SINT PALINDROAME"
210 IF I=0 THEN GO TO 270
220 PRINT " " NUMERE PERFECTE SINT: " " " ";
230 FOR L=1 TO I
240 LET VV=X(L): GO SUB 540
250 NEXT L
260 PRINT : GO TO 290
270 PRINT " " IN INTERVALUL CONSIDERAT NU"
280 PRINT " " SINT NUMERE PERFECTE"
290 GO TO 60
300
310 LET S1=0
320 FOR L=1 TO N-1
330 IF N/L=INT(N/L) THEN LET S1=S1+L
340 NEXT L
350 IF S1 <> N THEN GO TO 370
360 LET I=I+1: LET X(I)=N
370 RETURN
```



```

390 LET L=0: LET M=N
400 LET C=INT (M/10): LET R=M-10*C
410 IF C=0 THEN GO TO 430
420 LET L=L+1: LET A(L)=R: LET M=C: GO TO 400
430 LET L=L+1: LET A(L)=R: LET S2=0
440 FOR K=1 TO INT (L/2)
450 IF A(K)<>A(L+1-K) THEN LET S2=1: LET K=INT (L/2)
460 NEXT K
470 IF S2=0 THEN LET J=J+1: LET Y(J)=N
480 RETURN
540 FOR W=1 TO 4-LEN STR$ (VV): PRINT " ";: NEXT W
550 PRINT VV;: IF INT (L/5)=L/5 THEN PRINT " " ";
560 RETURN

```

INTERVALUL CONSIDERAT ESTE:
(1,30)

PALINDROAME SINT:

```

  2  3  4  5  6
  7  8  9 11 22

```

NUMERE PERFECTE SINT:

6 28

8.3.1

algoritmul 8.3.1 este:

citește a,b,c

dacă a=0

atunci

dacă b=0

atunci

dacă c=0

atunci scrie 'ecuație nedeterminată'

altfel scrie 'ecuație imposibilă'

sfârșit-dacă

altfel scrie 'ecuație grad I, x=' , - $\frac{c}{b}$

sfârșit-dacă

altfel $d \leftarrow b^2 - 4 \cdot a \cdot c$

dacă $d \geq 0$ atunci

dacă d=0

atunci $x_1 \leftarrow -\frac{b}{2 \cdot a}$

scrie 'rădăcină dublă', x_1

altfel $d \leftarrow \sqrt{d}$

$x_1 \leftarrow -\frac{b-d}{2 \cdot a}$

$x_2 \leftarrow -\frac{b+d}{2 \cdot a}$

scrie 'rădăcinile', x_1, x_2

sfârșit-dacă

altfel $d \leftarrow \sqrt{-d}$

rex $\leftarrow -\frac{b}{2 \cdot a}$

imx $\leftarrow \frac{d}{2 \cdot a}$

scrie 'rădăcini complexe', rex, imx

sfârșit-dacă

sfârșit-dacă

stop

```

50 PRINT "*****"
60 PRINT "* + ECUATIA DE GRADUL II *"
70 PRINT "* CU COEFICIENTI REALI *"
80 PRINT "*****"
90 INPUT "Introdu coeficientii a,b,c: " a=";A;" b=";B;" c
=";C
100 PRINT "Ecuatia data este: " ^A;"*X^2"+("+" AND B>=0);B;"*X"+
("+" AND C>=0);C;"=0"
110 IF A=0 THEN GO TO 230
120 LET D=B*B-4*A*C
130 IF D>=0 THEN GO TO 190
140 LET D=(-D)^(1/2)
150 LET REX=(-B)/(2*A): LET IMX=D/(2*A)
160 PRINT "Radacini complexe" X1=";REX;("+" AND IMX>=0);IMX
;"*i"
170 PRINT " X2=";REX;("+" AND -IMX>=0);-IMX;"*i"
180 GO TO 270
190 IF D=0 THEN LET X1=-B/(2*A): PRINT "Radacina dubla" X1
=X2=";X1: GO TO 270
200 LET D=D^(1/2): LET X1=(-B-D)/(2*A): LET X2=(-B+D)/(2*A)
210 PRINT "Radacini reale distincte"
220 PRINT " X1=";X1" X2=";X2: GO TO 270
230 IF B=0 THEN GO TO 250
240 PRINT "Ecuatie de gradul I, x=";-C/B: GO TO 270
250 IF C=0 THEN PRINT "Ecuatie nedeterminata": GO TO 270
260 PRINT "Ecuatie imposibila"
270 STOP

```

```

*****
* ECUATIA DE GRADUL II *
* CU COEFICIENTI REALI *
*****

```

Ecuatia data este:
 $1 \times X^2 - 3 \times X + 2 = 0$

Radacini reale distincte
 $X_1 = 1$
 $X_2 = 2$

Ecuatia data este:
 $1 \times X^2 - 2 \times X + 1 = 0$

Radacina dubla
 $X_1 = X_2 = 1$

Ecuatia data este:
 $1 \times X^2 + 1 \times X + 1 = 0$

Radacini complexe
 $X_1 = -0.5 + 0.8660254 \times i$
 $X_2 = -0.5 - 0.8660254 \times i$

Ecuatia data este:
 $0 \times X^2 + 1 \times X + 2 = 0$

Ecuatie de gradul I, $x = -2$

Ecuatia data este:
 $0 \times X^2 + 0 \times X - 3 = 0$

Ecuatie imposibila

Ecuatia data este:
 $0 \times X^2 + 0 \times X + 0 = 0$

Ecuatie nedeterminata

8.3.2

```
10 REM
20 REM *** CALCUL EXPRESIE COMPLEXA ***
30 REM
40 CLS
50 PRINT "*****"
60 PRINT "*   CALCUL EXPRESIE COMPLEXA   *"
70 PRINT "*****"
80 INPUT "Introdu valorile a(a1+i*a2):" a1=";A1;" a2="
;A2
90 INPUT "si b(b1+i*b2): b1=";B1;" b2=";B2
100 INPUT "Introdu valoarea z(z1+i*z2):" z1=";Z1;" z2=";
Z2
110 LET MODZ=SQR (Z1*Z1+Z2*Z2): REM modulul lui Z
120 LET X1=A1: LET X2=A2: LET Y1=Z1: LET Y2=Z2: GO SUB 280
130 LET REAZ=P1: LET IMAZ=P2: REM A*Z=REAZ+i*IMAZ
140 LET X1=Z1: LET X2=Z2: LET Y1=Z1: LET Y2=Z2: GO SUB 280
150 LET REZ2=P1: LET IMZ2=P2: REM Z^2=REZ2+i*IMZ2
160 IF MODZ<=1 THEN LET RE=3*REZ2-6*REAZ+B1: LET IM=3*IMZ2-
6*IMAZ+B2: GO TO 180
170 LET RE=6*REAZ-B1: LET IM=6*IMAZ-B2
180 PRINT "Pentru valorile:"
190 PRINT "  a=";A1;"+" AND A2>=0;A2;"*i"
200 PRINT "  b=";B1;"+" AND B2>=0;B2;"*i"
210 PRINT "  z=";Z1;"+" AND Z2>=0;Z2;"*i"
220 PRINT "Valoarea expresiei este:"
230 PRINT "E=";RE;"+" AND IM>=0;IM;"*i": REM E=RE+i*IM
240 STOP
250 REM
260 REM procedura calcul produs
270 REM
280 LET P1=X1*Y1-X2*Y2: REM P1=Re(X*Y)
290 LET P2=X1*Y2+X2*Y1: REM P2=Im(X*Y)
300 RETURN
```

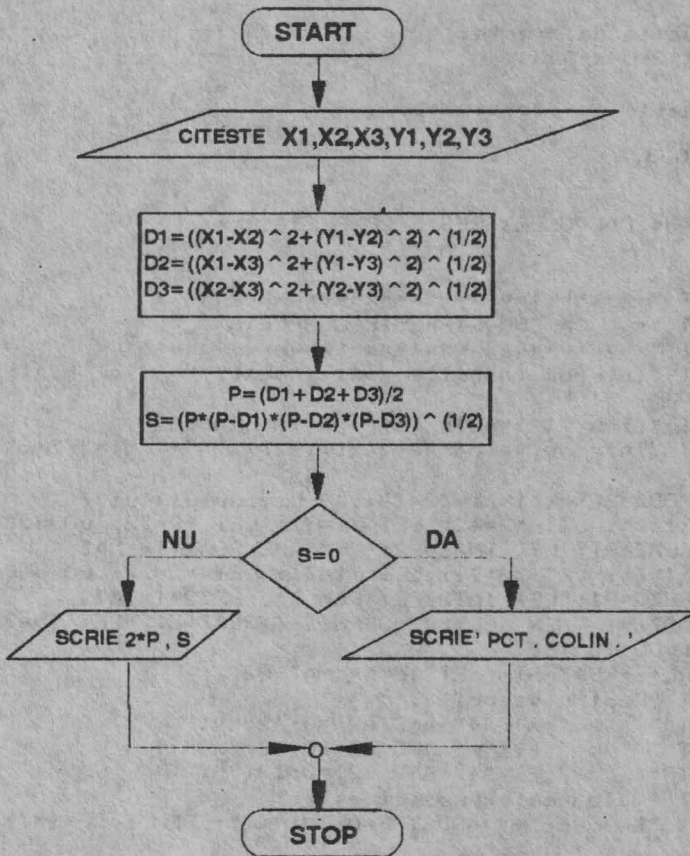
```
*****
*   CALCUL EXPRESIE COMPLEXA   *
*****
```

Pentru valorile:

a=1+2*i
b=2-1*i
z=-1-1*i

Valoarea expresiei este:
E=4-17*i

8.4.1 Avind coordonatele virfurilor triunghiului se obțin lungimile laturilor sale și apoi aria sa, folosind formula lui Heron. Cazul $S=0$ corespunde unui triunghi degenerat (cele trei virfuri sînt coliniare).



algoritmul 8.4.1. este:

citește $x_1, y_1, x_2, y_2, x_3, y_3$

$$d_1 \leftarrow \sqrt{(x_1-x_2)^2+(y_1-y_2)^2}$$

$$d_2 \leftarrow \sqrt{(x_1-x_3)^2+(y_1-y_2)^2}$$

$$d_3 \leftarrow \sqrt{(x_2-x_3)^2+(y_2-y_3)^2}$$

$$p \leftarrow \frac{d_1+d_2+d_3}{2}$$

$$s \leftarrow \sqrt{p \cdot (p-d_1) \cdot (p-d_2) \cdot (p-d_3)}$$

dacă $s=0$ atunci scrie 'puncte coliniare'

altfel scrie $2 \cdot p, s$

sfîrșit-dacă

stop


```

10 REM
20 REM *** PERIMETRUL SI ARIA TRIUNGHULUI ***
30 REM
40 CLS
50 PRINT "*****"
60 PRINT " *          PERIMETRUL SI ARIA          *"
70 PRINT " *          TRIUNGHULUI                    *"
80 PRINT "*****"
90 INPUT "Coordonatele lui A(x1,y1)" x1=";X1;
" y1=";Y1
100 INPUT "Coordonatele lui B(x2,y2)" x2=";X2;
" y2=";Y2
110 INPUT "Coordonatele lui C(x3,y3)" x3=";X3;
" y3=";Y3
120 PRINT "Punctele sint" A(";X1;";";Y1;")"
B(";X2;";";Y2;")" C(";X3;";";Y3;")"
130 LET D1=SQR ((X1-X2)*(X1-X2)+(Y1-Y2)*(Y1-Y2))
140 LET D2=SQR ((X1-X3)*(X1-X3)+(Y1-Y3)*(Y1-Y3))
150 LET D3=SQR ((X2-X3)*(X2-X3)+(Y2-Y3)*(Y2-Y3))
160 LET P=(D1+D2+D3)/2
170 LET S=SQR (P*(P-D1)*(P-D2)*(P-D3))
180 IF S=0 THEN PRINT "CELE 3 PUNCTE SINT COLINIA
RE": GO TO 200
190 PRINT "Perimetrul este:";2*P "Suprafata este:"
;S
200 STOP

```

```

*****
*          PERIMETRUL SI ARIA          *
*          TRIUNGHULUI                    *
*****

```

```

Punctele sint
A(0,0)
B(3,0)
C(0,4)

```

```

Perimetrul este:12
Suprafata este:6

```

```

Punctele sint
A(-3,0)
B(3,0)
C(0,6)

```

```

Perimetrul este:19.416408
Suprafata este:18

```

```

Punctele sint
A(1,1)
B(2,2)
C(3,3)

```

```

CELE 3 PUNCTE SINT COLINIARE

```

8.4.2 Fie $A(x_1, y_1)$ și $B(x_2, y_2)$ două dintre virfurile triunghiului. Atunci mijlocul laturii AB este punctul $M((x_1+x_2)/2, (y_1+y_2)/2)$ iar coeficientul unghiular al dreptei AB este: $m = (y_1 - y_2) / (x_1 - x_2)$, ceea ce permite scrierea ecuației mediatoarei segmentului AB :

$$y - \frac{y_1 + y_2}{2} = - \frac{1/m}{x - (x_1 + x_2)/2}$$

Analog se scrie ecuația celei de a doua mediatoare, coordonatele centrului cercului circumscris triunghiului obținându-se intersectând cele două mediatoare (adică rezolvând sistemul format de ecuațiile lor). Raza cercului este egală cu distanța de la centrul cercului la unul dintre virfurile triunghiului. Dacă cele trei puncte sunt coliniare atunci coeficienții unghiulari a oricare două laturi sînt egale. Dacă unul dintre coeficienții unghiulari este nul trebuie ținut cont de faptul că acea mediatoare este paralelă cu axa Oy . Separat se tratează cazurile $x_1 = x_2$ respectiv $x_1 = x_3$. Dacă $x_1 = x_2$ și $y_1 = y_3$ atunci triunghiul este dreptunghic și centrul cercului circumscris se află la mijlocul ipotenuzei.

algoritmul 8.4.2. este:

citește $x_1, y_1, x_2, y_2, x_3, y_3$

dacă $(x_1 - x_2) \cdot (x_1 - x_3) \neq 0$ atunci $m_1 \leftarrow (y_1 - y_2) / (x_1 - x_2)$

$m_2 \leftarrow (y_1 - y_3) / (x_1 - x_3)$

dacă $m_1 \neq m_2$ atunci cheamă P

altfel serie

'puncte coliniare'

sfîrșit-dacă

altfel

dacă $x_1 = x_3$ atunci cheamă P_1

altfel cheamă P_2

sfîrșit-dacă

sfîrșit-dacă

stop

procedura P este:

$x_4 \leftarrow (x_1 + x_2) / 2$

$x_5 \leftarrow (x_1 + x_3) / 2$

$y_4 \leftarrow (y_1 + y_2) / 2$

$y_5 \leftarrow (y_1 + y_3) / 2$

dacă $m_1 \cdot m_2 = 0$ atunci

dacă $m_1 = 0$ atunci $m_2 \leftarrow (-1) / m_2$

$y \leftarrow y_4$

$x \leftarrow (y - y_5 + m_2 \cdot x_5) / m_2$

altfel $m_1 \leftarrow (-1) / m_1$

$y \leftarrow y_5$

$x \leftarrow (y - y_4 + m_1 \cdot x_4) / m_1$

sfîrșit-dacă

altfel $m_1 \leftarrow (-1) / m_1$

$m_2 \leftarrow (-1) / m_2$

$x \leftarrow m_2 \cdot x_5 - m_1 \cdot x_4 - y_5 + y_4$

$x \leftarrow x / (m_2 - m_1)$

$y \leftarrow y_4 + m_1 \cdot (x - x_4)$

sfîrșit-dacă

$r \leftarrow \sqrt{(x - x_1)^2 + (y - y_1)^2}$

serie x, y, r

sfîrșit-procedură

procedura P1 este:

```
dacă  $x_1 \neq x_2$ 
  atunci  $m_1 \leftarrow (y_1 - y_2) / (x_1 - x_2)$ 
    dacă  $m_1 \neq 0$  atunci  $x_4 \leftarrow (x_1 + x_2) / 2$ 
       $x \leftarrow x_4$ 
       $y_4 \leftarrow (y_1 + y_2) / 2$ 
       $y \leftarrow y_4 - (x - x_4) / m_1$ 
    altfel  $x \leftarrow (x_2 + x_3) / 2$ 
       $y \leftarrow (y_2 + y_3) / 2$ 
  sfîrșit-dacă
   $r \leftarrow \sqrt{(x - x_1)^2 + (y - y_1)^2}$ 
  scrie x, y, r
altfel scrie 'puncte coliniare'
sfîrșit-dacă
sfîrșit-procedură
```

procedura P2 este:

```
 $m_2 \leftarrow (y_1 - y_3) / (x_1 - x_3)$ 
dacă  $m_2 = 0$  atunci  $x \leftarrow (x_2 + x_3) / 2$ 
   $y \leftarrow (y_2 + y_3) / 2$ 
altfel  $x_5 \leftarrow (x_1 + x_3) / 2$ 
   $y_5 \leftarrow (y_1 + y_3) / 2$ 
   $x \leftarrow x_5$ 
   $y \leftarrow y_5 - (x - x_5) / m_2$ 
sfîrșit-dacă
 $r \leftarrow \sqrt{(x - x_1)^2 + (y - y_1)^2}$ 
scrie x, y, r
sfîrșit-procedură
```

```
10 REM
20 REM *** CENTRUL SI RAZA CERCULUI CIRCUMSCRIS ***
30 REM
40 CLS
50 PRINT "*****"
60 PRINT "*   CENTRUL SI RAZA CERCULUI   *"
70 PRINT "*           CIRCUMSCRIS           *"
80 PRINT "*****"
90 INPUT "Virful A(x1,y1)  x1=";X1;"   y1=";Y1
100 INPUT "Virful B(x2,y2) x2=";X2;"   y2=";Y2
110 INPUT "Virful C(x3,y3) x3=";X3;"   y3=";Y3
120 IF (X1-X2)*(X1-X3)<>0 THEN GO TO 150
130 IF X1=X3 THEN GO SUB 180: GO TO 400
140 GO SUB 260: GO TO 400
150 LET M1=(Y1-Y2)/(X1-X2): LET M2=(Y1-Y3)/(X1-X3)
160 IF M1<>M2 THEN GO SUB 300: GO TO 400
170 PRINT "PUNCTELE:";": GO SUB 360: PRINT "SINT COLINI
ARE!!!": GO TO 400
180 IF X1<>X2 THEN GO TO 200
190 PRINT "PUNCTELE:";": GO SUB 360: PRINT "SINT COLINI
ARE!!!": RETURN
200 LET M1=(Y1-Y2)/(X1-X2)
210 IF M1<>0 THEN LET X4=(X1+X2)/2: LET X=X4: LET Y4=(Y
1+Y2)/2: LET Y=Y4-(1/M1)*(X-X4): GO TO 230
220 LET X=(X2+X3)/2: LET Y=(Y2+Y3)/2
230 LET R=SQR ((X-X1)*(X-X1)+(Y-Y1)*(Y-Y1))
240 PRINT "TRIUNghiUL CU VIRFURILE:";": GO SUB 360
250 PRINT "ARE CERCUL CIRCUMSCRIS DE RAZA:";"/" R=";
R: RETURN
260 LET M2=(Y1-Y3)/(X1-X3)
```

```

270 IF M2=0 THEN LET X=(X2+X3)/2: LET Y=(Y2+Y3)/2: GO TO 290
280 LET X5=(X1+X3)/2: LET Y5=(Y1+Y3)/2: LET X=X5: LET Y=
Y5-(1/M2)*(X-X5)
290 GO SUB 230: RETURN
300 LET X4=(X1+X2)/2: LET X5=(X1+X3)/2: LET Y4=(Y1+Y2)/2
: LET Y5=(Y1+Y3)/2
310 IF M1*M2=0 THEN GO TO 330
320 LET M1=-1/M1: LET M2=-1/M2: LET X=M2*X5-M1*X4-Y5+Y4:
LET X=X/(M2-M1): LET Y=Y4+M1*(X-X4): GO TO 350
330 IF M1=0 THEN LET M2=-1/M2: LET Y=Y4: LET X=(Y-Y5+M2
*X5)/M2: GO TO 350
340 LET M1=-1/M1: LET Y=Y5: LET X=(Y-Y4+M1*X4)/M1
350 GO SUB 230: RETURN
360 PRINT " A(";X1;",";Y1;")"
370 PRINT " B(";X2;",";Y2;")"
380 PRINT " C(";X3;",";Y3;")"
390 RETURN
400 STOP

```

```

*****
* CENTRUL SI RAZA CERCULUI *
* CIRCUMSCRIS *
*****

```

TRIUNghiUL CU VIRFURILE:

```

A(0,0)
B(3,0)
C(0,4)

```

ARE CERCUL CIRCUMSCRIS DE RAZA:
R=2.5

TRIUNghiUL CU VIRFURILE:

```

A(-1,-1)
B(1,2)
C(4,0)

```

ARE CERCUL CIRCUMSCRIS DE RAZA:
R=2.5495098

TRIUNghiUL CU VIRFURILE:

```

A(0,0)
B(0,1)
C(2,1)

```

ARE CERCUL CIRCUMSCRIS DE RAZA:
R=1.118034

PUNCTELE:

```

A(1,1)
B(2,2)
C(3,3)

```

SINT COLINIARE!!!

8.4.3 Pentru ca scrierea să se poată face la sfârșitul algoritmului, se folosește un switch inițializat cu zero care ia valoarea unu dacă două dintre laturile triunghiului sînt egale.

