

INSTITUTUL CENTRAL DE FIZICA
INSTITUTUL DE FIZICA SI INGINERIE NUCLEARA
Bucuresti, B.O.B.5206, ROMANIA

IFIN NC 510 585
MC-8-1980 V Aprilie

Programe FORTRAN și ASSEMBLER

N. MOLDOVAN

Abstract: This publication is a collection of programs written in FORTRAN and ASSEMBLER programming languages used in DOS-IBM. The problems solved are of different sorts: linear programming, integration, matrix calculus, computation of absorbed doses in teletherapy, data sets (files) on magnetic tapes and disks, completion of DOS operating system etc. For reasons of space no details are given on the numerical methods or supplements and devices developed in order to achieve superior programs as to computation time and accuracy of result, although these might have been of use. All the programs in the collection have been checked up on an IBM 370/135 computer.

IA LOC DE PREFATA

Prezentăm o colecție de programe în limbajele (IBM) FORTRAN și ASSEMBLER pentru tipuri diverse de probleme. Nu includem aici metodele de calcul folosite - cu o excepție - din lipsă de spațiu, deși ar fi util. Mai ales că pentru calcul numeric efectiv, pentru a obține rezultate optime cu eforturi minime, între metodele teoretice din analiza numerică ce par apropiate, există diferențe esențiale din acest punct de vedere. Mai mult, "aranjarea" unui algoritm pentru programare este un lucru de mare importanță. Iar aceste chestiuni nu prea apar în cărți. Toate programele prezentate în colecție au fost rulate pe calculatorul IBM 370/135.

PROGRAMARE LINIARA

Am realizat un program pentru programarea liniară în ipoteza că în memoria principală a calculatorului încap toate informațiile necesare (programe și date de intrare).

Cu acest program se pot rezolva probleme generale (cu inecuații, cu legături incompatibile, cu legături liniar dependente, cu variabile de orice semn etc.).

Am folosit metoda variabilelor pilot (leading variables) al cărei algoritm a fost îmbunătățit, cu scopul de a mări precizia deciziilor luate în procesul de calcul și a rezultatelor obținute. În acest fel, lucrul în precizie simplă (6 cifre hexazecimale) asigură rezultate bune pentru probleme de dimensiuni destul de mari și nu prea bine condiționate.

Programul realizat, împreună cu MAINPGM-ul, cere cca. 12800 bytes de memorie, iar o problemă în M variabile și N legă-

turi (fără cele de tip X.GE.C.) necesită $4MN+12N+32M+12NLIM+8INNS-8EGNS+10$ bytes pentru date și arie de lucru. Am notat : $NLIM-NR$. Variabilelor cu limite inferioare nenule, $INNS-NR$. Inecuațiilor liniar independente, $EGNS-NR$. Egalităților liniar independente în sistemul de legături. $INNS$ nu conține legăturile de tip X.GE.C. La acest spațiu se mai adaugă spațiul cerut de rutinele sistemului de operare (DOS, OS, etc.) care completează programele de mai sus în execuție.

REZOLVAREA PROBLEMELOR DE PROGRAMARE LINIARA

O problemă de programare liniară se enunță astfel:

Să se găsească valorile nenegative ale variabilelor x_j , $j = 1, \dots, n$ pentru care funcția liniară numită și funcție economică, beneficiu sau cost

$$E = e_0 + \sum_{j=1}^n e_j x_j \quad (1)$$

are valoarea optimă (maximă sau minimă) și verifică sistemul liniar (restricții sau legături ale problemei)

$$a_{i_0} + \sum_{j=1}^n a_{ij} x_j = 0, \quad i = 1, \dots, m \quad (2)$$

Un set de valori ale variabilelor x_j care satisface aceste condiții îl vom numi soluție a problemei sau, mai scurt, soluție.

O soluție nenegativă a sistemului (2) o numim plan. Dacă cel mult m componente ale unui plan sînt nenule, îl vom numi plan fundamental. Interpretînd planul ca punct în spațiul euclidian n dimensional, se arată că mulțimea tuturor planelor este fie mulțime vidă, fie un poliedru în acest spațiu. Virfurile poliedrului corespund planelor fundamentale, celelalte puncte ale

poliedrului corespund planelor cu cel puțin $m+1$ componente nenule.

Dacă problema are soluție finită, se arată că există un plan fundamental ca soluție și că toate soluțiile problemei se obțin din combinațiile convexe ale tuturor planelor fundamentale soluție. Deci problema este complet rezolvată prin găsirea tuturor planelor fundamentale-soluții.