```

10 REM
20 REM *** FORMEAZA TRIUNGHI ISOSCEL ? ***
30 REM
40 CLS
50 PRINT "*****"
60 PRINT "* FORMEAZA TRIUNGHI ISOSCEL ? *"
70 PRINT "*****"
80 INPUT "Virful A(x1,y1) x1=";X1;" y1=";Y1
90 INPUT "Virful B(x2,y2) x2=";X2;" y2=";Y2
100 INPUT "Virful C(x3,y3) x3=";X3;" y3=";Y3
110 LET SW=0
120 REM === calcul dimensiuni laturi ===
130 LET D1=(X1-X2)*(X1-X2)+(Y1-Y2)*(Y1-Y2)
140 LET D2=(X1-X3)*(X1-X3)+(Y1-Y3)*(Y1-Y3)
150 LET D3=(X2-X3)*(X2-X3)+(Y2-Y3)*(Y2-Y3)
160 REM ??? test triunghi ???
170 IF SQR D1>=SQR D2+SQR D3 THEN GO TO 80
180 IF SQR D2<=SQR D3 THEN LET D=SQR D3-SQR D2: GO TO 200
190 LET D=SQR D2-SQR D3
200 IF SQR D1<=D THEN GO TO 80
210 REM *** verificare egalitate laturi ***
220 IF D1=D2 THEN LET SW=1: GO TO 260
230 IF D1=D3 THEN LET SW=1: GO TO 260
240 IF D2=D3 THEN LET SW=1
250 REM !!! scrie rezultat !!!
260 PRINT "TRIUNGHIUL CU VIRFURILE:"
270 PRINT " A(";X1;" ";Y1;")"
280 PRINT " B(";X2;" ";Y2;")"
290 PRINT " C(";X3;" ";Y3;")"
300 IF SW=0 THEN PRINT "NU ";
310 PRINT "ESTE ISOSCEL!!!"

```

```

*****
* FORMEAZA TRIUNGHI ISOSCEL ? *
*****

```

TRIUNGHIUL CU VIRFURILE:

```

A(-1,-1)
B(1,-1)
C(0,4) ESTE ISOSCEL!!!

```

TRIUNGHIUL CU VIRFURILE:

```

A(0,0)
B(3,0)
C(0,4) NU ESTE ISOSCEL!!!

```

TRIUNGHIUL CU VIRFURILE:

```

A(0,1)
B(2,3)
C(2,1) ESTE ISOSCEL!!!

```

8.5.1 Indicele i al șirului se inițializează cu zero și se mărește cu 1 înainte de a depune un element. Un contor k , inițializat cu doi, arată câte elemente consecutive egale cu -1 se vor inițializa în etapa următoare. Testul $i+k+1 < n$ arată dacă următoarea secvență de tipul 1,-1,-1,... are loc sau nu în întregime în șir.

```

10 CLS : LET z1=6: LET z2=19: GO SUB 280
20 PRINT " * GENERARE DE SIR *"
30 GO SUB 280
40 INPUT "Introduceti dim sirului ";N: DIM X(N)
50 LET I=0: LET K=2
60 IF I+K+1>N THEN GO TO 80
70 GO SUB 210: GO TO 60
80 IF I=N THEN GO TO 110
90 IF I=N-1 THEN LET X(N)=1: GO TO 110
100 LET K=N-I-1: GO SUB 210
110 PRINT "/ " SIRUL CU ";N;" ELEMENTE GENERAT
120 PRINT " ESTE:"/" " ";
130 LET L=0
140 FOR I=1 TO N
150 IF X(I)=1 THEN PRINT " ";
160 PRINT X(I);" ";
170 LET L=L+1: IF L=8 THEN PRINT "/ " " "; : LET L=0
180 NEXT I
190 PRINT : GO TO 40
200
210 LET I=I+1: LET X(I)=1
220 FOR J=1 TO K
230 LET X(I+J)=-1
240 NEXT J
250 LET I=I+K: LET K=K+1
260 RETURN
270
280 FOR V=1 TO Z1: PRINT " ";: NEXT V
290 FOR V=1 TO Z2: PRINT " *": NEXT V
300 PRINT : RETURN

```

```

*****
* GENERARE DE SIR *
*****

```

```

SIRUL CU 18 ELEMENTE GENERAT
ESTE:

```

```

  1 -1 -1  1 -1 -1 -1  1
-1 -1 -1 -1  1 -1 -1 -1
-1 -1

```

```

SIRUL CU 23 ELEMENTE GENERAT
ESTE:

```

```

  1 -1 -1  1 -1 -1 -1  1
-1 -1 -1 -1  1 -1 -1 -1
-1 -1  1 -1 -1 -1 -1

```

8.5.2 Se folosesc patru variabile k_1, k_2, l_1, l_2 cu următoarea semnificație: pentru fiecare indice $i \geq 3$, k_1 (respectiv l_1) arată câte caractere "a" conține termenul al $(i-2)$ -lea (respectiv al $(i-1)$ -lea) iar k_2 (respectiv l_2) arată câte caractere "b" conțin acești termeni. Aceste variabile se inițializează pentru termenii S_1 , respectiv S_2 , iar apoi se calculează numărul de caractere "a" și "b" conținute în termenul S_k . Pentru scrierea termenilor se folosesc variabile de tip caracter pentru care concatenarea se face cu operatorul "+".

```

50 INPUT " Introduceți dim sirului ";K
60 LET K1=1: LET K2=0: LET L1=0: LET L2=1
70 PRINT " " " TERM. CAR.A CAR.B"
80 PRINT "   1       1       0"
90 PRINT "   2       0       1"
100 FOR I=3 TO K
110 LET X=K1+L1: LET Y=K2+L2
120 LET K1=L1: LET L1=X
130 LET K2=L2: LET L2=Y
140 PRINT "   "; I; "       ";L1; "       ";L2
150 NEXT I
160 PRINT " " " SIRUL ESTE URMATORUL: "
170 PRINT "   A " "   B "
180 LET X$="A": LET Y$="B"
190 LET SW=0: LET M=1: LET N=1
200 FOR I=3 TO K
210 LET Z$=X$+Y$: PRINT "   "; Z$
220 LET X$=Y$: LET Y$=Z$
230 NEXT I
240 PAUSE 0: CLS : STOP
250
260 FOR V=1 TO Z1: PRINT " ";: NEXT V
270 FOR V=1 TO Z2: PRINT " * ";
280 NEXT V: PRINT : RETURN

```

TERM.	CAR.A	CAR.B
1	1	0
2	0	1
3	1	1
4	1	2
5	2	3
6	3	5
7	5	8
8	8	13

SIRUL ESTE URMATORUL:

```

A
B
AB
BAB
ABBAB
BABABBAB
ABBABBABABBAB
BABABBABABBABABBAB

```

8.5.3 Algoritmul de căutare binară este următorul: se compară y (valoarea căutată în șirul dat) cu elementul aflat la mijlocul șirului; dacă el este mai mare, îl comparăm cu elementul aflat la mijlocul jumătății din dreapta a șirului iar dacă este mai mic cu cel aflat la mijlocul jumătății din stânga și așa mai departe. Pentru aceasta se inițializează o variabilă p cu 1 și o variabilă q cu n (n este dimensiunea șirului X). Se

compară y cu x_i , unde $i = \lfloor (p+q)/2 \rfloor$. Dacă ele sînt egale, algoritmul se încheie cu succes. Dacă $y < x_i$, q ia valoarea $i-1$ iar dacă $y > x_i$ atunci p ia valoarea $i+1$, după care se reia algoritmul de la compararea lui y cu x_i atîta vreme cît $p < q$. Dacă y nu se află printre elementele șirului X după un anumit număr de pași p îl depășește pe q , ceea ce indică încheierea algoritmului fără succes. Programul conține o rutină care verifică dacă șirul X este sau nu ordonat și se trece la căutare doar în caz afirmativ.

algoritmul 8.5.3. este:

```

citește y, n
pentru i=1, n execută
    citește xi
sfîrșit-pentru
p ← 1
q ← n
l ← 0
execută
    dacă q < p
        atunci scrie
            'fără succes'
            l ← 1
        altfel cheamă P
sfîrșit-dacă
ieșim-cînd l=1
stop

```

procedura P este:

```

i ← ⌊ (p+q) / 2 ⌋
dacă y ≠ xi
    atunci
        dacă y < xi
            atunci q ← i-1
            altfel p ← i+1
        sfîrșit-dacă
    altfel scrie 'cu succes', i
    l ← 1
sfîrșit-dacă
sfîrșit-procedură

```

```

50 PRINT "*****"
60 PRINT "*      CAUTARE BINARA      *"
70 PRINT "*****"
80 INPUT "dimensiunea sirului n= "; n
90 IF n <= 0 OR n <> INT n THEN GO TO 80
100 INPUT "elementul care trebuie cautat: "; y
110
120 DIM x(n): LET p=1: LET q=n: LET l=0: LET sw=0
130
140 PRINT ""sirul initial x de dimensiune "; n; " es
te:"
160 FOR i=1 TO n
170   INPUT "x("; i; ")="; x(i)
180   PRINT x(i); " ";
190 NEXT i
200
210 FOR i=1 TO n-1
220   IF x(i) > x(i+1) THEN LET sw=1: LET i=n
230 NEXT i
240 IF sw=1 THEN PRINT ""acest sir nu este ordonat
": GO TO 120
250
260 IF q < p THEN PRINT ""y; " nu se afla in sir": LE
T l=1: GO TO 310
270 LET i=INT ((p+q)/2)
280 IF y=x(i) THEN PRINT ""!!! "; y; " se afla pe po
zitia "; i; " !!!": LET l=1: GO TO 310
290 IF y < x(i) THEN LET q=i-1: GO TO 310
300 LET p=i+1
310 IF l <> 1 THEN GO TO 260
320 STOP

```



```
*****
*          CAUTARE BINARA          *
*****
```

sirul initial x de dimensiune 13 este:
0 2 5 1 -7 -5 9 13 -8 3 5 7 -1

acest sir nu este ordonat

sirul initial x de dimensiune 13 este:
-23 -20 -11 -6 -1 0 0 10 12 33 35 46 123

13 nu se afla in sir

sirul initial x de dimensiune 13 este:
-4 -2 -1 0 0 2 4 7 13 24 35 100 123

!!! 13 se afla pe pozitia 9 !!!

8.5.4

algoritmul 8.5.4. este:

```

citește n
  pentru i=1,n execută
    citește  $x_i$ 
  sfârșit-pentru
  pentru i=1,n execută
    scrie  $x_i$ 
  sfârșit-pentru
  pentru i=2,n execută
    cheamă P
  sfârșit-pentru
  pentru i=1,n execută
    scrie  $x_i$ 
  sfârșit-pentru
stop
```

procedura P este:

```

sw ← 0
y ←  $x_1$ 
  pentru j=i-1,1,-1 execută
    dacă  $y > x_j$  atunci  $x_{j+1} ← y$ 
                                     j ← 1
                                     sw ← 1
    altfel  $x_{j+1} ← x_j$ 
  sfârșit-dacă
  sfârșit-pentru
  dacă sw=0 atunci  $x_1 ← y$ 
  sfârșit-dacă
sfârșit-procedură
```

```

40 CLS
50 PRINT "*****"
60 PRINT "*   ORDONARE PRIN INSERTIE   *"
70 PRINT "*****"
80 INPUT "dimensiunea sirului n= ";n: LET n=ABS (INT n)
90 IF n=0 THEN GO TO 80
100
110 DIM x(n)
120
130 PRINT "sirul initial x de dimensiune ";n;" este:"
140
150 FOR i=1 TO n
160   INPUT "x(";i);"=";x(i)
170   PRINT x(i);" ";
180 NEXT i
190
```

```

200 PRINT ""SE OBTINE SUCCESIV"
210 FOR i=2 TO n
220 LET y=x(i): LET sw=0: LET j=i-1
230 IF j<=0 THEN GO TO 270
240 IF y<x(j) THEN LET x(j+1)=x(j): GO TO 260
250 LET x(j+1)=y: LET j=j-1: LET sw=1
260 LET j=j-1: GO TO 230
270 IF sw=0 THEN LET x(1)=y
280 GO SUB 330
290 NEXT i
300
310 PRINT ""sirul ordonat este:""
320 GO SUB 330: STOP
330 FOR j=1 TO n
340 PRINT x(j); " ";
350 NEXT j
360 PRINT : RETURN

```

```

*****
*   ORDONARE PRIN INSERTIE   *
*****

```

sirul initial x de dimensiune 10 este:

6 -9 1 0 3 -5 13 2 -1 8

SE OBTINE SUCCESIV

-9 6 1 0 3 -5 13 2 -1 8

-9 1 6 0 3 -5 13 2 -1 8

-9 0 1 6 3 -5 13 2 -1 8

-9 0 1 3 6 -5 13 2 -1 8

-9 -5 0 1 3 6 13 2 -1 8

-9 -5 0 1 3 6 13 2 -1 8

-9 -5 0 1 2 3 6 13 -1 8

-9 -5 -1 0 1 2 3 6 13 8

-9 -5 -1 0 1 2 3 6 8 13

sirul ordonat este:

-9 -5 -1 0 1 2 3 6 8 13

8.5.5 Considerăm cele două permutări ca fiind date prin două șiruri X și Y, ambele de dimensiune n, ceea ce înseamnă că:

$$p_1 = \begin{pmatrix} 1 & 2 & \dots & n \\ x_1 & x_2 & \dots & x_n \end{pmatrix} \quad \text{deci} \quad p_1^{-1} = \begin{pmatrix} x_1 & x_2 & \dots & x_n \\ 1 & 2 & \dots & n \end{pmatrix}$$

$$p_2 = \begin{pmatrix} 1 & 2 & \dots & n \\ y_1 & y_2 & \dots & y_n \end{pmatrix} \quad \text{de unde rezultă:}$$

$$s = p_1^{-1} \cdot p_2 = \begin{pmatrix} x_1 & x_2 & \dots & x_n \\ 1 & 2 & \dots & n \end{pmatrix} \cdot \begin{pmatrix} 1 & 2 & \dots & n \\ y_1 & y_2 & \dots & y_n \end{pmatrix}$$

Deci s_i este egal cu acel indice j pentru care $x_j = y_i$, pentru $i=1, 2, \dots, n$. Pentru a nu folosi alte șiruri în afara celor citite inițial, elementele s_i vor fi scrise pe măsura gășirii lor. De remarcat că în momentul introducerii celor două șiruri ce definesc permutările trebuie făcute verificările:

- a) șirul X este o mulțime?
 b) șirul Y este o mulțime?
 c) elementul Y_i aparține mulțimii X? ($i=1,2,\dots,n$).

algoritmul 8.5.5. este:

```

citește n
sw ← 0
execută
  pentru i=1,n execută
    citește  $x_i$ 
  sfârșit-pentru
cheamă P1
ieșim-cînd sw=0
execută
  sw2 ← 0
  pentru i=1,n execută
    sw ← 0
    execută
      citește  $y_i$ 
      cheamă P3
    ieșim-cînd sw=0
  sfârșit-pentru
cheamă P2
ieșim-cînd sw2=0
  pentru i=1,n execută
    scrie  $x_i$ 
  sfârșit-pentru
  pentru i=1,n execută
    scrie  $y_i$ 
  sfârșit-pentru
  pentru i=1,n execută
    j ← 1
    cît-timp  $x_j \neq x_i$  execută
      j ← j+1
    sfârșit-cît-timp
    scrie j
  sfârșit-pentru
stop
  
```

procedura P1 este:

```

pentru i=1,n execută
  pentru j=i+1,n execută
    dacă  $x_i = x_j$ 
      atunci sw ← 1
      j ← n
      i ← n
  sfârșit-dacă
sfârșit-pentru
sfârșit-procedură
  
```

procedura P2 este:

```

pentru i=1,n execută
  pentru j=i+1,n execută
    dacă  $y_i = y_j$ 
      atunci sw2 ← 1
      j ← n
      i ← n
  sfârșit-dacă
sfârșit-pentru
sfârșit-procedură
  
```

procedura P3 este:

```

sw1 ← 0
pentru j=1,n execută
  dacă  $y_i = x_j$ 
    atunci sw1 ← 1
    j ← n
  sfârșit-dacă
sfârșit-pentru
dacă sw1=0 atunci sw ← 1
sfârșit-dacă
sfârșit-procedură
  
```

```

10 REM
20 REM *** ECUATIE CU PERMUTARI ***
30 REM
40 CLS
50 PRINT "*****"
60 PRINT "* ECUATIE CU PERMUTARI *"
70 PRINT "*****"
80 INPUT "dimensiune siruri n=";n: LET n=ABS (INT n)
90 IF n=0 THEN GO TO 80
100
110 DIM x(n): DIM y(n)
120
130 FOR i=1 TO n
140 INPUT "sirul x este x(;"(i);")=";x(i)
150 NEXT i
160 LET sw=0: GO SUB 560
170 IF sw=1 THEN PRINT "nu este permutare!!! REPETA!!"
: GO TO 130
180
  
```

```

190 FOR i=1 TO n
200 INPUT "sirul y este y(";i);)="";y(i)
210 LET sw=0: GO SUB 720
220 IF sw=1 THEN PRINT "element incorect!!!REPETA!!!":
GO TO 200
230 NEXT i
240 LET sw=0: GO SUB 640
250 IF sw=1 THEN PRINT "permutare gresita !!!REPETA!!!"
: GO TO 190
260
270 PRINT "permutarea p1 este:"""
280 GO SUB 790
290 FOR i=1 TO n
300 LET v=x(i)
310 GO SUB 520
320 PRINT x(i);
330 NEXT i
340 PRINT
350 PRINT ""permutarea p2 este:"""
360 GO SUB 790
370 FOR i=1 TO n
380 LET v=y(i)
390 GO SUB 520
400 PRINT y(i);
410 NEXT i
420 PRINT
430
440 PRINT ""solutia este:"""
450 FOR i=1 TO n
460 LET j=1
470 IF x(j)=y(i) THEN PRINT j;" ";: GO TO 490
480 LET j=j+1: GO TO 470
490 NEXT i
500 PRINT : STOP
510
520 FOR u=1 TO 3-LEN (STR$ (v)): PRINT " ";: NEXT u
530 RETURN
540
550 REM *** verificare X este multime ? ***
560 FOR i=1 TO n
570 FOR j=i+1 TO n
580 IF x(i)=x(j) THEN LET sw=1: LET j=n: LET i=n
590 NEXT j
600 NEXT i
610 RETURN
620
630 REM *** verificare Y este multime ? ***
640 FOR i=1 TO n
650 FOR j=i+1 TO n
660 IF y(i)=y(j) THEN LET sw=1: LET j=n: LET i=n
670 NEXT j
680 NEXT i
690 RETURN
700
710 REM verificare elementul y(i) apartine multimii X?
720 LET sw1=0
730 FOR j=1 TO n
740 IF y(i)=x(j) THEN LET sw1=1: LET j=n
750 NEXT j
760 IF sw1=0 THEN LET sw=1
770 RETURN

```



```

790 FOR i=1 TO n
800 LET v=1
810 GO SUB 520
820 PRINT i;
830 NEXT i
840 PRINT : RETURN

```

```

*****
*      ECUATIE CU PERMUTARI      *
*****

```

permutarea p1 este:

```

1 2 3 4 5 6 7 8 9 10 11 12 13
5 2 10 4 1 13 3 8 7 9 12 6 11

```

permutarea p2 este:

```

1 2 3 4 5 6 7 8 9 10 11 12 13
9 7 3 5 12 4 11 13 10 8 2 6 1

```

solutia este:

```

10 9 7 1 11 4 13 6 3 8 2 12 5

```

8.5.6 a) Se calculează diferența primilor doi termeni. Dacă diferența este nulă, șirul nu este progresie aritmetică. Dacă nu este nulă se inițializează un switch cu zero, se parcurge șirul în continuare și se verifică pentru fiecare termen dacă se obține din precedentul adăugînd rația. La întîlnirea unui termen care nu satisface această condiție se modifică valoarea switch-ului și se iese forțat din structura repetitivă deoarece șirul nu este o progresie aritmetică. Dacă șirul este o progresie aritmetică ieșirea din structura repetitivă se face normal, switch-ul păstrînd valoarea inițială zero.

```

80 INPUT "dimensiune sir n=";n: LET n=ABS (INT n)
90 IF n<3 THEN GO TO 80
100
110 DIM x(n)
120
130 PRINT "sirul initial x este:"
140 FOR i=1 TO n
150 INPUT "x(";i);"=";x(i)
160 PRINT x(i);" ";
170 NEXT i
180
190 LET r=x(2)-x(1)
200 IF r=0 THEN PRINT ""NU!!!""sirul nu este progresie
aritmetica": GO TO 270
210 LET sw=0
220 FOR i=2 TO n-1
230 IF x(i+1)-x(i)<>r THEN LET sw=1: LET i=n-1
240 NEXT i
250 IF sw=0 THEN PRINT ""DA!!!""sirul este progresie a
ritmetica de ratie r=";r: GO TO 270
260 PRINT ""NU!!!""sirul nu este progresie aritmetica"
270 STOP

```

sirul initial x este:

-5 -3 -1 1 3 5 7 9 11

DA!!!

sirul este progresie aritmetica de ratie r=2

sirul initial x este:

9 8 7 6 4 3 2 1 0

NU!!!

sirul nu este progresie aritmetica

b) Soluția este analogă cu cea de la punctul a).
Trebuie avut însă în vedere că termenii unei progresii geometrice sînt obligatoriu nenuli.

algoritmul 8.5.6.b este:

```
citește n
pentru i=1,n execută
  citește xi
sfîrșit-pentru
pentru i=1,n execută
  scrie xi
sfîrșit-pentru
p ← x1·x2
dacă p=0
  atunci scrie 'X nu este
    progresie geometrică'
  altfel cheamă P
sfîrșit-dacă
stop
```

procedura P este:

```
q ← x2/x1
sw ← 0
pentru i=3,n execută
  dacă not (xi=xi-1·q)
    atunci sw ← 1
    i ← n
  sfîrșit-dacă
sfîrșit-pentru
dacă sw=0
  atunci scrie 'X este
    progresie geometrică'
  altfel scrie 'X nu este
    progresie geometrică'
sfîrșit-dacă
sfîrșit-procedură
```

```
50 PRINT "*****"
60 PRINT "*   PROGRESIE GEOMETRICA ?   *"
70 PRINT "*****"
80 INPUT "dimensiune sir n=";n: LET n=ABS (INT n)
90 IF n<3 THEN GO TO 80
100
110 DIM x(n)
120
130 PRINT "sirul initial x este:"
140 FOR i=1 TO n
150   INPUT "x(";i);")=";x(i)
160   PRINT x(i);" ";
170 NEXT i
180
190 LET p=x(2)*x(1)
200 IF p=0 THEN PRINT ""NU!!!""sirul nu este progresie
geometrica": GO TO 270
210 LET sw=0: LET q=x(2)/x(1)
220 FOR i=3 TO n
230   IF x(i)<>x(i-1)*q THEN LET sw=1: LET i=n
240 NEXT i
250 IF sw=0 THEN PRINT ""DA!!!""sirul este progresie g
eometrica de ratie q=";q: GO TO 270
260 PRINT ""NU!!!""sirul nu este progresie geometrica"
270 STOP
```

```

*****
*   PROGRESIE GEOMETRICA ?   *
*****

```

sirul initial x este:

1 2 4 8 16 32 64 128

DA!!!

sirul este progresie geometrica de ratie q=2

sirul initial x este:

2 4 8 16 32 64 128 256

NU!!!

sirul nu este progresie geometrica

8.5.7 Succesiunea cifrelor se păstrează într-o matrice cu 9 linii și 9 coloane, inițializată cu zero și care conține pe linia i cifrele ce pot apărea după cifra i și zero în continuare. Pentru fiecare element $a_{i,j}$ nenul se scrie numărul $y=100 \cdot i + 10 \cdot l + a_{i,k}$ pentru fiecare $a_{i,k}$ nenul ($l=a_{i,j}$).

algoritmul 8.5.7 este:

```

i ← 1
pentru i=1,9 execută
  citește k
  pentru j=1,k execută
    citește ai,j
  sfârșit-pentru
sfârșit-pentru
cheamă P
stop

```

procedura P este:

```

pentru i=1,9 execută
  pentru j=1,9 execută
    dacă ai,j=0 atunci j ← 9
    altfel cheamă P1
  sfârșit-dacă
sfârșit-pentru
sfârșit-procedură

```

procedura P1 este:

```

l ← ai,j
pentru k=1,9 execută
  dacă ai,k=0 atunci k ← 9
  altfel scrie i,l,ai,k
sfârșit-dacă
sfârșit-pentru
sfârșit-procedură

```

```

10 LET Z1=4: LET Z2=22: GO SUB 310
20 PRINT " * GENERARE DE NUMERE *"
30 PRINT " * CU TREI CIFRE *": GO SUB 310
40 DIM A(9,9)
50 PRINT "" " SUCCESIUNEA POSIBILA"" " DE CIFRE ESTE:"
60 FOR I=1 TO 9
70 PRINT " DUPA ";I;" URMEAZA ";
80 PRINT #0;"Cite cifre pot aparea dupa ";I;"?": PAUSE 50
90 INPUT K
100 FOR J=1 TO K
110 INPUT A(I,J): PRINT A(I,J);" ";
120 NEXT J
130 PRINT
140 NEXT I
150 PRINT "" " NUMERELE DE 3 CIFRE FORMATE"
160 PRINT " CU SUCCESIUNILE DATE SINT:""" " ";
170 LET K1=0
180 FOR I=1 TO 9
190 FOR J=1 TO 9
200 IF A(I,J)=0 THEN LET J=9: GO TO 270
210 LET L=A(I,J)
220 FOR K=1 TO 9
230 IF A(L,K)=0 THEN LET K=9: GO TO 260
240 PRINT I;L;A(L,K);" ";: LET K1=K1+1
250 IF K1=7 THEN PRINT "" "": LET K1=0
260 NEXT K
270 NEXT J
280 NEXT I
290 PAUSE 0: CLS : STOP
300
310 FOR W=1 TO Z1: PRINT " ";: NEXT W
320 FOR W=1 TO Z2: PRINT " *": NEXT W
330 PRINT : RETURN

```

```

*****
* GENERARE DE NUMERE *
* CU TREI CIFRE *
*****

```

SUCCESIUNEA POSIBILA

DE CIFRE ESTE:

DUPA 1 URMEAZA 4 7 8

DUPA 2 URMEAZA 1 3 7 8

DUPA 3 URMEAZA 9 5

DUPA 4 URMEAZA 9 5

DUPA 5 URMEAZA 1

DUPA 6 URMEAZA 1

DUPA 7 URMEAZA 1

DUPA 8 URMEAZA 5

NUMERELE DE 3 CIFRE FORMATE
CU SUCCESIUNILE DATE SINT:

149 145 171 185 214 217 218
239 235 271 285 396 397 398
399 351 496 497 498 499 451
514 517 518 614 617 618 714
717 718 851 961 971 985 996
997 998 999

8.5.8 Toate șirurile formate până la un moment dat sînt depuse într-o matrice A . Din această matrice se formează o nouă matrice B astfel: pentru fiecare linie a matricei A se testează dacă se mai poate adăuga un termen. Dacă nu, înseamnă că șirul este complet și el se tipărește. Dacă $a_{ik}+3 \leq 9$ (a_{ik} fiind ultimul element nenul al liniei i), atunci elementele liniei i se mută în două linii consecutive ale matricei B , apoi se completează cu cite un element, $a_{ik}+2$ și respectiv, $a_{ik}+3$. Dacă $a_{ik}=7$ atunci se completează șirul cu 9 și se scrie. Dacă în matricea B s-a depus cel puțin o linie procedeul continuă și pentru aceasta matricea B se mută în A .

algoritmul 8.5.8 este:

```

 $a_{1,1} \leftarrow 1$ 
execută
   $l \leftarrow 1$ 
   $k \leftarrow 1$ 
  cheamă  $P$ 
  [
    cît-timp  $n > 0$  execută
    [
      cheamă  $p$ 
      sfîrșit-cît-timp
       $a_{1,1} \leftarrow a_{1,1} + 1$ 
    ]
  ]
  ieșim-cînd  $a_{1,1} \neq 2$ 

```

stop

procedura P este:

```

 $n \leftarrow 0$ 
[
  pentru  $i=1, l$  execută
  [
    cheamă  $Q$ 
    sfîrșit-pentru
     $l \leftarrow n$ 
     $k \leftarrow k+1$ 
  ]
  pentru  $i=1, l$  execută
  [
    cheamă  $R$ 
    sfîrșit-pentru
  ]
]
sfîrșit-procedură

```

procedura R este:

```

[
  pentru  $j=1, k$  execută
  [
     $a_{i,j} \leftarrow b_{i,j}$ 
    sfîrșit-pentru
  ]
]
sfîrșit-procedură

```

```

procedura Q este:
  dacă  $a_{i,k}+3 \leq 9$  atunci  $n \leftarrow n+1$ 
    pentru  $j=1, k$  execută
       $b_{n,j} \leftarrow a_{i,j}$ 
       $b_{n+1,j} \leftarrow a_{i,j}$ 
    sfârșit-pentru
       $b_{n,k+1} \leftarrow a_{i,k}+2$ 
       $b_{n+1,k+1} \leftarrow a_{i,k}+3$ 
       $n \leftarrow n+1$ 
    altfel
      dacă  $a_{i,r}+2 \leq 9$  atunci  $m \leftarrow k+1$ 
         $a_{i,m} \leftarrow a_{i,k}+2$ 
        altfel  $m \leftarrow k$ 
      sfârșit-dacă
      pentru  $j=1, m$  execută
        scrie  $a_{i,j}$ 
      sfârșit-pentru
  sfârșit-dacă
sfârșit-procedură

```

```

40 PRINT "" PROGRAMUL FORMEAZA SIRURI DENUMERE NATURALE";
50 PRINT " CE SATISFAC ANU-MITE CONDITII DATE"
60 PRINT "" ACESTEA SINT:""
70 DIM A(20,10): DIM B(20,10)
80 LET S=0: LET A(1,1)=1
100 LET K=1: LET L=1
110 GO SUB 160
120 IF N>0 THEN GO TO 110
130 IF A(1,1)=1 THEN LET A(1,1)=2: GO TO 100
140 PAUSE 0: CLS : STOP
150
160 LET N=0
170 FOR I=1 TO L
180   IF A(I,K)+3<=9 THEN GO TO 270
190   IF A(I,K)+2>9 THEN LET M=K: GO TO 210
200   LET M=K+1: LET A(I,M)=A(I,K)+2
210   FOR J=1 TO M
220     PRINT A(I,J); " ";
230   NEXT J
240   PRINT " ";: LET SW=1-SW
250   IF SW=1 THEN PRINT
260   GO TO 330
270   LET N=N+1
280   FOR J=1 TO K
290     LET B(N,J)=A(I,J): LET B(N+1,J)=A(I,J)
300   NEXT J
310   LET B(N,K+1)=A(I,K)+2
320   LET B(N+1,K+1)=A(I,K)+3: LET N=N+1
330 NEXT I
340 LET L=N: LET K=K+1
350 FOR I=1 TO L
360   FOR J=1 TO K
370     LET A(I,J)=B(I,J)
380   NEXT J
390 NEXT I
400 RETURN
410

```

PROGRAMUL FORMEAZA SIRURI DE
NUMERE NATURALE CE SATISFAC ANU-
MITE CONDITII DATE

ACESTEA SINT:

1 4 7 9	1 3 5 7 9
1 3 5 8	1 3 6 8
1 3 6 9	1 4 6 8
1 4 6 9	2 4 7 9
2 5 7 9	2 5 8
2 4 6 8	2 4 6 9

8.5.9 După formarea șirului se citesc doi indici i și j . Dacă $i < j$ atunci notăm $l = (j-1)^2 + 2 \cdot j - i$ iar în caz contrar $l = (i-1)^2 + j$. În ambele cazuri $a_{l,i} = x_i$.

algoritmul 8.5.9. este:

```

citește n
pentru i=1, n execută
    pentru j=1, n execută
        citește ai,j
    sfîrșit-pentru
sfîrșit-pentru
x1 ← a1,1
m ← 1
pentru i=2, n execută
    cheamă P
sfîrșit-pentru
citeste i, j
dacă i < j atunci l ← (j-1)2 + 2 · j - i
    altfel l ← (i-1)2 + j
sfîrșit-dacă
scrie l, xl, al,i
stop
    
```

procedura P este:

```

pentru j=1, i execută
    m ← m+1
    xm ← al,j
sfîrșit-pentru
pentru k=i-1, 1, -1 execută
    m ← m+1
    xm ← ak,i
sfîrșit-pentru
sfîrșit-procedură
    
```

```

10 CLS : LET Z1=2: LET Z2=27: GO SUB 360
20 PRINT " * FORMARE DE SIR DIN *"
30 PRINT " * ELEMENTELE UNEI MATRICI *"
40 GO SUB 360: INPUT "Introduceti dim matricei ";N
50 DIM A(N,N): DIM X(N^2)
60 PRINT "" MATRICEA ESTE: ""
70 FOR I=1 TO N
80 FOR J=1 TO N
90 INPUT A(I,J): LET V=A(I,J): GO SUB 330
100 NEXT J
110 PRINT
120 NEXT I
130 PRINT "" SIRUL ESTE: ""
140 LET X(1)=A(1,1): LET M=1
150 LET V=A(1,1): GO SUB 330
160 FOR I=2 TO N
170 FOR J=1 TO I
180 LET M=M+1: LET X(M)=A(I,J): LET V=X(M): GO SUB 330
190 IF M/6=INT(M/6) THEN PRINT
200 NEXT J
    
```

```

210 FOR K=I-1 TO 1 STEP -1
220 LET M=M+1: LET X(M)=A(K,I): LET VV=X(M): GO SUB 330
230 IF M/6=INT (M/6) THEN PRINT
240 NEXT K: NEXT I
250 NEXT I
260 PRINT
270 INPUT "Introduceti indicii elementului cautat ";I,J
280 IF I<J THEN LET L=(J-1)^2+2*J-I: GO TO 300
290 LET L=(I-1)^2+J
300 PRINT " " A("I";",",";J")=X("L")="";X(L)
310 GO TO 270
320
330 FOR V=1 TO 5-LEN STR$(VV): PRINT " ";: NEXT V
340 PRINT VV;: RETURN
350
360 FOR V=1 TO Z1: PRINT " ";: NEXT V
370 FOR V=1 TO Z2: PRINT "*";
380 NEXT V
390 PRINT : RETURN

```

```

*****
* FORMARE DE SIR DIN *
* ELEMENTELE UNEI MATRICI *
*****

```

MATRICEA ESTE:

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16

SIRUL ESTE:

1	5	6	2	9	10
11	7	3	13	14	15
16	12	8	4		

$A(3,2)=X(6)=10$

$A(2,3)=X(8)=7$

8.5.10 Fie l indicele unui element oarecare al șirului și k partea întreagă a radicalului de ordinul doi al lui l iar $m=l-k$. Dacă $m > k+1$ atunci considerăm $j=k+1$ și $i=2 \cdot j-m$ iar în caz contrar dacă $m=0$ avem $i=1$ și $j=k$, altfel $i=k+1$ și $j=m$. Se obține egalitatea $x_l = a_{i,j}$.

procedura P este:

- [pentru $j=1, i$ execută
 - $m \leftarrow m+1$
 - $a_{i,j} \leftarrow x_m$
- [sfârșit-pentru
- [pentru $k=i-1, 1, -1$ execută
 - $m \leftarrow m+1$
 - $a_{k,i} \leftarrow x_m$
- [sfârșit-pentru
- [sfârșit-procedură

algoritmul 8.5.10. este:

```
execută
  citește m
   $n \leftarrow \sqrt{m}$ 
iesim-cînd  $n=[n]$ 
pentru  $i=1, m$  execută
  citește  $x_i$ 
sfîrșit-pentru
 $m \leftarrow 1$ 
 $a_{1,1} \leftarrow x_1$ 
pentru  $i=2, n$  execută
  cheamă P
sfîrșit-pentru
pentru  $i=1, n$  execută
  pentru  $j=1, n$  execută
    scrie  $a_{i,j}$ 
  sfîrșit-pentru
sfîrșit-pentru
citește l
 $k \leftarrow \lceil \sqrt{l} \rceil$ 
 $m \leftarrow l - k^2$ 
dacă  $m > k+1$  atunci  $j \leftarrow k+1$ 
  altfel  $i \leftarrow 2 \cdot j - m$ 
    dacă  $m=0$  atunci  $i \leftarrow 1$ 
      altfel  $i \leftarrow k+1$ 
    sfîrșit-dacă
sfîrșit-dacă
scrie  $x_i$ 
scrie  $a_{i,j}$ 
stop
```

```
10 CLS : LET Z1=2: LET Z2=27: GO SUB 410
20 PRINT " * FORMAREA UNEI MATRICI *"
30 PRINT " * DIN ELEMENTELE UNUI SIR *": GO SUB 410
40 INPUT "Introduceti dim sirului ";M: DIM X(M)
50 LET N=SQR (M): LET Q=N-INT n
60 IF ABS Q>0.01 THEN PRINT "DIMENSIUNE GRESITA": GO TO 40
70 PRINT "" " SIRUL ESTE:"""
80 FOR I=1 TO M
90 INPUT X(I)
100 LET VV=X(I): GO SUB 380
110 IF I/6=INT (I/6) THEN PRINT
120 NEXT I: PRINT
130 DIM A(N,N): LET M=1: LET A(1,1)=X(1)
140 FOR I=2 TO N
150 FOR J=1 TO I
160 LET M=M+1: LET A(I,J)=X(M)
170 NEXT J
180 FOR K=I-1 TO 1 STEP -1
190 LET M=M+1: LET A(K,I)=X(M)
200 NEXT K
210 NEXT I
220 PRINT "" " MATRICEA ESTE:"""
```

```

230 FOR I=1 TO N
240   FOR J=1 TO N
250     LET VV=A(I,J): GO SUB 380
260   NEXT J
270   PRINT
280 NEXT I
290 INPUT "Introduceti indicele elementului cautat ";I
300 LET K=INT (SOR (L)): LET M=L-K^2
310 IF M>K+1 THEN GO TO 340
320 IF M=0 THEN LET I=1: LET J=K: GO TO 350
330 LET I=K+1: LET J=M: GO TO 350
340 LET J=K+1: LET I=2*(K+1)-M
350 PRINT "X(";L;")=A(";I;",";J;")=";X(L)
360 PAUSE 0: CLS : STOP
370
380 FOR V=1 TO 5-LEN STR$ (VV): PRINT " ";: NEXT V
390 PRINT VV;: RETURN
400
410 FOR V=1 TO Z1: PRINT " ";: NEXT V
420 FOR V=1 TO Z2: PRINT "*";
430 NEXT V: PRINT : RETURN

```

```

*****
* FORMAREA UNEI MATRICI *
* DIN ELEMENTELE UNUI SIR *
*****

```

SIRUL ESTE:

1	2	3	4	5	6
7	8	9	10	11	12
13	14	15	16		

MATRICEA ESTE:

1	4	9	16
2	3	8	15
5	6	7	14
10	11	12	13

$X(8)=A(2,3)=8$

8.5.11 Primului element al șirului i se atribuie valoarea specificată în enunț. Pentru generarea celorlalte elemente se folosește o structură repetitivă care se încheie atunci când $l=1$, căci în acest caz nu mai există linii de indice mai mic decât l . Pentru a nu avea nevoie de două proceduri diferite, una pentru calculul maximului și alta pentru calculul minimului, se folosește o variabilă s ce va avea valoarea -1 când se calculează maximul și 1 când se calculează minimul și o comparație de tipul $s \cdot a_{ij} > s \cdot m$.

```

50 INPUT "Introduceti dim matrice ";N,M
60 DIM A(N,M)
70 PRINT / " MATRICEA ESTE:"; PRINT
80 FOR I=1 TO N
90 FOR J=1 TO M
100 GO SUB 410: LET VV=A(I,J): GO SUB 340
110 NEXT J
120 PRINT
130 NEXT I
140 INPUT "Introduceti indicii primului element ";L,K
150 PRINT / " L=";L;" K=";K
160 LET KK=1: DIM X(N): LET X(1)=A(L,K): LET SW=0
170 IF L=1 THEN GO TO 280
180 IF SW=0 THEN LET S=1: GO TO 200
190 LET S=-1
200 LET MM=A(1,1): LET L1=1
210 FOR I=1 TO L-1
220 FOR J=1 TO M
230 IF S*A(I,J)>S*MM THEN LET MM=A(I,J): LET L1=I
240 NEXT J
250 NEXT I
260 LET KK=KK+1: LET X(KK)=MM: LET SW=1-SW
270 LET L=L1: GO TO 170
280 PRINT / " SIRUL ESTE:"
290 FOR I=1 TO KK
300 LET VV=X(I): GO SUB 340
310 NEXT I
320 PAUSE 0: CLS : STOP
330
340 FOR V=1 TO 5-LEN STR$(VV): PRINT " ";: NEXT V
350 PRINT VV;: RETURN
410 REM Verificare ca elem sa fie 2 cite 2 distante
420 INPUT "A(";i);","; (j);")=";A(I,J)
430 LET SW=0
440 FOR L=1 TO I-1
450 FOR K=1 TO M
460 IF A(I,J)=A(L,K) THEN LET SW=1
470 NEXT K
480 NEXT L
490 FOR K=1 TO J-1
500 IF A(I,J)=A(I,K) THEN LET SW=1
510 NEXT K
520 IF SW=1 THEN GO TO 420
530 RETURN

```

MATRICEA ESTE:

3	6	8
5	9	2
1	10	21
24	-3	-8

L=3 K=2

SIRUL ESTE:

10 9 3

8.5.12 Această problemă este o problemă de grafuri și anume: dându-se matricea de adiacență a unui graf, să se găsească primul șir de indici în ordine lexicografică care reprezintă un lanț elementar și acest șir să fie de lungime maximă.

```

40 INPUT "Introduceti dim matricei ";N
50 DIM A(N,N): DIM X(N)
60 PRINT " " " MATRICEA ESTE:"""
70 FOR I=1 TO N
80 PRINT " ";
90 FOR J=1 TO N
100 INPUT "A(";(I);",";";(J);")=";A(I,J)
110 IF A(I,J)<>0 AND A(I,J)<>1 THEN GO TO 100
120 PRINT A(I,J);" ";
130 NEXT J
140 PRINT
150 NEXT I
160 LET L=0
170 FOR I=1 TO N
180 FOR J=1 TO N
190 IF A(I,J)=0 THEN GO TO 230
200 IF I=J THEN GO TO 230
210 LET L=L+1: LET X(L)=I: LET L=L+1: LET X(L)=J
220 LET J=N: LET I=N
230 NEXT J
240 NEXT I
250 LET I=X(L)
260 LET S2=1
270 FOR J=1 TO N
280 IF A(I,J)=0 THEN GO TO 360
290 LET S1=0
300 FOR K=1 TO L
310 IF J=X(K) THEN LET S1=1: LET K=L
320 NEXT K
330 IF S1<>0 THEN GO TO 360
340 LET L=L+1: LET X(L)=J: LET I=J: LET J=N
350 LET S2=0
360 NEXT J
370 IF S2<>1 THEN GO TO 260
380 PRINT " " " SIRUL FORMAT ESTE:"""
390 FOR K=1 TO L
400 LET VV=X(K): GO SUB 440
410 NEXT K
420 PAUSE 0: CLS : STOP
430
440 FOR V=1 TO 5-LEN STR$( VV): PRINT " ";: NEXT V
450 PRINT VV;; RETURN
460

```

MATRICEA ESTE:

0	1	1	1
1	0	1	0
1	0	1	1
1	0	0	0

SIRUL FORMAT ESTE:

1	2	3	4
---	---	---	---

8.5.13 Algoritmul este asemănător cu cel al ordonării elementelor unui șir cu deosebirea că în loc să se compare două elemente ale unui șir se compară primele elemente diferite a 2 linii și în loc să se inverseze două elemente ale unui șir se inversează două linii ale matricei.

```

10 CLS : LET Z1=5: LET Z2=22: GO SUB 390
20 PRINT " * MATRICE ORDONATA *": GO SUB 390
30 INPUT "Introduceti dim matricei "; N,M
40 DIM A(N,M)
50 PRINT "" " MATRICEA INITIALA ESTE:"
60 FOR I=1 TO N
70 FOR J=1 TO M
80 INPUT A(I,J): LET VV=A(I,J): GO SUB 360
90 NEXT J
100 PRINT
110 NEXT I
120 LET SW=1
130 IF SW<>1 THEN GO TO 270
140 LET SW=0
150 FOR I=1 TO N-1
160 LET J=1
170 IF J>M THEN GO TO 250
180 IF A(I,J)=A(I+1,J) THEN LET J=J+1: GO TO 170
190 IF A(I,J)<A(I+1,J) THEN GO TO 250
200 LET SW=1
210 FOR K=1 TO M
220 LET X=A(I,K): LET A(I,K)=A(I+1,K)
230 LET A(I+1,K)=X
240 NEXT K
250 NEXT I
260 GO TO 130
270 PRINT "" " MATRICEA REORDONATA ESTE:"""
280 FOR I=1 TO N
290 FOR J=1 TO M
300 LET VV=A(I,J): GO SUB 360
310 NEXT J
320 PRINT
330 NEXT I
340 PAUSE 0: CLS : STOP
350
360 FOR V=1 TO 5-LEN STR$(VV): PRINT " ";; NEXT V
370 PRINT VV;; RETURN
380
390 FOR V=1 TO Z1: PRINT " ";; NEXT V
400 FOR V=1 TO Z2: PRINT "*";;
410 NEXT V: PRINT : RETURN

```

```

*****
* MATRICE ORDONATA *
*****

```

MATRICEA INITIALA ESTE:

1	2	4	0
4	3	1	1
1	2	5	4
1	2	5	3
4	4	1	5
5	3	1	6
4	4	1	2

MATRICEA REORDONATA ESTE:

1	2	4	0
1	2	5	3
1	2	5	4
4	3	1	1
4	4	1	2
4	4	1	5
5	3	1	6

8.5.14 Elementele coloanelor de indice mai mare decit k se mută cu o poziție spre stanga, iar elementele liniilor de indice mai mare decit l se mută cu o poziție în sus.

algoritmul 8.5.14. este:

```

citește n,m
pentru i=1,n execută
  pentru j=1,m execută
    citește ai,j
  sfîrșit-pentru
sfîrșit-pentru
citește l,k
pentru i=1,n-1 execută
  pentru j=k,m-1 execută
    ai,j ← ai,j+1
  sfîrșit-pentru
sfîrșit-pentru
pentru i=1,n-1 execută
  cheamă Q
sfîrșit-pentru
n ← n-1
m ← m-1
pentru i=1,n execută
  pentru j=1,m execută
    scrie ai,j
  sfîrșit-pentru
sfîrșit-pentru
stop

```

procedura Q este:

```

pentru j=1,k-1 execută
  ai,j ← ai+1,j
sfîrșit-pentru
pentru j=k,m-1 execută
  ai,j ← ai+1,j+1
sfîrșit-pentru
sfîrșit-procedură

```

```

10 CLS : LET z1=4: LET z2=25: PRINT ' : GO SUB 460
20 PRINT " * SCOATEREA UNEI LINII *"
30 PRINT " * SI A UNEI COLOANE *"
40 PRINT " * DINTR-O MATRICE *"
50 GO SUB 460
60 PAUSE 0: CLS
70 INPUT "Introduceti dim matricei ";N,M
80 DIM A(N,M): DIM B(N,M)
90 PRINT "" MATRICEA ESTE:""
100 FOR I=1 TO N
110 FOR J=1 TO M
120 INPUT A(I,J): LET B(I,J)=A(I,J): LET VV=A(I,J)
130 GO SUB 430
140 NEXT J
150 PRINT
160 NEXT I
170 INPUT "Introduceti indicii liniei i coloanei ";L,K
180 FOR I=1 TO N-1
190 FOR J=K TO M-1
200 LET A(I,J)=A(I,J+1)
210 NEXT J
220 NEXT I

```

```

230 FOR I=L TO N-1
240   FOR J=1 TO K-1
250     LET A(I,J)=A(I+1,J)
260   NEXT J
270   FOR J=K TO M-1
280     LET A(I,J)=A(I+1,J+1)
290   NEXT J
300 NEXT I
310 LET N=N-1: LET M=M-1
320 PAUSE 0: CLS
330 PRINT "" DUPA SCOATEREA LINIEI ";L;" SI A"
340 PRINT " COLOANEI ";K;" SE OBTINE MATRI-    CEA:"""
350 FOR I=1 TO N
360   FOR J=1 TO M
370     LET VV=A(I,J): GO SUB 430
380   NEXT J
390 PRINT
400 NEXT I
410 PAUSE 0: CLS : STOP
430 FOR W=1 TO 4-LEN STR$(VV): PRINT " ";: NEXT W:PRINT VV;
440 RETURN
460 FOR V=1 TO Z1: PRINT " ";: NEXT V
470 FOR V=1 TO Z2: PRINT "*"": NEXT V
480 PRINT : RETURN

```

```

*****
* SCOATEREA UNEI LINII *
* SI A UNEI COLOANE *
* DINTR-O MATRICE *
*****

```

MATRICEA ESTE:

```

2   5   3   7
6   8   2   5
4   7   5   1
4   3   0   9
5   6   3   4

```

DUPA SCOATEREA LINIEI 4 SI A
COLOANEI 3 SE OBTINE MATRI-
CEA:

```

2   5   7
6   8   5
4   7   1
5   6   4

```

8.5.15

algoritmul 8.5.15. este:

citește n

dacă $n/2 = [n/2]$ atunci $l \leftarrow [n/2]$

altfel $l \leftarrow [n/2]+1$

sfârșit-dacă

pentru $i=1, n$ execută

cheamă P

sfârșit-pentru

pentru $k=1, l-1$ execută

cheamă Q

sfârșit-pentru

stop

procedura P este:

pentru $j=1, n$ execută

$a_{i,j} \leftarrow 0$

sfârșit-pentru

$a_{i,i} \leftarrow 1$

$a_{i,n+1-i} \leftarrow 1$

sfârșit-procedură

procedura Q este:

pentru $i=1, n-k$ execută

cheamă Q1

sfârșit-pentru

pentru $i=1, n$ execută

pentru $j=1, n$ execută

scrie $a_{i,j}$

sfârșit-pentru

sfârșit-pentru

sfârșit-procedură

procedura Q1 este:

$j \leftarrow i+k$

dacă $a_{i,j}=0$ atunci $a_{i,j} \leftarrow k+1$

sfârșit-dacă

dacă $a_{j,i}=0$ atunci $a_{j,i} \leftarrow k+1$

sfârșit-dacă

dacă $a_{i,n+1-j}=0$ atunci $a_{i,n+1-j} \leftarrow k+1$

sfârșit-dacă

dacă $a_{j,n+1-i}=0$ atunci $a_{j,n+1-i} \leftarrow k+1$

sfârșit-dacă

sfârșit-procedură

```
10 CLS : LET Z1=6: LET Z2=19: GO SUB 430
20 PRINT " * GENERAREA UNEI *"
30 PRINT " * MATRICI PATRATE *"
40 GO SUB 430
50 INPUT "Introduceti dim matricei "; N
60 DIM A(N,N)
70 IF N/2=INT (N/2) THEN LET L=INT (N/2): GO TO 90
80 LET L=INT (N/2)+1
90 FOR I=1 TO N
100 LET A(I,I)=1: LET A(I,N+1-I)=1
110 NEXT I
120 PRINT / " DUPA INTRODUCEREA LUI 1:""/
130 FOR I=1 TO N
140 PRINT " ";
150 FOR J=1 TO N
160 LET VV=A(I,J): GO SUB 400
170 NEXT J
180 PRINT
190 NEXT I: PRINT
200 FOR K=1 TO L-1
210 FOR I=1 TO N-K
220 LET J=I+K
230 IF A(I,J)=0 THEN LET A(I,J)=K+1
240 IF A(I,N+1-J)=0 THEN LET A(I,N+1-J)=K+1
250 IF A(J,I)=0 THEN LET A(J,I)=K+1
260 IF A(J,N+1-I)=0 THEN LET A(J,N+1-I)=K+1
270 NEXT I
280 PAUSE 0
290 PRINT / " DUPA INTRODUCEREA LUI ";K+1;"/:""/
300 FOR I=1 TO N
310 PRINT " ";
320 FOR J=1 TO N
330 LET VV=A(I,J): GO SUB 400
340 NEXT J
350 PRINT
360 NEXT I
```



```

370 NEXT K
380 PAUSE 0: CLS : STOP
390
400 FOR V=1 TO 4-LEN STR$(VV): PRINT " ";: NEXT V
410 PRINT VV;: RETURN
420
430 FOR U=1 TO Z1: PRINT " ";: NEXT U
440 FOR U=1 TO Z2: PRINT "*";: NEXT U
450 PRINT : RETURN

```

```

*****
* GENERAREA UNEI *
* MATRICI PATRATE *
*****

```

DUPA INTRODUCEREA LUI 1:

```

1 0 0 0 1
0 1 0 1 0
0 0 1 0 0
0 1 0 1 0
1 0 0 0 1

```

DUPA INTRODUCEREA LUI 2:

```

1 2 0 2 1
2 1 2 1 2
0 2 1 2 0
2 1 2 1 2
1 2 0 2 1

```

DUPA INTRODUCEREA LUI 3:

```

1 2 3 2 1
2 1 2 1 2
3 2 1 2 3
2 1 2 1 2
1 2 3 2 1

```

8.5.16 Se folosesc trei matrici A, B, C . In prima etapă se mută A în B și se efectuează produsul $C=A \cdot B$, in etapa următoare $B=AxC=A^3$ apoi $C=AxB=A^4$ etc.

procedura P este:

```

[ dacă sw=0 atunci cheamă PROD(c,b)
  altfel cheamă PROD(b,c)
  sfârșit-dacă
  sw ← 1-sw
sfârșit-procedură

```

procedura $PROD(x,y)$ este:

```

[ pentru i=1,n execută
  [ pentru j=1,n execută
    cheamă Q
  sfârșit-pentru
  sfârșit-pentru
sfârșit-procedură

```

procedura Q este:

```

s ← 0
[ pentru m=1,n execută
  s ← s+ai,m·xm,j
  sfârșit-pentru
yi,j ← s
sfârșit-procedură

```

```

algoritmul 8.5.16. este:
  citește n
  pentru i=1,n execută
  [
    pentru j=1,n execută
    [
      citește ai,j
    ]
    sfîrșit-pentru
  ]
  sfîrșit-pentru
  citește k
  b=a (* se face element cu element *)
  sw ← 0
  pentru l=1,k-1 execută
  [
    cheamă P
  ]
  sfîrșit-pentru
  dacă sw=0 atunci
  [
    pentru i=1,n execută
    [
      pentru j=1,n execută
      [
        scrie bi,j
      ]
      sfîrșit-pentru
    ]
    sfîrșit-pentru
  ]
  altfel
  [
    pentru i=1,n execută
    [
      pentru j=1,n execută
      [
        scrie ci,j
      ]
      sfîrșit-pentru
    ]
    sfîrșit-pentru
  ]
  sfîrșit-dacă
stop

```