Investigarea tuturor vîrfurilor poliedrului, în număr de C_n^m , ar fi o metodă de rezolvare mult prea laborioasă.

Pentru a reduce din numărul de vîrfuri investigate, în metoda simplex și variantele acesteia, se procedează în modul următor. Se află un plan fundamental, nu t de plecare sau inițial, pentru care funcția economică are o anumită valoare. Dacă aceasta nu este optimă, se va căuta un plan fundamental vecin cu acesta (vîrfuri vecine pe poliedrul soluțiilor nenegative ale sistemului de legături) care să dea o valoare mai bună. Această metodă are, însă, două neajunsuri. În primul rînd găsirea planului fundamental inițial este o treabă neplăcută care se rezolvă mărind forțat dimensiunea problemei (prin introducerea de noi variabile), iar criteriul de îmbunătățire a valorii funcției economice în iterația următoare este destul de slab. O metodă mai bună decît acestea este metoda variabilelor pilot (leading variables). În această metodă nu se lucrează cu plane fundamentale ci cu soluții oarecare (și componente negative) ale sistemului (2), fapt ce permite o foarte ușoară plecare în procesul iterativ de îmbunătățire a valorii funcției economice. În al doilea rînd, criteriul de îmbunătățire este mai tare, el alegînd pe aceea dintre soluțiile vecine (tot cu cel mult m componente nenule) care îmbunătățește cel mai mult funcția econo-

mică. De aici rezultă, în general, o mai rapidă convergență a algoritmului. În plus, așa cum reiese din cele ce urmează, prin această metodă problema se rezolvă mai complet, programul realizat cuprinzând aproape toate situațiile teoretice și practice posibile.

Metoda variabilelor pilot

În programul realizat se tratează problema sub forma cea mai generală, deci sistemul de legături putând fi format din ecuații sau inecuații de orice tip :

$$\alpha_{i_0} + \sum_{j=1}^n \alpha_{ij} x_j \geq, \leq, = 0, i = 1, \dots, n \quad (2')$$

Soluția inițială a lui (2') se obține în modul următor: Se iau în considerație relațiile , pe rând, care pot fi de tip egalitate sau inegalitate. Dacă este vorba de o inegalitate, aceasta va fi transformată în egalitate prin introducerea unei variabile nenegative față de care se va rezolva respectiva relație. Dacă relația este egalitate se va rezolva față de variabila ce are cel mai mare (în valoare absolută) coeficient. În acest mod, se ajunge a avea sistemul de legături sub forma:

$$x_i = \alpha_{i_0} + \sum_{j=k+1}^L \alpha_{ij} x_j, i = 1, \dots, k \quad (k \leq m, L \geq n)$$

și funcția economică

$$E = \beta_0 + \sum_{j=k+1}^L \beta_j x_j$$

În acest mod se acceptă spre rezolvare și probleme cu legături dependente, incompatibile etc. programul realizat dînd răspunsul corespunzător.

Problema inițială, adusă la această formă, este transformată într-o problemă apropiată de aceasta, prin introducerea a încă unei variabile, x_0 , și încă unei ecuații (numită ecuație pilot) :

$$\begin{cases} x_i = \alpha_{i_0} x_0 + \sum_{j=k+1}^L \alpha_{ij} x_j, & i = 1, \dots, k \\ 1 = x_0 + \omega (x_{k+1} + \dots + x_L) \\ E = \beta_0 x_0 + \sum_{j=k+1}^L \beta_j x_j \end{cases} \quad (3)$$

ω este un parametru nenegativ mic (ori de câte ori se pune problema valorii lui ω , aceasta este pozitivă și oricât de mică).

Se arată că problema inițială și cea transformată sînt echivalente din punct de vedere al existenței și coincidenței soluțiilor, cu mențiunea că problema transformată are o variabilă în plus x_0 . Această variabilă poate avea în soluția problemei transformate două valori 1 și 0. Cînd $x_0 = 1$, problema inițială are soluție mărginită (valoarea optimă finită), iar cînd $x_0 = 0$, problema inițială are soluție nemărginită (valoarea optimă nemărginită pe mulțimea soluțiilor nenegative ale sistemului de legături). Pe parcursul procesului de rezolvare se păstrează forma ultimă a problemei, exceptînd ecuația pilot care are forma generală :

$$1 = \sum z_j x_j$$

Variabilele din dreapta egalităților se numesc variabile pilot, iar cele din stînga baza variabilelor.