```

40 INPUT "Introduceti dim matricei ";N: DIM A(N,N)
45 DIM B(N,N): DIM C(N,N)
50 INPUT "Introduceti puterea K:";K
60 PRINT " MATRICEA PATRATA ESTE"
70 FOR I=1 TO N
80 FOR J=1 TO N
90 INPUT A(I,J): LET B(I,J)=A(I,J)
100 LET VV=A(I,J): GO SUB 510
110 NEXT J
120 PRINT
130 NEXT I
140 LET SW=0
150 FOR L=1 TO K-1
160 FOR I=1 TO N
170 FOR J=1 TO N
180 LET S=0
190 FOR M=1 TO N
200 IF SW=0 THEN GO TO 220
210 LET S=S+A(I,M)*C(M,J): GO TO 230
220 LET S=S+A(I,M)*B(M,J)
230 NEXT M
240 IF SW=0 THEN LET C(I,J)=S: GO TO 260
250 LET B(I,J)=S
260 NEXT J
270 NEXT I
280 LET SW=1-SW
290 PRINT " PUTEREA A ";L+1;"-A ESTE:"
300 IF SW=0 THEN GO TO 380
310 FOR I=1 TO N
320 FOR J=1 TO N
330 LET VV=C(I,J): GO SUB 510
340 NEXT J
350 PRINT
360 NEXT I
370 GO TO 440

```

```

380 FOR I=1 TO N
390   FOR J=1 TO N
400     LET VV=B(I,J): GO SUB 510
410   NEXT J
420   PRINT
430 NEXT I
440 NEXT L
510 FOR Q=1 TO 5-LEN STR$ VV
520   PRINT " ";
530 NEXT Q
540 PRINT VV;: RETURN

```

MATRICEA PATRATA ESTE

1	4	3	7
6	8	5	6
4	5	3	4
2	5	4	7

PUTEREA A 2-A ESTE:

51	86	60	92
86	143	97	152
54	91	62	98
62	103	71	109

PUTEREA A 3-A ESTE:

991	1652	1131	1757
1636	2733	1872	2912
1044	1744	1195	1858
1182	1972	1350	2099

8.6.1 Având două șiruri ce reprezintă mulțimi, obținute din șirurile inițiale X și Y prin eliminarea elementelor ce se repetă, se formează un șir conținând elementele intersecției $X \cap Y$ și alt șir cu elementele diferenței $X \setminus Y$. Pentru reuniune se folosește formula $X \cup Y = (X \setminus Y) \cup Y$. ○

```

10 PRINT " *****"
20 PRINT " *   MULTIMI REPREZENTATE   *"
30 PRINT " *           CA SIRURUI     *"
40 PRINT " *           OPERATII CU ELE  *"
50 PRINT " *****"
60 INPUT "Introduceti dim sirurilor ";n,m
70 DIM x(n): DIM y(n+m): DIM a(n+m)
80 PRINT "PRIMUL SIR ESTE:""{" ";
90 FOR i=1 TO n
100  INPUT x(i): LET a(i)=x(i): LET vv=x(i): GO SUB 820
110 NEXT i
120 PRINT " )"
130 LET k=n: GO SUB 720
140 LET k1=1
150 IF k1=n THEN PRINT "SIRUL REPREZINTA O MULTIME": GO TO 220
160 PRINT "OBTINEM MULTIMEA:""{" ";
170 FOR i=1 TO k1
180  LET vv=a(i): GO SUB 820
190  LET x(i)=a(i)
200 NEXT i
210 PRINT " }"

```

```

220 PRINT "AL 2-LEA SIR ESTE" "(" ;
230 FOR i=1 TO m
240   INPUT y(i): LET a(i)=y(i): LET vv=y(i): GO SUB 820
250 NEXT i
260 PRINT " )"
270 LET k=m
280 GO SUB 720
290 LET k2=1
300 IF k2=m THEN PRINT "SIRUL REPREZINTA O MULTIME": GO TO 360
310 PRINT "OBTINEM MULTIMEA:" "(" "{" ;
320 FOR i=1 TO k2
330   LET vv=a(i): GO SUB 820
340 NEXT i
350 PRINT " }";
360 FOR i=1 TO k1
370   LET a(k2+i)=x(i)
380 NEXT i
390 LET k=k1+k2: GO SUB 720
400 PRINT "REUNIUNEA ESTE:" "(" "{" ;
410 FOR i=1 TO l
420   LET vv=a(i): GO SUB 820
430 NEXT i
440 PRINT " }"
450 DIM b(n+m): DIM c(n+m)
460 LET l=0: LET k=0
470 FOR i=1 TO k1
480   LET sw=0
490   FOR j=1 TO k2
500     IF x(i)=y(j) THEN LET sw=1: LET j=k2
510   NEXT j
520   IF sw=1 THEN LET l=l+1: LET b(l)=x(i): GO TO 540
530   LET k=k+1: LET c(k)=x(i)
540 NEXT i
550 IF l=0 THEN GO TO 620
560 PRINT "INTERSECTIA MULTIIMILOR ESTE" "(" "{" ;
570 FOR i=1 TO l
580   LET vv=b(i): GO SUB 820
590 NEXT i
600 PRINT " }"
610 GO TO 650
620 PRINT "INTERSECTIA ESTE VIDA"
630 PRINT "DIFERENTA ESTE PRIMA MULTIME"
640 GO TO 700
650 PRINT "DIFERENTA ESTE" "(" "{" ;
660 FOR i=1 TO k
670   LET vv=c(i): GO SUB 820
680 NEXT i
690 PRINT " }"
700 PAUSE 0: CLS : STOP
720 LET l=1
730 FOR i=2 TO k
740   LET sw=0
750   FOR j=1 TO i-1
760     IF a(i)=a(j) THEN LET sw=1: LET j=i-1
770   NEXT j
780   IF sw=0 THEN LET l=l+1: LET a(l)=a(i)
790 NEXT i
800 RETURN
820 FOR q=1 TO 4-LEN STR$ vv: PRINT " ";: NEXT q
830 PRINT vv;
840 RETURN
220

```


PRIMUL SIR ESTE:
 (2 3 2 4)
 OBTINEM MULTIMEA:
 { 2 3 4 }
 AL 2-LEA SIR ESTE
 (3 4 5 1 6)
 SIRUL REPEZINTA O MULTIME

 REUNIUNEA ESTE:
 { 3 4 5 1 6 2 }
 INTERSECTIA MULTIMILOR ESTE
 { 3 4 }
 DIFERENTA ESTE
 { 2 }

PRIMUL SIR ESTE:
 (2 3 4)
 SIRUL REPEZINTA O MULTIME
 AL 2-LEA SIR ESTE
 (1 5 6 5)
 OBTINEM MULTIMEA:
 { 1 5 6 }
 REUNIUNEA ESTE:
 { 1 5 6 2 3 4 }
 INTERSECTIA ESTE VIDA
 DIFERENTA ESTE PRIMA MULTIME

8.6.2 Se formează un șir X cu punctele intermediare b , $b+h$, $b+2h, \dots, b+(n-1)h$, și o matrice D ce conține pe linia j elementele mulțimii A_j . Concomitent se formează un șir cu cardinalele acestor mulțimi.

procedura P este:
 [dacă $x_i > a$ atunci $a \leftarrow x_i$
 [sfârșit-dacă
 [dacă $x_i < b$ atunci $b \leftarrow x_i$
 [sfârșit-dacă
 sfârșit-procedură

procedura Q este:
 1 \leftarrow 0
 [pentru $i=1, n$ execută
 cheamă $Q1$
 [sfârșit-pentru
 c \leftarrow 1
 sfârșit-procedură

procedura S este:
 max \leftarrow c_1
 [pentru $j=2, m$ execută
 [dacă $c_j > \text{max}$
 [atunci max \leftarrow c_j
 [sfârșit-dacă
 sfârșit-pentru
 [pentru $j=1, m$ execută
 cheamă $S1$
 [sfârșit-pentru
 sfârșit-procedură

algoritmul 8.6.2 este:

citește n
pentru $i=1, n$ execută
citește x_i

sfârșit-pentru
 $a \leftarrow x_1$

$b \leftarrow a$

pentru $i=2, n$ execută
cheamă P

sfârșit-pentru
citește m
 $m \leftarrow m+1$

$h \leftarrow \frac{a-b}{m}$

pentru $j=1, m-1$ execută
cheamă Q

sfârșit-pentru
 $l \leftarrow 0$

pentru $i=1, n$ execută
cheamă R

sfârșit-pentru
 $c_m \leftarrow l$

cheamă S

stop

procedura R este:

dacă $x_i \geq a-h$ atunci $l \leftarrow l+1$

$d_{m,l} \leftarrow x_i$

sfârșit-dacă

sfârșit-procedură

procedura S este:

dacă $c_j = \max$

atunci pentru $l=1, c_j$ execută

scrie $d_{j,l}$

sfârșit-pentru

sfârșit-dacă

sfârșit-procedură

procedura Q este:

dacă $x_i \geq b+(j-1) \cdot h$

atunci dacă $x_i < b+j \cdot h$

atunci $l \leftarrow l+1$

$d_{j,l} \leftarrow x_i$

sfârșit-dacă

sfârșit-dacă

sfârșit-procedură

```
10 CLS : LET Z1=9: LET Z2=13: GO SUB 670
20 PRINT " * INTERVALE *"
30 GO SUB 670: PRINT
40 INPUT "Introduceti cardinalul multimii";N: DIM X(N)
50 PRINT " MULTIMEA ESTE:" X="";
60 FOR I=1 TO N
70 INPUT X(I)
80 LET VV=X(I): GO SUB 630
90 IF I/5=INT(I/5) THEN PRINT " ";
100 NEXT I: PRINT " "
110 LET A=X(1): LET B=A
120 FOR I=2 TO N
130 IF X(I)>A THEN LET A=X(I)
140 IF X(I)<B THEN LET B=X(I)
150 NEXT I
160 INPUT "Introduceti nr de pct interm ";M: LET M=M+1
170 DIM C(M): DIM D(M,N): LET H=(A-B)/M
180 PRINT " LUNGIMEA INTERVALELOR:";H
190 PRINT " PUNCTELE INTERMEDIARE :";
200 LET VV=B: GO SUB 630
210 FOR I=1 TO M
220 LET VV=B+I*H: GO SUB 630
230 IF (I+1)/6=INT((I+1)/6) THEN PRINT " ";
240 NEXT I: PRINT
250 FOR J=1 TO M-1
260 LET L=0
270 FOR I=1 TO N
280 IF X(I)<B+(J-1)*H THEN GO TO 310
290 IF X(I)>=B+J*H THEN GO TO 310
300 LET L=L+1: LET D(J,L)=X(I)
310 NEXT I
320 LET C(J)=L
330 NEXT J
```

```

340 LET L=0
350 FOR I=1 TO N
360 IF X(I)<A-H THEN GO TO 380
370 LET L=L+1: LET D(M,L)=X(I)
380 NEXT I
390 LET C(M)=L
400 PRINT " " SUBMULTIMILE SINT:""
410 FOR J=1 TO M
420 IF C(J)=0 THEN GO TO 470
430 PRINT " {"
440 FOR L=1 TO C(J)
450 LET VV=D(J,L): GO SUB 630
460 NEXT L: PRINT " }"
470 NEXT J
480 LET MAX=C(1)
490 FOR J=2 TO M
500 IF C(J)>MAX THEN LET MAX=C(J)
510 NEXT J
520 PRINT " " CARDINALUL MAXIM=":MAX
530 PRINT " " SUBMULT. DE CARDINAL MAXIM:"
540 FOR J=1 TO M
550 IF C(J)<>MAX THEN GO TO 600
560 PRINT " {":: FOR L=1 TO C(J): LET VV=D(J,L)
570 FOR L=1 TO C(J)
580 LET VV=D(J,L): GO SUB 630
590 NEXT L: PRINT " }"
600 NEXT J
610 PAUSE 0: CLS : STOP
620
630 FOR Z=1 TO 5-LEN STR$ (VV): PRINT " " : NEXT Z
640 PRINT VV:
650 RETURN
660
670 FOR V=1 TO Z1: PRINT " " : NEXT V
680 FOR V=1 TO Z2: PRINT " * " : NEXT V
690 PRINT : RETURN

```

```

*****
* INTERVALE *
*****

```

MULTIMEA ESTE:

```

X={ 10 26 83 110 13
    7 2 20 45 74
    125 146 }

```

LUNGIMEA INTERVALELOR:12

PUNCTELE INTERMEDIARE :

```

 2 14 26 38 50 62
74 86 98 110 122 134
146

```

SUBMULTIMILE SINT:

```

{ 10 13 7 2 }
{ 20 }
{ 26 }
{ 45 }
{ 83 74 }
{ 110 }
{ 125 }
{ 146 }

```

CARDINALUL MAXIM=4

SUBMULT. DE CARDINAL MAXIM:

```

{ 10 13 7 2}

```

8.6.3 Produsul cartezian poate fi considerat o matrice cu m coloane (m fiind numărul de mulțimi) și atâtea linii câte elemente conține produsul cartezian. O primă metodă de rezolvare este următoarea: se formează cite un șir cu elementele fiecărei coloane a matricei, mulțimile inițiale obținându-se prin eliminarea elementelor ce se repetă în aceste șiruri. Dezavantajul acestei metode constă în faptul că elementele se pot repeta de un număr foarte mare de ori. Cel de-al doilea algoritm este următorul: se rearanjează liniile matricei (adică elementele produsului cartezian) ca în problema 8.5.13. Se parcurge matricea astfel ordonată pe coloană. Atâta vreme cât elementele primelor coloane coincid ultima coloană conține ultima mulțime. Celelalte mulțimi se obțin luind pe sărite elementele din coloanele corespunzătoare lor.

```

10 PRINT ""
20 PRINT " *****"
30 PRINT " * DIN PRODUSUL CARTEZIAN *"
40 PRINT " * DEDUCEM MULTIMILE *"
50 PRINT " *****"
60 PAUSE 0: CLS
70 INPUT "Introduceti nr. de multimi:";M
80 INPUT "Introduceti nr. de elem. ale prod. cartezian:";N
90 PRINT "" "PRODUSUL CARTEZIAN ESTE""
100 DIM A(N,M): DIM X(N): DIM B(N)
110 LET KK=0
120 FOR I=1 TO N
130 PRINT "(";
140 FOR J=1 TO M
150 INPUT A(I,J): LET VV=A(I,J): GO SUB 300
160 NEXT J
170 PRINT " )";
180 LET KK=1-KK
190 IF KK=0 THEN PRINT
200 NEXT I
210 PRINT "" "MULTIMILE SINT:"""
220 FOR J=1 TO M
230 FOR I=1 TO N
240 LET X(I)=A(I,J)
250 NEXT I
260 GO SUB 330
270 NEXT J

```



```

280 PAUSE 0: CLS : STOP
290
300 FOR V=1 TO 4-LEN STR$(VV): PRINT " ";: NEXT V
310 PRINT VV;: RETURN
320
330 PRINT "{";: LET VV=X(1): GO SUB 300
340 DIM B(N): LET L=0
350 FOR U=2 TO N
360   LET SW=0
370   FOR Z=1 TO U-1
380     IF X(U)=X(Z) THEN LET SW=1: LET Z=U
390   NEXT Z
400   IF SW=0 THEN LET L=L+1: LET B(L)=X(U)
410 NEXT U
420 FOR U=1 TO L
430   LET VV=B(U): GO SUB 300
440 NEXT U
450 PRINT "  }"
460 RETURN

```

```

*****
* DIN PRODUSUL CARTEZIAN *
*   DEDUCEM MULTIMILE   *
*****

```

PRODUSUL CARTEZIAN ESTE

```

( 1  4  5 )( 2  3  5 )
( 1  3  5 )( 2  4  5 )

```

MULTIMILE SINT:

```

{ 1  2  }
{ 4  3  }
{ 5  }

```

8.6.4 După citirea mulțimii ca șir X cu k elemente, se formează un șir S tot de dimensiune k cu toate elementele nule. Pentru fiecare dintre cele m submulțimi se citește cardinalul ei și apoi elementele sale. Acestea nu se rețin (deci se pot citi în aceeași variabilă b) însă pentru fiecare element citit se găsește indicele i astfel încât $x_i = b$ și se mărește s_i cu 1. După citirea tuturor celor m submulțimi elementele s_i arată câte din cele m submulțimi conțin elementul x_i .

procedura P este:

```

  citește a
  pentru j=1,a execută
  cheamă P1
  sfârșit-pentru
sfârșit-procedură

```

procedura $P2$ este:

```

  dacă  $x_i = b$  atunci  $s_i \leftarrow s_i + 1$ 
  sfârșit-dacă
sfârșit-procedură

```

procedura $P1$ este:

```

  citește b
  pentru l=1,k execută
  cheamă P2
  sfârșit-pentru
sfârșit-procedură

```

```

algoritmul 8.6.4. este:
  citește k,m,r
  pentru i=1,k execută
    citește  $x_i$ 
  sfîrșit-pentru
  pentru i=1,k execută
     $s_i \leftarrow 0$ 
  sfîrșit-pentru
  pentru i=1,m execută
    cheamă P
  sfîrșit-pentru
    n  $\leftarrow$  0
  pentru i=1,k execută
    dacă  $s_i=r$  atunci n  $\leftarrow$  n+1
    sfîrșit-dacă
  sfîrșit-pentru
    dacă n $\neq$ 0 atunci
      pentru i=1,k execută
        dacă  $s_i=r$  atunci scrie  $x_i$ 
        sfîrșit-dacă
      sfîrșit-pentru
        altfel scrie 'nu sînt'
    sfîrșit-dacă
  stop

```

```

10 PRINT "": LET Z1=6: LET Z2=16: GO SUB 460
20 PRINT "      * M SUBMULTIMI *"
30 GO SUB 460
40 INPUT "INTR. NR. DE ELEM. ALE MULTIMII ";K
50 PRINT "      MULTIMEA ESTE:": PRINT "      {";
60 PRINT #0;"Introduceti elem. mult.": PAUSE 100
70 DIM X(K): DIM S(K)
80 LET NN=5
90 FOR I=1 TO K
100  INPUT X(I): LET VV=X(I)
110  GO SUB 410: LET S(I)=0
120  NEXT I: PRINT " }"
130  INPUT "Introduceti nr. submult. si nr. K";M,R
140  FOR I=1 TO M
150    LET NN=5
160    PRINT "      SUBMULTIMEA ";I;" ESTE:"" {";
170    INPUT "Introduceti card. submult.":A
180    PRINT #0;"Introduceti elem. submult.": PAUSE 50
190    FOR J=1 TO A
200      INPUT B: LET VV=B
210      GO SUB 410
220      FOR L=1 TO K
230        IF X(L)=B THEN LET S(L)=S(L)+1
240      NEXT L
250    NEXT J: PRINT " }"
260  NEXT I: PRINT
270  LET N=0
280  FOR I=1 TO K
290    IF S(I)=R THEN LET N=N+1
300  NEXT I
310  IF N=0 THEN GO TO 500
320  PRINT "      IN ";R;" SUBMULTIMI ";
330  IF N=1 THEN PRINT "ESTE ELEMEN- TUL ";: GO TO 360
340  PRINT "SINT ELEMEN- TELE:";
350  LET NN=0

```