Soluția de plecare dată de $x_0 = 1$, $x_i = \alpha_{i_0}$, $i = 1, \dots, k$ și $x_i = 0$ pentru $i = k+1, \dots, L$ duce la soluția vecină (cele două diferă printr-o componentă cu valoarea nenulă) care are proprietatea că are valori nenegative pentru variabi-

lele pilot și este cea mai bună cu această proprietate (în cazul căutării unui minim pentru E) :

$$x_0=0, x_1=0, i=k+1, \dots, N-1, N+1, \dots, L, x_N=1/q_N, \text{ unde}$$

N se determină din relația

$$\beta_N/q_N = \min_{q_j > 0} \{ \beta_j/q_j \}$$

cînd există cel puțin un $q_j > 0$, sau

$x_0 = 1, x_1 = 0, i = k+1, \dots, L$, în caz conncrar. Variabila x_N (sau x_0) se numește variabilă (pilot) privilegiată. Celelalte variabile (baza) au valorile rezultate din înlocuirea valorilor variabilelor libere în egalitățile (3). Acestea pot fi și negative. Dacă toate ar fi pozitive, soluția obținută pentru sistemul de legături ar fi cea optimă, deci soluția problemei. Pentru a găsi o soluție mai bună, în cazul cînd nu s-a ajuns la optim, se va lua ca variabilă pilot una dintre variabilele bazei care a avut valoarea cea mai negativă și se va duce în bază variabila pilot privilegiată. Faptul revine la a rezolva una dintre legăturile porblemei față de variabila privilegiată și eliminarea acesteia din celelalte legături și funcția economică. Noua formă a problemei este analogă precedentei și i se aplică același procedeu. Se pot ivi două situații: se ajunge la soluția problemei transformate fără ca x_0 să fi fost privilegiată sau să fi fost pe acest rol (eventual în ultima iterație). În primul caz problema pusă are soluție nemărginită (funcție economică nemărginită pe mulțimea planelor). În al doilea caz, x_0 este supusă tratamentului obișnuit, dar înainte de a face următoarea iterație se va lua $x_0 = 1, s = 0$ și ecuația din care a fost dus în bază x_0 va lua locul ecuației pilot. Din acest

moment cele două probleme coincid identic.

Procesul iterativ descris este convergent și duce la soluție. Dificultățile ce apar pot fi două : alegerea variabilei din bază care să devină variabilă pilot nu este unică sau alegerea variabilei privilegiate nu este unică. Prima situație nu aduce neplăceri mari, orice alegere ducând la soluție (într-un număr mai mare sau mai mic de iterații). În programul realizat s-a ales prima variabilă din bază (în ordinea investigației) care are cea mai mică valoare (negativă). Faptul că alegerea variabilei privilegiate nu ar fi unică aduce oarecare neplăceri: posibilitatea unui ciclu infinit de baze investigate. Acest neajuns este înlăturat, când apare, prin artificiiul polinomului în ϵ (parametru mic) :

$$P(\epsilon) = \epsilon x_0 + \epsilon^2 x_1 + \dots + \epsilon^{L+1} x_L$$

Se alege ca variabilă privilegiate aceea pentru care valoarea polinomului este optimă.

Algoritmul folosit dă posibilitatea de a găsi toate soluțiile problemei (când există mai multe). În modul următor: se rezolvă ecuația pilot față de ultima variabilă privilegiate și aceasta se elimină din funcția economică. Forma obținută a funcției are toți coeficienții de același semn. Toate variabilele care au coeficienții nuli în această formă a funcției economice, împreună cu ultima bază obținută sînt toate variabilele ce pot avea valori nenule în soluțiile problemei. Sistemul liniar în aceste variabile, obținut astfel, conține toate soluțiile problemei și reciproc. Programul realizat dă ca rezultat și acest sistem. Funcția economică obținută arată variabilele de care depinde efectiv valoarea ei, orice valoare nenulă a acestor variabile strică valoarea optimului.

Introducerea datelor în calculator este descrisă în instrucțiunile comentariu din programul principal.

Deoarece sistemul de legături sub formă rezolvată față de o bază este, în general, de dimensiuni mult inferioare formei inițiale, precum și din alte motive, am realizat programul algoritmului în două etape, cărora le corespund subrutinele LPROG și OPTIM. Operațiile care se repetă cel mai des în aceste subrutine au fost realizate în ASSEMBLER.

Prin modul de realizare a programului s-a urmărit reducerea activității de "paging" care se face la sistemele care lucrează în memorie virtuală (VS).

Descrierea MAINPGM-ului

Alocarea dinamică a memoriei calculatorului necesită compilarea unui program care să rezerve spațiul necesar problemei de rezolvat. Programul acesta este minim posibil și se reduce în esență, la compunerea unei instrucțiuni DIMENSION (conform cu indicațiile din comentariile MAINPGM-ului). Cu o singură rulare a MAINPGM-ului se pot rezolva oricâte probleme, luând dimensiuni acoperitoare.

Descrierea subrutinei LPROG

În subrutină se citesc datele problemei, se tipăresc pentru a putea fi verificate și se aduce problema la forma (3). O realizare importantă a subrutinei constă în modificarea problemei pentru a o aduce la o formă în care mărimile coeficienților sistemului de legături să rămână cât mai stabile în cursul calculului, în sensul că aceștia să fie cât mai puțin dispersați pe axa numerelor reale. Faptul va permite evaluarea mai exactă a diferi-

telor situații în care erorile de rotunjire datorate lucrului cu numere de lungime fixă ar putea duce la concluzii false. În cazul unor probleme foarte instabile din acest punct de vedere se recomandă lucrul în precizie dublă. Pentru asta programul suferă mici modificări.

În instrucțiunile FORMAT ale subrutinei se pot vedea o serie de mesaje-răspuns diferitelor probleme posibile teoretic și practic.

Subrutina LPROG face apel la subrutinele STDNME, ELIMS și OPTIM.

Descrierea subrutinei OPTIM

Subrutina optimizează valoarea funcției economice, prin procesul descris anterior, înlătură ciclurile infinite de baze, calculează o soluție și sistemul tuturor soluțiilor etc. O realizare prețioasă a subrutinei constă în modificarea problemei într-o problemă în care exponentul valorilor soluțiilor posibile să poată fi cât mai diferit (10 la puterea + sau - 50), fără ca eroarea relativă a acestora să sufere, așa cum se poate vedea pe exemplele rezolvate și prezentate la sfârșitul acestei lucrări.

Diferitele mesaje-rezultat se pot vedea în instrucțiunile FORMAT ale subrutinei. OPTIM face apel la subrutinele ELIM și STDNME.

Descrierea subrutinelor ELIMS, ELIM și STDNME

Subrutina ELIMS elimină din legături variabila de introdus în bază, la rezolvarea sistemului față de o bază.

Subrutina ELIM elimină din legături variabila pilot privilegiată.

Subrutina STDNME generează nume standard $X(n)$, dându-se n ca număr întreg. Funcționează pentru n mai mic decât 1000.

Toate aceste trei subrutine sînt scrise în ASSEMBLER.

Folosirea programului

Orice sistem de calcul care are compilatoare pentru limbajele FORTRAN IV și ASSEMBLER (fără macro-instrucțiuni), poate în principiu, rula programul de față, cu condiția ca problema să încapă în memoria calculatorului.

Modul de introducere a datelor inițiale este descris în instrucțiunile comentariu ale MAINPGM-ului, aici vom face unele precizări.

Legăturile pot fi egalități sau/și inegalități, cu termeni liberi în stînga pe locul întii, iar membrii dreپți nuli. Variabilele pot fi și negative, dar limitate inferior.

Pentru a reduce timpul de calcul și spațiul de memorie ocupat, se recomandă a trata legăturile de tipul X.GE.T. conform cu indicațiile de mai jos.

Indicatorii legăturilor definesc tipurile legăturilor (egalitate, inegalitate.GE. sau inegalitate .LE.), fără cele tratate conform punctului 8.

Formatul de prezentare a datelor de intrare (sistemul de legături și funcția economică) este la dispoziția noastră. Aceasta poate fi util cînd matricea de intrare are mulți coeficienți nuli. Același format se folosește la tipărirea lor în scopul verificării perforării pe cartele.

În numele variabilelor blankul se ia în considerare.

Dimensiunile vectorilor de rezervarea spațiului sînt acoperitoare, deci cînd nu se cunosc INNS și /sau ECNS, ori cînd

se rezolvă mai multe probleme într-o singură execuție a MAINPGM-ului se pot lua mai mari, cu independență între vectori.

În soluții apar numai variabilele cu valori nenule. Din sistemul soluțiilor se obțin toate soluțiile problemei adunând fiecărei soluții (componentelor ei) limitele inferioare ale variabilelor (și celor cu valoare zero în soluție).

Am realizat și alte variante de programe pentru problema de care ne ocupăm, dintre care mai importante sînt două. Una dă posibilitatea lucrului ca în cazul folosirii unui subprogram oarecare (se trimit datele din MAINPGM și se întoarce înapoi o soluție și valoarea optimului). Varianta cere 8920 bytes memorie principală. A doua variantă vizată mai sus face o verificare a soluției obținută și o îmbunătățire, cînd e posibilă, a soluției și cere 14430 bytes.