```

360 FOR I=1 TO K
370 IF S(I)=R THEN LET VV=X(I): GO SUB 410
380 NEXT I
390 GO TO 510
400
410 FOR V=1 TO 3-LEN STR$(VV): PRINT " ";: NEXT V
420 PRINT VV;: LET NN=NN+3
430 IF NN>=27 THEN LET NN=4: PRINT " ";
440 RETURN
450
460 FOR U=1 TO Z1: PRINT " ";: NEXT U
470 FOR U=1 TO Z2: PRINT "*";: NEXT U
480 PRINT : RETURN
490
500 PRINT " NU SINT ELEMENTE IN ";R;" SUB- MULTIMI"
510 PAUSE 0: CLS : STOP

```

```

*****
* M SUBMULTIMI *
*****

```

MULTIMEA ESTE:

```
{ 3 7 19 14 32 21 }
```

SUBMULTIMEA 1 ESTE:

```
{ 3 7 19 14 32 21 }
```

SUBMULTIMEA 2 ESTE:

```
{ 3 19 21 }
```

SUBMULTIMEA 3 ESTE:

```
{ 14 32 }
```

SUBMULTIMEA 4 ESTE:

```
{ 3 19 14 21 }
```

SUBMULTIMEA 5 ESTE:

```
{ 14 7 32 }
```

IN 3 SUBMULTIMI SINT ELEMEN-
TELE: 3 19 32 21

8.6.5 Fie X șirul de dimensiune n ce conține elementele mult.ii. Se formează un șir Y tot de dimensiune n , inițializat cu zero, șir în care se vor forma, prin adunare cu 1 în baza doi, toți vectorii caracteristici. Pentru fiecare vector caracteristic astfel generat se vor tipări acele elemente x_i pentru care $y_i=1$

procedura Q este:

```
k ← 0
```

```
pentru i=n,1,-1 execută
```

```
    dacă  $y_i \neq 0$  atunci  $y_i \leftarrow 0$ 
```

```
        altfel  $y_i \leftarrow 1$ 
```

```
        k ← 1
```

```
        i ← 1
```

```
    sfîrșit-dacă
```

```
sfîrșit-pentru
```

```
sfîrșit-procedură
```

algoritmul 8.6.5. este:

citește n

pentru $i=1,n$ execută

 citește x_i

sfîrșit-pentru

pentru $i=1,n$ execută

$y_i \leftarrow 0$

sfîrșit-pentru

 cheamă Q

 execută

 cheamă P

 cheamă Q

 ieșim-cînd $k=0$

stop

procedura P este:

pentru $i=1,n$ execută

 dacă $y_i=1$ atunci scrie x_i

 sfîrșit-dacă

sfîrșit-pentru

sfîrșit-procedură

```

10 PRINT ///
20 PRINT " *****"
30 PRINT " * GENERARE DE SUBMULTIMI *"
40 PRINT " * FOLOSIND VECTORUL *"
50 PRINT " * CARACTERISTIC *"
60 PRINT " *****"
70 INPUT "Introduceti cardinalul multimii ";n
80 DIM x(n): DIM y(n)
90 PRINT ///" MULTIMEA ESTE:"
100 PRINT " {";
110 FOR i=1 TO n
120 INPUT x(i): LET vv=x(i): GO SUB 300
130 NEXT i
140 PRINT " }""
150 PRINT " SUBMULTIMILE SINT:"""
160 LET k=0
170 FOR i=n TO 1 STEP -1
180 IF y(i)<>0 THEN LET y(i)=0: GO TO 200
190 LET y(i)=1: LET k=1: LET i=1
200 NEXT i
210 IF k=0 THEN GO TO 280
220 PRINT " {";
230 FOR i=1 TO n
240 IF y(i)=1 THEN LET vv=x(i): GO SUB 300
250 NEXT i
260 PRINT " }"
270 GO TO 160
280 PAUSE 0: CLS : STOP
290
300 FOR q=1 TO 4-LEN STR$ vv: PRINT " ";: NEXT q
310 PRINT vv;
320 RETURN

```

```

*****
* GENERARE DE SUBMULTIMI *
* FOLOSIND VECTORUL *
* CARACTERISTIC *
*****

```

MULTIMEA ESTE:
{ -1 0 7 -17 }

SUBMULTIMILE SINT:

```

{ -17 }
{ 7 }
{ 7 -17 }
{ 0 }
{ 0 -17 }
{ 0 7 }
{ 0 7 -17 }
{ -1 }
{ -1 -17 }
{ -1 7 }
{ -1 7 -17 }
{ -1 0 }
{ -1 0 -17 }
{ -1 0 7 }
{ -1 0 7 -17 }

```


8.6.6 Se folosește un șir Y de aceeași dimensiune n ca și șirul X ce conține elementele mulțimii. Șirul Y va conține indicii elementelor mulțimii X ce aparțin unei submulțimi și în rest zerouri. Se pornește de la șirul Y cu toate elementele nule, șir ce corespunde mulțimii vide. Având șirul Y se generează următorul astfel:

-se determină cel mai mare indice i ce satisface proprietățile:
 $y_i < i, \quad y_{i+1} = i+1 \quad \dots \quad y_{n-1} = n-1, \quad y_n = n.$

Șirul ce urmează lexicografic este:

$(y_1, \dots, y_{i-1}, y_i+1, y_i+2, \dots, y_i+n-i+1).$

Dacă nu există nici un indice i cu proprietățile cerute, înseamnă că șirul este $Y=(1,2,\dots,n)$, șir ce corespunde mulțimii totale, deci au fost deja generate toate submulțimile.

algoritmul 8.6.6. este:

```

citește n
pentru i=1, n execută
    citește  $x_i$ 
sfârșit-pentru
pentru i=1, n execută
     $y_i \leftarrow 0$ 
sfârșit-pentru
 $k \leftarrow 0$ 
pentru i=n, 1, -1 execută
    dacă  $y_i \neq i$  atunci  $k \leftarrow i$ 
         $i \leftarrow 1$ 
    sfârșit-dacă
sfârșit-pentru
cît-timp  $k \neq 0$  execută
    cheamă Q
     $k \leftarrow 0$ 
    pentru i=n, 1, -1 execută
        dacă  $y_i \neq i$  atunci  $k \leftarrow i$ 
             $i \leftarrow 1$ 
        sfârșit-dacă
    sfârșit-pentru
sfârșit-cît-timp
stop

```

procedura Q este:

```

 $y_k \leftarrow y_k + 1$ 
pentru i=k+1, n execută
     $y_i \leftarrow y_{i-1} + 1$ 
sfârșit-pentru
pentru i=1, n execută
    dacă  $y_i \neq 0$ 
        atunci  $j \leftarrow y_i$ 
        scrie  $x_j$ 
    sfârșit-dacă
sfârșit-pentru
sfârșit-procedură

```

```

10 PRINT ""
20 PRINT " *****"
30 PRINT " * GENERARE DE SUBMULTIMI *"
40 PRINT " * FOLOSIND SIRURI ORDONATE *"
50 PRINT " * LEXICOGRAFIC *"
60 PRINT " *****"
70 INPUT "Introduceti cardinalul multimii "; n
80 DIM x(n): DIM y(n)
90 PRINT "" MULTIMEA ESTE: ""
100 PRINT " {"
110 FOR i=1 TO n
120 INPUT x(i): LET vv=x(i): GO SUB 350
130 NEXT i
140 PRINT " }""
150 PRINT " SUBMULTIMILE SINT: ""
160 LET k=0
170 FOR i=n TO 1 STEP -1
180 IF y(i)<>i THEN LET k=i: LET i=1
190 NEXT i

```

```

200 IF k=0 THEN GO TO 330
210 IF k=n THEN LET y(n)=y(n)+1: GO TO 260
220 LET y(k)=y(k)+1
230 FOR i=k+1 TO n
240 LET y(i)=y(i-1)+1
250 NEXT i
260 PRINT "  {";
270 FOR i=1 TO n
280 IF y(i)=0 THEN GO TO 300
290 LET vv=x(y(i)): GO SUB 350
300 NEXT i
310 PRINT " }"
320 GO TO 160
330 PAUSE 0: CLS : STOP
340
350 FOR q=1 TO 4-LEN STR$ vv: PRINT " "; NEXT q
360 PRINT vv;
370 RETURN

```

```

*****
* GENERARE DE SUBMULTIMI *
* FOLOSIND SIRURI ORDONATE *
* LEXICOGRAFIC *
*****

```

MULTIMEA ESTE:
{ -1 0 7 -17 }

SUBMULTIMILE SINT:

```

{ -1 }
{ 0 }
{ 7 }
{ -17 }
{ -1 0 }
{ -1 7 }
{ -1 -17 }
{ 0 7 }
{ 0 -17 }
{ 7 -17 }
{ -1 0 7 }
{ -1 0 -17 }
{ -1 7 -17 }
{ 0 7 -17 }
{ -1 0 7 -17 }

```

8.6.7 Fiind dată o permutare p a mulțimii $\{1, 2, \dots, n\}$ i se atașează un vector de inversiune avind n componente definite în felul următor: elementul de ordinul k al său reprezintă numărul componentelor lui p situate înaintea lui p_k și mai mari decît el. Se observă că pentru $k=1, 2, \dots, n-1$ elementul p_k este numărul natural cu proprietatea că există exact i_k numere mai mari decît el și egale cel mult cu n , necuprinse în $\{p_{k+1}, \dots, p_n\}$. Această observație permite generarea unei permutări pornind de la vectorul său de inversiune. Deci generarea permutărilor mulțimii $\{1, 2, \dots, n\}$ poate fi făcută astfel: se

generează pe rind vectorii $I \in D_1 \cdot D_2 \dots D_n$, unde $D_k = \{0, 1, \dots, k-1\}$ și din fiecare vector astfel generat se obține permutarea corespunzătoare.

```

70 INPUT "n=";n
80 LET pp=1
90 FOR i=1 TO n
100 LET pp=pp*i
110 NEXT i
120 LET kk=0
130 DIM a(n): DIM v(n): DIM p(n)
140 LET k1=0
150 LET i=n
160 IF kk=pp THEN PAUSE 0: CLS : STOP
170 IF a(i)<i THEN GO SUB 250: LET a(i)=a(i)+1: GO TO 160
180 LET i=i-1
190 IF a(i)>=i-1 THEN GO TO 180
200 LET a(i)=a(i)+1
210 FOR j=i+1 TO n
220 LET a(j)=0
230 NEXT j
240 GO TO 150
250 LET kk=kk+1
260 FOR l=1 TO n
270 LET v(l)=1
280 NEXT l
290 LET m=n
300 FOR l=n TO 1 STEP -1
310 LET l1=m-a(l)
320 LET p(l)=v(l1)
330 FOR j=l1 TO m-1
340 LET v(j)=v(j+1)
350 NEXT j
360 LET m=m-1
370 NEXT l
380 IF k1=0 THEN PRINT
390 PRINT " (";
400 FOR l=1 TO n
410 PRINT p(l);" ";
420 NEXT l
430 PRINT ")";
440 LET k1=1-k1
450 RETURN

```

(1 2 3 4)	(1 2 4 3)
(1 3 4 2)	(2 3 4 1)
(1 3 2 4)	(1 4 2 3)
(1 4 3 2)	(2 4 3 1)
(2 3 1 4)	(2 4 1 3)
(3 4 1 2)	(3 4 2 1)
(2 1 3 4)	(2 1 4 3)
(3 1 4 2)	(3 2 4 1)
(3 1 2 4)	(4 1 2 3)
(4 1 3 2)	(4 2 3 1)
(3 2 1 4)	(4 2 1 3)
(4 3 1 2)	(4 3 2 1)

8.6.8 Se folosește problema 8.6.13.

```

algoritmul 8.6.8 este:
  citește n
  execută
  [
    citește m
    ieșim-cînd  $m < n$ 
     $j \leftarrow n$ 
    cheamă P
     $p \leftarrow p_1$ 
     $j \leftarrow n - m$ 
    cheamă P
     $p \leftarrow p/p_1$ 
    pentru  $i=1, m$  execută
    [
       $x_i \leftarrow i$ 
      sfîrșit-pentru
      cît-timp  $j \leq p$  execută
      [
        execută
        cheamă Q
        ieșim-cînd  $i \neq 0$  or  $j > p$ 
        dacă  $j \leq p$  atunci
        [
          cît-timp  $x_i = n$  execută
          [
            execută
             $x_i \leftarrow 1$ 
             $i \leftarrow i - 1$ 
            ieșim-cînd  $i \neq 0$  or  $x_i \neq n$ 
          ]
          sfîrșit-cît-timp
          dacă  $x_i \neq n$  atunci  $x_i \leftarrow x_i + 1$ 
          sfîrșit-dacă
        ]
      ]
      sfîrșit-dacă
    ]
    sfîrșit-cît-timp
  ]
  stop

procedura P este:
   $p_1 \leftarrow 1$ 
  pentru  $i=2, j$  execută
  [
     $p_1 \leftarrow p_1 \cdot i$ 
  ]
  sfîrșit-pentru
  sfîrșit-procedură

procedura Q este:
   $sw \leftarrow 0$ 
  pentru  $i=1, m-1$  execută
  [
    pentru  $l=i+1, m$  execută
    [
      dacă  $x_l = x_i$  atunci  $sw \leftarrow 1$ 
      [
         $l \leftarrow m$ 
         $i \leftarrow m-1$ 
      ]
    ]
    sfîrșit-dacă
  ]
  sfîrșit-pentru
  sfîrșit-pentru
  dacă  $sw \neq 1$  atunci
  [
     $j \leftarrow j+1$ 
    pentru  $i=1, m$  execută
    [
      scrie  $x_i$ 
      sfîrșit-pentru
       $i \leftarrow m$ 
    ]
  ]
  sfîrșit-dacă
  sfîrșit-procedură

```



```

60 INPUT "Introduceti n,m(n>m) ";n,m
70 DIM x(m)
80 LET j=n: GO SUB 380: LET p=p1
90 LET j=n-m: GO SUB 380: LET p=p/p1
100 PRINT " " ARANJAMENTE DE ";n;" CITE ";m;" SINT"
110 LET kk=0
120 FOR i=1 TO m
130 LET x(i)=i
140 NEXT i
150 LET j=1
160 IF j>p THEN GO TO 360
170 LET sw=0
180 FOR i=1 TO m-1
190 FOR l=i+1 TO m
200 IF x(l)=x(i) THEN LET sw=1: LET l=m: LET i=m-1
210 NEXT l
220 NEXT i
230 IF sw=1 THEN GO TO 330
240 LET j=j+1
250 PRINT " (" ;
260 FOR i=1 TO m
270 PRINT x(i);" ";
280 NEXT i
290 PRINT ")";
300 IF kk=1 THEN PRINT
310 LET kk=1-kk
320 LET i=m
330 IF i=0 THEN GO TO 160
340 IF x(i)=n THEN LET x(i)=1: LET i=i-1: GO TO 330
350 LET x(i)=x(i)+1: GO TO 160
360 PAUSE 0: CLS : STOP
370
380 LET p1=1
390 FOR i=2 TO j
400 LET p1=p1*i
410 NEXT i
420 RETURN

```

ARANJAMENTE DE 4 CITE 3 SINT

```

( 1 2 3 ) ( 1 2 4 )
( 1 3 2 ) ( 1 3 4 )
( 1 4 2 ) ( 1 4 3 )
( 2 1 3 ) ( 2 1 4 )
( 2 3 1 ) ( 2 3 4 )
( 2 4 1 ) ( 2 4 3 )
( 3 1 2 ) ( 3 1 4 )
( 3 2 1 ) ( 3 2 4 )
( 3 4 1 ) ( 3 4 2 )
( 4 1 2 ) ( 4 1 3 )
( 4 2 1 ) ( 4 2 3 )
( 4 3 1 ) ( 4 3 2 )

```

8.6.9 Problema se reduce la generarea tuturor submulțimilor cu maximum m elemente ale unei mulțimi date și la însumarea elementelor pentru fiecare dintre aceste submulțimi.

```

10 LET Z1=6: LET Z2=22: GO SUB 610
20 PRINT "      * SUMA ELEMENTELOR  *"
30 PRINT "      * SUBMULTIMILOR CU  *"
40 PRINT "      * MAXIMUM M. ELEMENTE  *"
50 PRINT "      * ALE UNEI MULTIMI  *"
60 PRINT "      *          DATE          *": GO SUB 610
70 PRINT /
80 INPUT "Introduceti card multimei si nr M ";N,M
90 DIM A(N): DIM B(N): DIM X(50)
100 LET NN=4
110 PRINT "      MULTIMEA ESTE:"/"      {";
120 FOR I=1 TO N
130   INPUT A(I): LET VV= A(I): LET B(I)=0: GO SUB 540
140 NEXT I: PRINT " }"
150 PRINT /"      URMATOARELE NUMERE SE SCRUI"
160 PRINT "      CA SUMA A MAXIMUM ";M;" ELEMENTE"
170 PRINT "      ALE MULTIMII DATE:"/"
180 LET B(N)=1
190 LET L=0: LET K=1
200 IF K>M THEN GO TO 390
210 LET S=0
220 FOR I=1 TO N
230   LET S=S+A(I)*B(I)
240 NEXT I
250 LET SW=0
260 FOR I=1 TO L
270   IF S=X(I) THEN LET SW=1
280 NEXT I
290 IF SW=1 THEN GO TO 390
300 LET L=L+1: LET X(L)=S
310 PRINT "      "": LET NN=0: LET VV=S: GO SUB 540: LET KK=0
320 PRINT "=";
330 FOR I=1 TO N: IF B(I)=0 THEN GO TO 380
340   IF B(I)=0 THEN GO TO 380
350   LET KK=KK+1: IF KK=1 THEN GO TO 370
360   IF A(I)>=0 THEN PRINT "+";
370   PRINT A(I);
380 NEXT I: PRINT
390 LET I=N
400 IF B(I)=0 THEN GO TO 420
410 LET B(I)=0: LET K=K-1: LET I=I-1: GO TO 400
420 LET B(I)=1: LET K=K+1
430 IF K<>N THEN GO TO 200
440 LET P=1
450 FOR I=2 TO L
460   LET SW=0
470   FOR J=1 TO I-1
480     IF X(I)=X(J) THEN LET SW=1: LET J=I
490   NEXT J
500   IF SW=0 THEN LET P=P+1: LET X(P)=X(I)
510 NEXT I
520 PAUSE 0: CLS : STOP
540 FOR U=1 TO 4-LEN (STR$ (VV))
550   PRINT " ";
560 NEXT U
570 PRINT VV;: LET NN=NN+4
580 IF NN>26 THEN PRINT /"      "": LET NN=4
590 RETURN
610 FOR U=1 TO Z1: PRINT " "": NEXT U
620 FOR U=1 TO Z2: PRINT " *": NEXT U: PRINT : RETURN

```