Subliniem forma optimizată a programelor realizate, atît ca spațiu de memorie ocupat, cît și ca timp de calcul, folosind în acest scop, o serie de artificii (lucrul cu aceeași mulțime de numere fie ca vector, fie ca matrice, în același subprogram, folosirea spațiilor de lucru pentru mărimi de orice tip, în același program, forma optimă a unor instrucțiuni din limbajul FORTRAN IV ținînd cont de unele caracteristici ale compilatorului DOS FORTRAN IV 360N-FO-479 3-8 și a structurii hardware-ului sistemelor de calcul IBM.

Mai precizăm grija programelor în ce privește plaja largă a soluțiilor posibile, fără a dăuna preciziei relative (cifre semnificative exacte) a acestora. Acest lucru s-a realizat prin transformări ale problemei în timpul calculului cu scopul de a face cît mai "egale" ponderile variabilelor în sistemul de legături, revenind la forma inițială doar la sfîrșitul calculelor.

Atragem atenția asupra fenomenului de pierdere de capacitate" (dispariția la adunări/scăderi a primelor cifre semnificative ale termenilor), care în algoritmele de rezolvarea problemelor de programare liniară este mai complex decât ar părea, el putând să apară datorită unor baze intermediare ale căror matrici ar fi slab condiționate cu toate că baza finală este bine condiționată. Fenomenul mai poate fi cauzat și de termenii liberi ai unor inegalități din sistemul de legături, sau mai bine zis a verificării prea "tare" de către soluție a respectivei inegalități. Varianta amintită mai sus de verificare și îmbunătățire a soluției înlătură fenomenul cauzat de unii termeni liberi. Acest lucru se realizează prin schimbarea legăturii în cauză cu alta echivalentă în care pierderea de capacitate este eliminată. În general, corectarea soluției este bine să fie făcută nu mărinđ lungimea de lucru (precizie dublă sau extinsă), ci plecând cu o soluție aproximativă cu baza optimă corect găsită, salvind în acest fel spațiu de memorie și timp de calcul. Tehnica aceasta reduce problema la rezolvarea unui sistem de ecuații liniare slab condiționat, în cazul cel mai nefavorabil.

Bibliografie

- /1/ COLECTIE DE PROGRAME, Ed. Acad. RSR, 1967, pag. 269 - 312.
- /2/ SAUL I. GASS, Linear programming. Methods and application, New York, 1958.
- /3/ S. VAJDA, The theory of games and linear programming, New York, 1956.
- /4/ M. SIMONNARD, Programmation lineaire, DUNOD, 1962.

***** MAINPGM *** *****

--- PREZENTAREA PROBLEMEI ---

UN SISTEM LINIAR CU NR LEGATURI (EGALITATI SAU SI INEGALITATI)
SI O FUNCTIE LINIARA (FC. ECONOMICA), IN NC-1 VARIABILE LIMITATE
INFERIOR (PRIN VALORI DE ORICE SEMN)
NC+NR. INEGALITATI < 1000. TERMENII LIBERI DIN LEGATURI IN MEM-
BRI STINGI PE LOCUL INTII, MEMBRI DREPTI NULI. FC. ECONOMICA CU
TERMEN LIBER PE LOCUL INTII. ORDINEA VARIABILELOR ACEASI IN
TOATE LEGATURILE SI IN FC. ECONOMICA).

--- REZULTATUL ---

SE OBTINE O SOLUTIE SI SISTEMUL SOLUTIILOR PT. CARE FC.
ECONOMICA ESTE OPTIMA. SE FAC CEL MULT 4*NR ITERATII.
IN SOLUTII POT APARE VAR. NEGATIVE X(NC), X(NC+1), ... PRO-
VENIND DIN TRANSFORMAREA (IN PROGRAM) A INEG. IN EG. (ORDI-
NEA DIN 2). IPOTEZA: TATE SOLUTIILE DIFERA DE 1.E-55
ABREVIATII IN MESAJE: COEF-COEFICIENT, LEG-LEGATURA, SOL-SOLUTIE,
VAR-VARIABILA, NEG-NEGATIV, FC-FUNCTIE, T-TERMEN, SIST-SISTEM ETC.

--- INTRODUCEREA DATELOR PE CARTELE ---

- 1) NC, NR - CU FORMAT(2,5)
- 2) OPERATORII LEGATURILOR - FORMAT(16,15). NR INTREGI: 0 PT.
'EGALITATE', -1 PT. INEGALITATE 'CEL MULT' (.LE.), 1 PT. INEGALITATE
'CEL PUTIN' (.GE.), CITE 16 PE O CARTELA (EXCEPTIND, EVENTUAL, ULTIMA).
- 3) NUMELE VARIABILELOR - FORMAT(8(A8,2X)). NUMELE (CEL MULT 8
CARACTERE TIPARIBILE) IN ORDINEA APARITIEI IN LEGA-
TURI, PE UN GRUP DE CARTELE. FOLOSIREA NUMELOR X(1), ..., X(NC-1),
CERE O SINGURA CARTELA CONTININD CUVINTUL 'STANDARD' - FORMAT(A8)
- 4) (XXEXX.X) SAU (XXEYY.X) - FORMAT(A9). X POATE FI CIFRA
ZECIMALA, BLANK SAU ABSENT. FORMATUL SUB CARE SE VOR CITI SI
TIPARI LEGATURILE SI FC. ECONOMICA (DATE INITIALE).
- 5) MAXIM SAU MINIM - FORMAT(A9). SE INDICA TIPUL OPTIMULUI.
TIP GRESIT INDICAT SE CONSIDERA MINIM.
- 6) FC. ECONOMICA - FORMAT INDICAT LA 4. NR COEFICIENTI
AI FUNCTIEI DE OPTIMIZAT. PE UN GRUP DE CARTELE.
- 7) NR LEGATURI - FORMAT INDICAT LA 4. SE INTRODUC FIECARE
LEGATURA (NC NUMERE) PE CITE UN GRUP DE CARTELE (ORDINEA DELA 2)
NU INCLUDETI AICI LEG. DE TIPUL X.CE.C (LIMITARI INFERIOARE)
8) LIMITARILE INFERIOARE - FORMAT(A8,217,7). NUMELE VAR. SI VALCA-
REA LIMITEI. DACA NU SINT LIMITE NENULE PUNCTUL POATE FI OMIS.
VARIABILA REPETATA ARE LIMITA=MAX(LIMITE)
PESTE NC-1 VAR. (DIFERITE) LIMITATE INFERIOARE ILF 2211

```

C 5) ENDLIMIT = FORMAT(A8).SE ARATA CA NU(MAI) SINT LIMITARI.
C 10) K SAU V, IN COLONANA I, PT. REZOLVARE SAU VERIFICARE A DATELOR
    REAL*8 D
C  SCHIMBAREA ORDINII IN DIMENSION ESTE INTERZISA(=ORICE EROARE)
    DIMENSION A(1),P(NC*NR),C(NR),E(NC+NLIM-1),F(NC+INNS),F(NC),
    1 G(NC),H(NC-EGNS),I(NC+INNS),J(NLIM),K(NC-EGNS)
C  INNS= NR. INEGALITATI NESUPERFLUE, EGNS=NR. EGAL. NESUPERFLUE
C  NLIM=NR. VAR. DISTINCTE INCLUSE LA PUNCTUL 8
C  DIMENSIUNILE LUI E SI LUI J CEL PUTIN 3 SI 1, RESPECTIV.
C  D EXECUTIE A MAINPGM-ULUI REZOLVA ORICITE PROBLEME LUI D
C  DIMENSIUNI ACOPERITOARE IN DIMENSION.
    1 READ(1,2,END=3) L,M
    2 FORMAT(2I5)
    CALL LPROG(A,B,C,D,E,F,G,H,I,J,K,L,M)
    GO TO 1
    3 STOP
    END

```

```

C  ****      SUBROUTINE LPROG      ****
    SUBROUTINE LPROG(V,A,IBAZA,ANUME,IJ,COST,ANORM,LIBERE,
    1 ECREZ,H,ECPII,NC,NR)
    REAL*8 STD,SCOP,ENDL,MAX,ANUME
    DIMENSION IJ(1),V(1),A(NC,1),COST(1),IBAZA(1),ANORM(1),
    1 ANUME(1),FCPII(1),LIBERE(1),ECREZ(1),H(1),FMT(3)
    EQUIVALENCE (R,INR),(CMAX,INC)
    DATA ENDL,STD,MAX,IREZ/'ENDLIMIT','STANDARD','MAXIM 1.1'
C  A(I,J)=V(J*NC+I+1-NC)
    WRITE(3,15) NC,NR
    READ(1,1) (IBAZA(I),I=1,NR)
    1 FORMAT(16I5)
    WRITE(3,1) (IBAZA(I),I=1,NR)
    IR=NC-1
    READ(1,2) (ANUME(I),I=1,9)
    2 FORMAT(8(A8,2X))
    IF(ANUME(1).EQ.STD) GO TO 3
    IF(IR.LE.8) GO TO 5
    READ(1,2) (ANUME(I),I=9,IR)
    GO TO 5
    3 DO 4 I=1,IR
    CALL STONME(SCOP,I)