```

*****
*   SUMA ELEMENTELOR   *
*   SUBMULTIMILOR CU  *
*   MAXIMUM M ELEMENTE *
*   ALE UNEI MULTIMI  *
*       DATE           *
*****

```

MULTIMEA ESTE:
{ -3 6 -2 0 1 }

URMATOARELE NUMERE SE SCRIB
CA SUMA A MAXIMUM 3 ELEMENTE
ALE MULTIMII DATE:

```

1=1
0=0
-2=-2
-1=-2+1
6=6
7=6+1
4=6-2
5=6-2+1
-3=-3
-5=-3-2
-4=-3-2+1
3=-3+6

```

8.6.10 Se pleacă de la permutarea $X=(1,2,\dots,n)$. Fie o permutare $X=(x_1, x_2, \dots, x_n)$. Pentru determinarea permutării ce urmează în ordine lexicografică se caută cel mai mare indice k pentru care $x_k < x_{k+1}$. Apoi se determină cel mai mic dintre elementele de indice mai mare decât k . Fie x_i acel minim. Schimbând între ele elementele x_k și x_i se obține șirul: $X'=(x_1, \dots, x_{k-1}, x_i, x_{k+1}, \dots, x_{i-1}, x_k, x_{i+1}, \dots, x_n)$ în care ultimele $n-k$ elemente apar în ordine descrescătoare. Deci următoarea permutare se obține inversind ordinea ultimelor $n-k$ elemente ale șirului X .

procedura P este:

1 ← 0

pentru $i=n, 1, -1$ execută

dacă $x_k < x_i$ atunci 1 ← i

$y \leftarrow x_k$

$x_k \leftarrow x_i$

$x_i \leftarrow y$

$i \leftarrow 1$

sfârșit-dacă

sfârșit-pentru

pentru $j=1, [(n-k)/2]$ execută $y \leftarrow x_{k+j}$

$x_{k+j} \leftarrow x_{n+1-j}$

$x_{n+1-j} \leftarrow y$

sfârșit-pentru

sfârșit-procedură

algoritmul 8.6.10. este:

```
citește n
pentru i=1,n execută
     $x_i \leftarrow i$ 
sfîrșit-pentru
k  $\leftarrow$  0
pentru i=n-1,1,-1 execută
    dacă  $x_i < x_{i+1}$  atunci k  $\leftarrow$  i
        i  $\leftarrow$  1
    sfîrșit-dacă
sfîrșit-pentru
cît-timp k $\neq$ 0 execută
    cheamă P
    k  $\leftarrow$  0
    pentru i=n-1,1,-1 execută
        dacă  $x_i < x_{i+1}$  atunci k  $\leftarrow$  i
            i  $\leftarrow$  1
        sfîrșit-dacă
    sfîrșit-pentru
sfîrșit-cît-timp
stop
```

```
10 PRINT "//
20 PRINT " *****"
30 PRINT " * GENERAREA PERMUTARILOR *"
40 PRINT " * IN ORDINE LEXICOGRAFICA *"
50 PRINT " *****"
60 PRINT "//
70 INPUT "n=";n
80 DIM x(n)
90 LET k1=0
100 FOR i=1 TO n
110 LET x(i)=i
120 NEXT i
130 PRINT " (";
140 FOR i=1 TO n
150 PRINT x(i);" ";
160 NEXT i
170 PRINT ")";
180 IF k1=1 THEN PRINT
190 LET k1=1-k1
200 LET k=0
210 FOR i=n-1 TO 1 STEP -1
220 IF x(i) > x(i+1) THEN GO TO 240
230 LET k=i: LET i=1
240 NEXT i
250 IF k=0 THEN GO TO 370
260 LET l=0
270 FOR i=n TO 1 STEP -1
280 IF x(k) > x(i) THEN GO TO 310
290 LET l=i: LET y=x(k): LET x(k)=x(i)
300 LET x(i)=y: LET i=1
310 NEXT i
320 FOR j=1 TO INT ((n-k)/2)
330 LET y=x(k+j): LET x(k+j)=x(n+1-j)
340 LET x(n+1-j)=y
350 NEXT j
360 GO TO 130
370 STOP
```



```

*****
* GENERAREA PERMUTARILOR *
* IN ORDINE LEXICOGRAFICA *
*****

```

```

(1 2 3 4 )   (1 2 4 3 )
(1 3 2 4 )   (1 3 4 2 )
(1 4 2 3 )   (1 4 3 2 )
(2 1 3 4 )   (2 1 4 3 )
(2 3 1 4 )   (2 3 4 1 )
(2 4 1 3 )   (2 4 3 1 )
(3 1 2 4 )   (3 1 4 2 )
(3 2 1 4 )   (3 2 4 1 )
(3 4 1 2 )   (3 4 2 1 )
(4 1 2 3 )   (4 1 3 2 )
(4 2 1 3 )   (4 2 3 1 )
(4 3 1 2 )   (4 3 2 1 )

```

```

(1 2 3 )   (1 3 2 )
(2 1 3 )   (2 3 1 )
(3 1 2 )   (3 2 1 )

```

8.6.11 Se pleacă de la şirul $X=(1,2,\dots,k)$ şi se obţine submulţimea următoare în ordine lexicografică în felul următor: se foloseşte un vector caracteristic Y cu n elemente (iniţial primele k elemente ale sale sînt egale cu 1 şi următoarele nule) vector ce va indica în fiecare moment care elemente ale mulţimii apar şi care nu în şirul X . Pentru un şir dat X se generează următoarea submulţime ordonată astfel: se determină cel mai mare indice i cu proprietatea că x_i poate fi mărit, ceea ce înseamnă că cel puţin una dintre valorile x_{i+1} , x_{i+2}, \dots, x_n nu apare în şirul X . În procesul căutării indicelui i cu proprietatea cerută, ori de cîte ori se ajunge la un element, valoarea sa trebuie făcută disponibilă (adică anulat elementul corespunzător al şirului Y), deoarece toate elementele de indice mai mare sau egal cu i vor fi înlocuite în şirul X . Odată determinat indicele i , elementele x_i, x_{i+1}, \dots, x_n vor fi înlocuite cu cele mai mici numere disponibile, în ordine crescătoare. Dacă nu există indicele i cu proprietatea menţionată mai sus înseamnă că au fost generate toate submulţimile ordonate cu k elemente.

```

10 PRINT " *****"
20 PRINT " * ARANJAMENTE GENERATE *"
30 PRINT " * FOLOSIND SIRURI ORDONATE *"
40 PRINT " * LEXICOGRAFIC *"
50 PRINT " *****"
60 INPUT "Introduceti n,m (m<n) ";n,m
70 DIM x(m): DIM y(n)
80 PRINT "" ARANJAMENTE DE ";n;" LUATE CITE ";m
90 PRINT "" ( "?: LET kk=0
100 FOR i=1 TO m
110 LET x(i)=i: LET y(i)=1
120 PRINT x(i);" ";
130 NEXT i

```

```

140 PRINT ")";: LET kk=1-kk
150 IF kk=0 THEN PRINT
160 FOR i=m+1 TO n
170 LET y(i)=0
180 NEXT i
190 LET sw=0
200 FOR i=m TO 1 STEP -1
210 LET j=x(i): LET y(j)=0
220 IF x(i)=n THEN GO TO 440
230 FOR j=x(i)+1 TO n
240 IF y(j)<>0 THEN GO TO 370
250 LET x(i)=j: LET y(j)=1: LET k=0: LET l=i+1
260 IF l>m THEN GO TO 300
270 LET k=k+1
280 IF y(k)<>0 THEN GO TO 270
290 LET x(l)=k: LET y(k)=1: LET l=l+1: GO TO 260
300 PRINT " ( ";
310 FOR q=1 TO m
320 PRINT x(q);" ";
330 NEXT q
340 PRINT ")";: LET kk=1-kk
350 IF kk=0 THEN PRINT
360 LET sw=1: GO TO 200
370 NEXT j
380 LET s=0
390 FOR j=1 TO m
400 IF x(j)=n-j+1 THEN GO TO 420
410 LET s=1: LET j=m
420 NEXT j
430 IF s=0 THEN GO TO 460
440 NEXT i
450 IF sw=1 THEN GO TO 190
460 PAUSE 0: CLS : STOP

```

```

*****
* ARANJAMENTE GENERATE *
* FOLOSIND SIRURI ORDONATE *
* LEXICOGRAFIC *
*****

```

ARANJAMENTE DE 4 LUATE CITE 3

```

( 1 2 3 ) ( 1 2 4 )
( 1 3 2 ) ( 1 3 4 )
( 1 4 2 ) ( 1 4 3 )
( 2 1 3 ) ( 2 1 4 )
( 2 3 1 ) ( 2 3 4 )
( 2 4 1 ) ( 2 4 3 )
( 3 1 2 ) ( 3 1 4 )
( 3 2 1 ) ( 3 2 4 )
( 3 4 1 ) ( 3 4 2 )
( 4 1 2 ) ( 4 1 3 )
( 4 2 1 ) ( 4 2 3 )
( 4 3 1 ) ( 4 3 2 )

```

8.6.12 Se pleacă de la șirul $X=(1,2,\dots,m)$. Pentru o submulțime dată ca șir X se generează următoarea în ordine lexicografică în felul următor: se caută cel mai mare indice j cu proprietatea că $x_j < n-m+j$. Atunci următoarea submulțime este dată de șirul $(x_1, x_2, \dots, x_{j-1}, x_j+1, x_j+2, \dots, x_j+n-j+1)$.

Dacă un asemenea indice nu există, înseamnă că au fost generate toate combinările de n luate câte m .

algoritmul 8.6.12. este:

citește n, m

pentru $i=1, m$ execută

$x_i \leftarrow i$

sfârșit-pentru

pentru $i=1, m$ execută

scrie x_i

sfârșit-pentru

$k \leftarrow 0$

pentru $j=m, 1, -1$ execută

dacă $x_j \neq n-m+j$ atunci $k \leftarrow j$

$x_k \leftarrow x_k+1$

$j \leftarrow 1$

sfârșit-dacă

sfârșit-pentru

cît-timp $k \neq 0$ execută

pentru $j=k+1, m$ execută

$x_j \leftarrow x_{j-1}+1$

sfârșit-pentru

pentru $i=1, m$ execută

scrie x_i

sfârșit-pentru

$k \leftarrow 0$

pentru $j=m, 1, -1$ execută

dacă $x_j \neq n-m+j$ atunci $k \leftarrow j$

$x_k \leftarrow x_k+1$

$j \leftarrow 1$

sfârșit-dacă

sfârșit-pentru

sfârșit-cît-timp

stop

```

10 PRINT " *****"
20 PRINT " * COMBINARI GENERATE *"
30 PRINT " * FOLOSIND SIRURI ORDONATE *"
40 PRINT " * LEXICOGRAFIC *"
50 PRINT " *****"
60 INPUT "Introduceti n,m (m<n) ";n,m
70 DIM x(m)
80 FOR i=1 TO m
90 LET x(i)=i
100 NEXT i
110 PRINT "" COMBINARI DE ";n;" LUATE CITE ";m
120 PRINT "": LET kk=0
130 PRINT " (" ;
140 FOR i=1 TO m
150 PRINT x(i);" ";
160 NEXT i

```

```

170 PRINT ")";: LET kk=1-kk
180 IF kk=0 THEN PRINT
190 LET k=0
200 FOR j=m TO 1 STEP -1
210 IF x(j)=n-m+j THEN GO TO 230
220 LET k=j: LET x(k)=x(k)+1: LET j=1
230 NEXT j
240 IF k=0 THEN GO TO 290
250 FOR j=k+1 TO m
260 LET x(j)=x(j-1)+1
270 NEXT j
280 GO TO 130
290 PAUSE 0: CLS : STOP

```

```

*****
*   COMBINARI GENERATE   *
* FOLOSIND SIRURI ORDONATE *
*   LEXICOGRAFIC       *
*****

```

COMBINARI DE 5 LUATE CITE 3

```

( 1 2 3 ) ( 1 2 4 )
( 1 2 5 ) ( 1 3 4 )
( 1 3 5 ) ( 1 4 5 )
( 2 3 4 ) ( 2 3 5 )
( 2 4 5 ) ( 3 4 5 )

```

8.6.13 Deosebirea față de problema 8.6.5 constă în faptul că nu se scriu decît submulțimile pentru care vectorul caracteristic corespunzător are exact m elemente egale cu 1 (deci și șirul X are m elemente).

```

algoritmul 8.6.13. este:
citește n,m
pentru i=1,n-m execută
  xi ← 0
sfîrșit-pentru
pentru i=n-m+1,n execută
  xi ← 1
sfîrșit-pentru
k ← m
execută
  dacă k=m
    atunci cheamă SCR
  sfîrșit-dacă
  l1 ← 0
  pentru i=n,1,-1 execută
    cheamă P
  sfîrșit-pentru
ieșim-cînd l1=0
stop

```

```

procedura SCR este:
pentru i=1,m execută
  dacă xi=1
    atunci scrie i
  sfîrșit-dacă
sfîrșit-pentru
sfîrșit-procedură

```

```

procedura P este:
dacă xi=1 atunci xi ← 0
                                k ← k-1
altfel xi ← 1
                                k ← k+1
                                l1 ← 1
                                i ← 1
sfîrșit-dacă
sfîrșit-procedură

```



```

10 PRINT " *****"
20 PRINT " * COMBINARI GENERATE *"
30 PRINT " * FOLOSIND VECTORI *"
40 PRINT " * CARACTERISTICI *"
50 PRINT " *****"
60 INPUT "Introduceti n,m (m<n) ";n,m
70 DIM x(n)
80 FOR i=1 TO n-m
90 LET x(i)=0
100 NEXT i
110 LET k=m
120 FOR i=n-m+1 TO n
130 LET x(i)=1
140 NEXT i
150 PRINT "" COMBINARI DE ";n;" LUATE CITE ";m
160 PRINT "": LET kk=0
170 IF k<>m THEN GO TO 240
180 PRINT " (" ;
190 FOR i=1 TO n
200 IF x(i)=1 THEN PRINT i;" ";
210 NEXT i
220 PRINT ")";: LET kk=1-kk
230 IF kk=0 THEN PRINT
240 LET l1=0
250 FOR i=n TO 1 STEP -1
260 IF x(i)=1 THEN LET x(i)=0: LET k=k-1: GO TO 280
270 LET x(i)=1: LET k=k+1: LET l1=1: LET i=1
280 NEXT i
290 IF l1=1 THEN GO TO 170
300 PAUSE 0: CLS : STOP

```

```

*****
* COMBINARI GENERATE *
* FOLOSIND VECTORI *
* CARACTERISTICI *
*****

```

COMBINARI DE 5 LUATE CITE 3

```

( 3 4 5 ) ( 2 4 5 )
( 2 3 5 ) ( 2 3 4 )
( 1 4 5 ) ( 1 3 5 )
( 1 3 4 ) ( 1 2 5 )
( 1 2 4 ) ( 1 2 3 )

```

8.6.14

```

10 DIM A(11)
20 PRINT " *****"
30 PRINT " * NUMERE DE TELEFON *"
40 PRINT " * DE M CIFRE FORMATE *"
50 PRINT " * CU M CIFRE DATE *"
60 PRINT " *****"
70 INPUT "Introduceti M si N (M<N<10)"; M,N
80 IF M<1 OR N>9 THEN GO TO 70

```

```

90 LET KK=0
100 PRINT "/ "      N=";N;" M=";M
110 PRINT "/ "      NUMERELE DE TELEFON SINT:"/"
120 PRINT " "      "": LET K=0
130 LET A(M+1)=0: LET A(M)=1
140 FOR I=1 TO M-1
150   LET A(I)=0
160 NEXT I
170 FOR I=M TO 1 STEP -1
180   PRINT A(I);" ";
190 NEXT I
200 PRINT " "": LET K=K+1
210 LET KK=KK+2*M+1
220 IF KK>32-2*M THEN PRINT " "": LET KK=0
230 LET I=1
240 IF A(I)=N THEN LET A(I)=0: LET I=I+1: GO TO 240
250 LET A(I)=A(I)+1
260 IF A(M+1)=1 THEN GO TO 280
270 GO TO 170
280 PAUSE 0: CLS : STOP

```

```

*****
* NUMERE DE TELEFON *
* DE M CIFRE FORMATE *
* CU M CIFRE DATE *
*****

```

N=4 M=3

NUMERELE DE TELEFON SINT:

```

1 0 0 1 0 1 1 0 2 1 0 3
1 0 4 1 1 0 1 1 1 1 1 2
1 1 3 1 1 4 1 2 0 1 2 1
1 2 2 1 2 3 1 2 4 1 3 0
1 3 1 1 3 2 1 3 3 1 3 4
1 4 0 1 4 1 1 4 2 1 4 3
1 4 4 2 0 0 2 0 1 2 0 2
2 0 3 2 0 4 2 1 0 2 1 1
2 1 2 2 1 3 2 1 4 2 2 0
2 2 1 2 2 2 2 2 3 2 2 4
2 3 0 2 3 1 2 3 2 2 3 3
2 3 4 2 4 0 2 4 1 2 4 2
2 4 3 2 4 4 3 0 0 3 0 1
3 0 2 3 0 3 3 0 4 3 1 0
3 1 1 3 1 2 3 1 3 3 1 4
3 2 0 3 2 1 3 2 2 3 2 3
3 2 4 3 3 0 3 3 1 3 3 2
3 3 3 3 3 4 3 4 0 3 4 1
3 4 2 3 4 3 3 4 4 4 0 0
4 0 1 4 0 2 4 0 3 4 0 4
4 1 0 4 1 1 4 1 2 4 1 3
4 1 4 4 2 0 4 2 1 4 2 2
4 2 3 4 2 4 4 3 0 4 3 1
4 3 2 4 3 3 4 3 4 4 4 0
4 4 1 4 4 2 4 4 3 4 4 4

```

8.6.15 Se calculează într-o primă etapă suma U a tuturor elementelor șirului X și se inițializează toate elementele y_i cu valoarea $1+(U-x_i)$; apoi se calculează produsul tuturor elementelor șirului X de indice diferit de i , produs ce se adaugă la elementul y_i . Pentru fiecare indice k între valorile 2 și $n-2$ se generează toate submulțimile de k elemente ale mulțimii indicilor $\{1,2,\dots,n\}$ cu una dintre metodele cunoscute, submulțimi ce se vor păstra într-o matrice A cu C_n^k linii și k coloane. Pentru fiecare indice i se parcurge matricea A și pentru liniile ce nu conțin indicele i se calculează produsul elementelor șirului X cu indicii aflați în linia respectivă, produse ce se adaugă elementului y_i .