```

```
4 ANUME(1)=SCOP
5 READ(1,8) FMT,SCOP
  WRITE(3,1E) (ANUME(I),I=1,IR)
  V(1)=1.
  IF(SCOP.NE.MAX) GO TO 6
  V(1)=-1.
6 READ(1,FMT) (CCST(I),I=1,NC)
  WRITE(3,FMT) (CCST(I),I=1,NC)
7 FORMAT(3A4/AF)
  WRITE(3,1E) SCOP
C  LEG SINT COLOANE IN MATR. A
  DO 9 J=1,NR
    READ(1,FMT) (A(J,I),J=1,NC)
    9 WRITE(3,FMT) (A(J,I),J=1,NC)
    DO 11 J=1,NC
      19 READ(1,10) ANUME(IR+J),H(J)
      10 FORMAT(A8,E17.7)
      NLIM=J-1
      IF(ANUME(IR+J).EQ.ENDL) GO TO 12
      IF(NLIM.EQ.0) GO TO 11
      DO 16 I=1,NLIM
        IF(ANUMF(IR+J).EQ.ANUMF(IR+I)) GO TO 17
      16 CONTINUE
      GO TO 11
      17 IF(H(I).LT.H(J)) H(I)=H(J)
      GO TO 19
    11 CONTINUE
    REWIND 1
C  FISIEKUL CU DSRN 1 ESTE CARD READER
  15 FORMAT(/////7X,'NR.CDEF=',I4/7X,'NR.LEG=',I4/)
  18 FORMAT(9A10)
  12 READ(1,8) NRVL
C  ELIMINAREA LIMITARILOR INFERIOARE.
  IF(NLIM.EQ.0) GO TO 21
  DO 35 J=1,NLIM
    SCOP=ANUMF(1NC+J-1)
    WRITE(3,2E) SCOP,H(J)
  26 FORMAT(5X,A8,'.6E.',F17.7)
  DO 31 N=2,NC
    IF(SCOP.EQ.ANUMF(N-1)) GO TO 33
```



```
31 CONTINUE
30 WRITE(3,32)
32 FORMAT(5X,'VARIABILA')
72 NC=0
73 RETURN
33 DO 34 I=1,NP
34 A(I,I)=A(I,I)+H(J)*A(N,I)
   COST(I)=COST(I)+F(J)*COST(N)
36 CONTINUE
C   NRVL=R (CA NR,INTREG) ?
21 IF(NRVL.NE.IREZ) GO TO 72
C   NORMAREA LEG. PRIN MAX(COEF.VAR,I)
   DO 28 I=1,NR
   INR=0
   DO 23 J=2,NC
   CMAX=ABS(A(J,I))
   IF(INR.LT.INC) INR=INC
23 CONTINUE
   IF(INR.EC.0) GO TO 28
   R=1./R
   DO 27 J=1,NC
27 A(J,I)=A(J,I)*R
28 CONTINUE
C   NORMAREA T.L. PT VAR. (IN LEG. SI COST)
   DO 44 I=1,NC
   INR=0
   DO 39 J=1,NR
   CMAX=ABS(A(I,J))
   IF(INR.LT.INC) INR=INC
39 CONTINUE
   IF(INR.EC.0) R=1.
   ANORM(I)=R
   R=1./R
   DO 42 J=1,NR
42 A(I,J)=A(I,J)*R
44 COST(I)=COST(I)*R
C   REZOLVAREA SISTEMULUI.
C   IBAZA SI LIBERE: JUDICIU K PT. VAR. DE LUCRU X(K)
C   IMPARTE LEG. CU MAX(COEF.VAR,I)
   IR=NR
```

```
      NUV=NC
      DO 53 I=1,NR
48  INC=0
      DO 49 J=2,NC
      R=ABS(A(J,I))
      IF(INC.GE.INR) GO TO 49
      INC=INR
      NRVL=J
49  CONTINUE
      N=I+NR-IR
      IN=IBAZA(N)
C. *** ECJATIE BUNA,NULA SAU INCOMPATIBILA ? ****
C. *** LEG.SLAB INDEPENDENTE(PIERDERE DE CAPACITATE) INCURCA.
      IF(CMAX.GT.1.E-5) GO TO 54
      IF(ABS(A(1,1)).LE.1.E-5) GO TO 51
      IF(IN.NE.0) GO TO 154
      WRITE(3,50) N