```

10 CLS : LET Z1=6: LET Z2=19: GO SUB 740
20 PRINT " * GENERARE DE SIR *"
30 PRINT " * DIN ALT SIR *"
40 GO SUB 740
50 INPUT "Introduceti dim sirului initial"; N
60 DIM A(30,N)
70 DIM X(N): DIM Y(N): DIM Z(N)
80 PRINT "" " SIRUL INITIAL ESTE:""" X="( ";
90 FOR I=1 TO N
100 INPUT X(I): LET VV=X(I): GO SUB 710
110 IF I/6=INT (I/6) THEN PRINT "" ";
120 NEXT I: PRINT " )"
130 LET U=0
140 FOR I=1 TO N
150 LET U=U+X(I)
160 NEXT I
170 FOR I=1 TO N
180 LET Y(I)=1+U-X(I): LET P2=1
190 FOR J=1 TO N
200 IF I<>J THEN LET P2=P2*X(J)
210 NEXT J
220 LET Y(I)=Y(I)+P2
230 NEXT I
240 FOR K=2 TO N-2
250 GO SUB 510
260 FOR I=1 TO N
270 LET S1=0
280 FOR J=1 TO L
290 LET SW=0
300 FOR P=1 TO K
310 IF A(J,P)<>I THEN GO TO 330
320 LET SW=1: LET P=K
330 NEXT P
340 IF SW=1 THEN GO TO 400
350 LET P1=1
360 FOR P=1 TO K
370 LET L1=A(J,P): LET P1=P1*X(L1)
380 NEXT P
390 LET S1=S1+P1
400 NEXT J
410 LET Y(I)=Y(I)+S1
420 NEXT I
430 NEXT K

```

```

440 PRINT " " SIRUL GENERAT ESTE: "" Y=( ";
450 FOR I=1 TO N
460 LET VV=Y(I): GO SUB 710
470 IF I/6=INT (I/6) THEN PRINT " " ";
480 NEXT I: PRINT " )"
490 PAUSE 0: CLS : STOP
500
510 LET L=0
520 FOR I=1 TO K
530 LET Z(I)=I
540 NEXT I
550 LET L=L+1
560 FOR I=1 TO K
570 LET A(L,I)=Z(I)
580 NEXT I
590 LET KK=0
600 FOR J=K TO 1 STEP -1
610 IF Z(J)=N-K+J THEN GO TO 630
620 LET KK=J: LET Z(KK)=Z(KK)+1: LET J=1
630 NEXT J
640 IF KK=0 THEN GO TO 690
650 FOR J=KK+1 TO K
660 LET Z(J)=Z(J-1)+1
670 NEXT J
680 GO TO 550
690 RETURN
700
710 FOR W=1 TO 4-LEN STR$ (VV): PRINT " ";; NEXT I
720 PRINT VV;; RETURN
730
740 FOR V=1 TO Z1: PRINT " ";; NEXT V
750 FOR V=1 TO Z2: PRINT "*";
760 NEXT V: PRINT : RETURN

```

```

*****
* GENERARE DE SIR *
* DIN ALT SIR *
*****

```

SIRUL INITIAL ESTE:

X=(1 2 3 4 5)

SIRUL GENERAT ESTE:

Y=(360 240 180 144 120)

SIRUL INITIAL ESTE:

X=(1 5 2)

SIRUL GENERAT ESTE:

Y=(18 6 12)

8.7.1 O relație binară între elementele unei mulțimi finite se definește folosind o matrice booleană pătrată cu atâtea linii, respectiv coloane, câte elemente are mulțimea considerată, elementul de indici i și j luind valoarea 1 în cazul în care elementul i este în relație cu elementul j . Proprietățile studiate sînt: reflexivitatea, simetria, antisimetria și tranzitivitatea. Relația binară este reflexivă dacă toate elementele diagonalei principale sînt egale cu 1, ceea ce se poate verifica însumînd elementele situate pe diagonala principală, sumă ce trebuie să fie egală cu dimensiunea matricei. Proprietatea de simetrie se traduce în faptul că matricea de definiție este simetrică. Relația binară este antisimetrică dacă, pentru orice pereche de indici diferiți i și j , suma elementelor de indici i și j , respectiv j și i este diferită de doi. Pentru a verifica tranzitivitatea se generează toate tripletele de indici i, j, k ; dacă suma elementelor de indici (i, j) și (j, k) este egală cu doi și elementul de indici (i, k) este nul cel puțin pentru unul dintre tripletele generate, înseamnă că relația nu este tranzitivă.

```

10 PRINT "": LET z1=7: LET z2=18: GO SUB 960
20 PRINT " * RELATII BINARE *"
30 GO SUB 960
40 PRINT
50 INPUT "Introduceti numarul de elemente al multimii ";n
60 PRINT : PRINT " Multimea are ";n;" elemente"
70 DIM a(n,n)
80 PRINT " Matricea ce defineste rela-"" tia este:"
90 FOR i=1 TO n
100 PRINT " ";
110 FOR j=1 TO n
120 INPUT a(i,j)
130 IF a(i,j)<>0 AND a(i,j)<>1 THEN GO TO 120
140 PRINT a(i,j);" ";
150 NEXT j
160 PRINT
170 NEXT i
180 PRINT "
190 PRINT #0;"Studiem reflexivitatea": PAUSE 100
200 INPUT " ": GO SUB 400
210 IF sr<>1 THEN GO TO 230
220 PRINT " Relatia este reflexiva"
230 PRINT #0;"Studiem simetria": PAUSE 100: INPUT " "
240 GO SUB 660
250 IF s<>1 THEN GO TO 270
260 PRINT " Relatia este simetrica"
270 PRINT #0;" Studiem tranzitivitatea": PAUSE 100
280 INPUT " ": GO SUB 490
290 IF s1<>1 THEN GO TO 310
300 PRINT " Relatia este tranzitiva"
310 PRINT #0;" Studiem antisimetria": PAUSE 100: INPUT " "
320 GO SUB 830
330 IF s2=0 THEN GO TO 350
340 PRINT " Relatia este antisimetrica"
350 IF (sr+s+s1=3) THEN PRINT "" Relatia este de echivalenta"
360 IF (sr+s1+s2=3) THEN PRINT "" Relatia este de ordine"
370 PAUSE 0: CLS : STOP
380
390 REM studiul proprietatii de reflexivitate
400 LET sr=1: LET suma=0

```

```

410 FOR i=1 TO n
420   LET suma=suma+a(i,i)
430   IF a(i,i)=1 THEN GO TO 470
440   PRINT "   Relatia nu este reflexiva"/" deoarece ";i
450   PRINT " nu este in re-"" latie cu el insusi"
460   LET sr=0: LET i=n
470 NEXT i
501 RETURN
490
500 REM Studiem tranzitivitatea
510 LET s1=1
520 FOR i=1 TO n
530   FOR j=1 TO n
540     FOR k=1 TO n
550       IF (a(i,j)+a(j,k)<>2) OR (a(i,k)=1) THEN GO TO 620
560       PRINT ' "   Relatia nu este tranzitiva"
570       PRINT "   deoarece ";i;" este in relatie"
580       PRINT "   cu ";j;" care este in relatie"
590       PRINT "   cu ";k;" insa ";i;" nu este in re-"
600       PRINT "   latie cu ";k
610       LET s1=0: LET k=n: LET j=n: LET i=n
620     NEXT k
630   NEXT j
640 NEXT i
650 RETURN
660
670 REM Studiem simetria
680 LET s=1
690 FOR i=1 TO n-1
700   FOR j=i+1 TO n
710     IF a(i,j)=a(j,i) THEN GO TO 800
720     IF (a(i,j)=1) THEN LET ii=i: LET jj=j: GO TO 740
730     LET ii=j: LET jj=i
740     PRINT ' "   Relatia nu este simetrica"
750     PRINT "   deoarece ";ii;" este in relatie ";jj
760     PRINT "   insa ";jj;" nu este in relatie"
770     PRINT "   cu ";ii
780     LET i=n: LET j=n
790     LET s=0: LET i=n-1: LET j=n
800   NEXT j
810 NEXT i
820 RETURN
830
840 REM Studiem antisimetria
850 LET s2=1
860 FOR i=1 TO n
870   FOR j=i+1 TO n
880     IF a(i,j)+a(j,i)<>2 THEN GO TO 930
890     PRINT ' "   Relatia nu este antisimetri-"
900     PRINT "   ca deoarece ";i;" este in rela-"
910     PRINT "   tie cu ";j;" si reciproc"
920     LET s2=0: LET i=n: LET j=n
930   NEXT j
940 NEXT i
950 RETURN
960
970 FOR w=1 TO z1: PRINT " ";: NEXT w
980 FOR w=1 TO z2: PRINT "*";: NEXT w
990 PRINT : RETURN

```

* RELATII BINARE *

Multimea are 4 elemente

Matricea ce defineste relatia este:

0 1 0 1
1 1 0 1
0 0 1 0
1 1 0 1

Relatia nu este reflexiva
deoarece 1 nu este in relatie
cu el insusi
Relatia este simetrica

Relatia nu este tranzitiva
deoarece 1 este in relatie
cu 2 care este in relatie
cu 1 insa 1 nu este in relatie
cu 1

Relatia nu este antisimetri-
ca deoarece 1 este in rela-
tie cu 2 si reciproc

Multimea are 3 elemente

Matricea ce defineste rela-
tia este:

1 0 1
0 1 0
1 0 1

Relatia este reflexiva
Relatia este simetrica
Relatia este tranzitiva

Relatia nu este antisimetri-
ca deoarece 1 este in rela-
tie cu 3 si reciproc

Relatia este de echivalenta

8.7.2 O operație binară definită pe elementele unei mulțimi finite se definește printr-o matrice pătrată A (de dimensiune egală cu cardinalul mulțimii), elementul de indici (i, j) fiind egal cu rezultatul compunerii elementelor i și j . Proprietățile studiate sînt: comutativitatea, asociativitatea, existența eventualului element neutru, existența (dacă este cazul) a elementelor simetrizabile cu precizarea simetricului, respectiv a simetricelor acestora. Studiul comutativității se reduce la studiul simetriei matricei de definiție. Pentru a studia asociativitatea se generează toate tripletele de indici pentru a verifica egalitatea: $i \cdot (j \cdot k) = (i \cdot j) \cdot k$ care este echivalentă cu egalitatea $a_{il} = a_{mk}$ unde $l = a_{jk}$ și $m = a_{ij}$. Pentru a găsi eventualul element neutru se folosește o matrice pătrată suplimentară de același ordin n ale cărei elemente se inițializează cu 0. În această matrice se depun în linia i și în coloana j pentru care $i \cdot j = j \cdot i = i$. Dacă în cel puțin o linie nu s-a depus nici un element, operația nu admite element neutru. Dacă în toate liniile există elemente nenule se caută dacă există unul comun tuturor liniilor. Acesta, dacă există, este unic și reprezintă elementul neutru. Pentru a găsi simetricul, sau simetricile elementelor, în cazul în care operația admite element neutru, se formează câte un șir pentru fiecare indice i cu acei indici j pentru care $i \cdot j = j \cdot i =$ indicele elementului neutru. Dacă șirul este vid, elementul respectiv este nesimetrizabil. În caz contrar șirul conține simetricul, respectiv simetricile sale.

```

10 PRINT " ";
20 FOR i=1 TO 28: PRINT "*";: NEXT i
30 PRINT : PRINT " * STRUCTURI ALGEBRICE-GRUP *"
40 PRINT " ";: FOR i=1 TO 28: PRINT "*";: NEXT i
50 PAUSE 0: CLS
60 INPUT "Introduceti numarul de elemente ale multimii "; n
70 PRINT " " " Multimea are ";n;" elemente"
80 DIM a(n,n)
90 DIM x(n,n)
100 DIM d(n)
110 PRINT " " " Matricea este:" : PRINT
120 FOR i=1 TO n
130 PRINT " ";
140 FOR j=1 TO n
150 INPUT a(i,j)
160 IF a(i,j)<1 AND a(i,j)>n THEN GO TO 150
170 PRINT a(i,j);" ";
180 NEXT j
190 PRINT
200 NEXT i
210 PRINT #0;"Studiem comutativitatea": PAUSE 100: INPUT " "
220 GO SUB 610
230 IF s<>0 THEN GO TO 250
240 PRINT " " " Operatia este comutativa"
250 PRINT #0;"Studiem asociativitatea": PAUSE 100: INPUT " "
260 GO SUB 440
270 IF s1<>0 THEN GO TO 290
280 PRINT " " " Operatia este asociativa"
290 PRINT #0;"Studiem existenta elementului neutru"
300 PAUSE 100: INPUT " "
310 GO SUB 740
320 IF k=0 OR l=0 THEN GO TO 430
330 PRINT #0;"Cautam elementele simetrizabile"
340 PAUSE 100: INPUT " "

```



```

350 GO SUB 1100
360 IF s2<>n THEN GO TO 1300
370 IF s1<>0 THEN GO TO 1300
380 IF s=0 THEN GO TO 410
390 PRINT " " " Operatia data determina o"
400 PRINT " " " structura de grup necomutativ ": GO TO 430
410 PRINT " " " Operatia data determina o"
420 PRINT " " " structura de grup abelian"
430 PAUSE 0: CLS : GO TO 60
440
450 REM Studiem asociativitatea
460 LET s1=0
470 FOR i=1 TO n
480   FOR j=1 TO n
490     FOR k=1 TO n
500       LET l=a(i,j): LET m=a(j,k)
510       IF a(l,k)=a(i,m) THEN GO TO 570
520       PRINT " " " Operatia nu este asociativa"
530       PRINT " " " deoarece (";
540       PRINT i;"*";j;"")*";k;"=";a(l,k);" iar "
550       PRINT " " " ";i;"*(";j;"*";k;"")=";a(i,m)
560       LET s1=1: LET k=n: LET j=n: LET i=n
570     NEXT k
580   NEXT j
590 NEXT i
600 RETURN
610
620 REM Studiem comutativitatea
630 LET s=0
640 FOR i=1 TO n-1
650   FOR j=i+1 TO n
660     IF a(i,j)=a(j,i) THEN GO TO 720
670     PRINT " " " Operatia este necomutativa"
680     PRINT " " " deoarece "; i;"*";j;"=";a(i,j);
690     PRINT " " " iar ";j;"*";i;"=";a(j,i)
700     LET s=1: LET i=n-1: LET j=n
710   NEXT j
720 NEXT i
730 RETURN
740
750 REM Studiem existenta elementului neutru
760 FOR i=1 TO n
770   LET l=0
780   FOR j=1 TO n
790     IF a(i,j)<>i THEN GO TO 820
800     IF a(j,i)<>i THEN GO TO 820
810     LET l=l+1: LET x(i,l)=j
820   NEXT j
830   IF l<>0 THEN GO TO 860
840   PRINT " " " Operatia nu admite element" " " neutru"
850   LET i=n
860 NEXT i
870 IF l=0 THEN GO TO 1090
880 LET k=0
890 FOR j=1 TO n
900   IF x(1,j)=0 THEN GO TO 1040
910   LET l=0
920   FOR l=2 TO n
930     FOR m=1 TO n
940       IF x(1,j)<>x(1,m) THEN GO TO 970

```

```

950         LET l1=l1+1
960         LET m=n
970     NEXT m
980     IF l1=l-1 THEN GO TO 1000
990         LET l=n
1000    NEXT l
1010    IF l1<>n-1 THEN GO TO 1040
1020        LET c=x(1,j)
1030        LET j=n: LET i=n: LET k=1
1040    NEXT j
1050    IF k=0 THEN GO TO 1080
1060    PRINT ' " Operatia admite elementul"/"      neutru ";c
1070    GO TO 1090
1080    PRINT " Operatia nu admite element"/"      neutru"
1090    RETURN
1100
1110    REM Cautam simetricile
1120    LET s2=0
1130    FOR i=1 TO n
1140        LET k=0
1150        FOR j=1 TO n
1160            IF a(i,j)<>c THEN GO TO 1180
1170            LET k=k+1: LET d(k)=j
1180        NEXT j
1190        IF k=0 THEN GO TO 1270
1200        IF k=1 THEN GO TO 1250
1210        PRINT " ";i;" admite simetricile";
1220        FOR m=1 TO k
1230            PRINT " ";d(m);
1240        NEXT m
1250        PRINT ' " ";i;" admite simetricul ";d(1)
1260        LET s2=s2+1: GO TO 1280
1270        PRINT " ";i;" este un element nesimetrizabil"
1280    NEXT i
1290    RETURN

```

```

*****
* STRUCTURI ALGEBRICE-GRUP *
*****

```

Multimea are 4 elemente

Matricea este:

```

1 1 1 1
2 4 3 1
3 4 2 1
4 1 4 1

```

Operatia este necomutativa
deoarece $1*2=1$ iar $2*1=2$

Operatia nu este asociativa
deoarece $(2*1)*2=4$ iar
 $2*(1*2)=2$

Operatia nu admite element
neutru

Multimea are 5 elemente

Matricea este:

```
1 2 3 4 5
2 1 2 3 4
3 2 1 2 3
4 3 2 1 2
5 4 3 2 1
```

Operatia este comutativa

Operatia nu este asociativa
deoarece $(2*2)*3=3$ iar
 $2*(2*3)=1$

Operatia admite elementul
neutru 1

```
1 admite simetricul 1
2 admite simetricul 2
3 admite simetricul 3
4 admite simetricul 4
5 admite simetricul 5
```

Multimea are 5 elemente

Matricea este:

```
1 2 3 4 5
2 3 1 3 1
3 1 5 1 2
4 3 1 2 3
5 1 2 3 3
```

Operatia este comutativa

Operatia nu este asociativa
deoarece $(2*2)*3=5$ iar
 $2*(2*3)=2$

Operatia admite elementul
neutru 1

```
1 admite simetricul 1
2 admite simetricul 3 5
2 admite simetricul 3
3 admite simetricul 2 4
3 admite simetricul 2

4 admite simetricul 3

5 admite simetricul 2
```

Multimea are 4 elemente

Matricea este:

1 2 3 4
2 1 4 3
3 4 1 2
4 3 2 1

Operatia este comutativa

Operatia este asociativa

**Operatia admite elementul
neutru 1**

1 admite simetricul 1

2 admite simetricul 2

3 admite simetricul 3

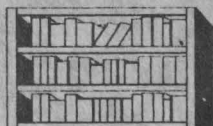
4 admite simetricul 4

**Operatia data determina o
structura de grup abelian**

In curs de aparitie:

Colectia:

*Biblioteca pentru
elevi si studenti*



- *Limbajul C*
(la cerere cu floppy-disk)
- *Microsoft*
GW-BASIC
(la cerere cu floppy-disk)
- *Culegere de probleme in Turbo-Pascal*
(la cerere cu floppy-disk)
- *dBASE III plus*
- *Sisteme de operare interactive*
- *Culegere de programe in C*
(la cerere cu floppy-disk)

*Initiere in
informatica*

ABC
Info

- *Primii pasi in programarea
calculatoarelor*

*Pentru
specialisti*

HIGH ^{info}

- *Limbajul PROLOG*
- *Metode si algoritmi de clasi ficare
si recunoasterea formelor*

CUPRINS

Prefața.....	3
Notă.....	4
I - Prezentare teoretică.....	5
1.1 Noțiuni generale.....	5
1.2 Reprezentarea algoritmilor.....	7
1.2.1 Schema logică.....	7
1.2.2 Pseudocodul.....	11
1.2.3 Limbajul de programare.....	12
1.3 Limbajul BASIC.....	13
1.3.1 Elementele limbajului.....	13
II - Enunțuri.....	19
1 Structuri liniare și alternative.....	19
2 Sume, produse, contoare.....	22
3 Structuri repetitive de tip WHILE-DO și DO-UNTIL.....	22
4 Structuri repetitive de tip FOR.....	25
5 Șiruri.....	27
6 Matrice.....	30
7 Șiruri de caractere.....	33
8 Probleme diverse.....	36
9 Probleme propuse.....	50
III Indicații și soluții.....	53

Com. 403 3000 exemplare

IMPRIMERIA "ARDEALUL" CLUJ



Lei 300,--