50  FORMAT(17X,'LEG',I4,' INCOMPATIBILA')
      GO TO 72
51  WRITE(3,52) N
52  FORMAT(17X,'LEG',I4,' SUPERFLUA')
      IR=IR-1
      IF(I.GT.IR) GO TO 64
      N=NC*IR+1
      NRVL=NC*I+2-NC
      DO 53 J=NRVL,N
53  V(J)=V(J+NC)
      GO TO 48
154  CMAX=1,
54  IF(IN) 55,57,56
55  CMAX=-CMAX
56  IRAZA(I)=NUV
      NUV=NUV+1
      GO TO 58
57  CMAX=-A(NRVL,I)
      IBAZA(I)=NRVL-1
58  CMAX=1./CMAX
      DO 59 J=1,NC
      R=A(J,I)*CMAX
```

```
A(J,1)=R
59 ECREZ(J)=R
   IF(IN.NE.0) GO TO 62
C   FLIMINAREA DIN LEGATURI
   CALL ELIMS(A,ECREZ,NRVL,IP,NC)
   DO 60 J=1,NC
60 A(J,1)=ECREZ(J)
C   FLIMINAREA DIN COST
   R=COST(NRVL)
   DO 61 J=1,NC
61 COST(J)=COST(J)+ECREZ(J)*R
62 IF(I.GE.IR) GO TO 64
63 CONTINUE
L   CONDENSAREA MATRICII SI COSTULUI
C   NRVL=NR.VAR.LIBERE(INCLUSIV XO)
64 NRVL=0
   DO 67 I=1,NC
   DO 65 J=1,IR
   IF(IBAZA(J).EQ.I-1) GO TO 67
65 CONTINUE
   NRVL=NRVL+1
   LIBERE(NRVL)=I-1
   ECPIL(NRVL)=1.
   COST(NRVL)=COST(I)
67 CONTINUE
   N=N+1
   DO 68 I=1,IR
   DO 68 J=1,NRVL
   N=N+1
68 V(N)=A(LIBERE(J)+1,I)
   CALL OPTIM(V,A,COST,ECPIL,ECREZ,ANORM,IJ,H,IBAZA,LIBERE,
1 ANUME,NC,NLIM,NRVL,IR)
   GO TO 73
   END
C ***** SUBROUTINE OPTIM *****
   SUBROUTINE OPTIM(V,A,COST,ECPIL,ECREZ,ANORM,VECHI,H,
1 IBAZA,LIBERE,ANUME,NC,NLIM,NRVL,IR)
   REAL*8 ANUME,NUME
   DIMENSION A(NRVL,1),COST(1),ECPIL(1),IBAZA(1),VECHI(1),
1 LIBERE(1),ECREZ(1),ANORM(1),ANUME(1),H(1),V(1)
```

```
FOUJVALENCE (R, INR), (RMIN, INRM)
C A(I, J)=V(J*NRVL+I+1-NRVL)
  AIND=V(I),
  V(I)=COST(I)*ANORM(I)
  COST(I)=0.
C NORMAREA PE VAR. (SI XC).
  VALCST=0.
  DO 107 I=1, NRVL
    INR=0
C P.C. LA TOTI COEF. SA NU NORMA.
  DO 104 J=1, NR
    RMIN=ABS(A(I, J))
    IF(INR.LT.INRM) INR=INRM
104 CONTINUE
  IF(INR.EQ.0) GO TO 106
  R=1./R
  DO 105 J=1, NR
105 A(I, J)=A(I, J)*R
  COST(I)=COST(I)*R
  IF(LIBERE(I).GE.NC) GO TO 106
  ANORM(LIBERE(I)+I)=ANORM(LIBERE(I)+I)/R
106 IF(VALCST.LT.ABS(COST(I))) VALCST=ABS(COST(I))
107 CONTINUE
C NORMAREA COSTULUI.
  IF(VALCST.NE.0.) AIND=AIND*VALCST
  DO 108 I=1, NRVL
108 COST(I)=COST(I)/AIND
C NORMAREA PE LEGATURI.
  DO 103 I=1, NR
    INR=0
C P.C. LA TOTI COEF. SA NU NORMA.
  DO 101 J=1, NRVL
    RMIN=ABS(A(J, I))
    IF(INR.LT.INRM) INR=INRM
101 CONTINUE
  IF(INR.EQ.0) GO TO 103
  R=1./R
  DO 102 J=1, NPVL
102 A(J, I)=A(J, I)*R
  IF(LIBAZA(I).GE.NC) GO TO 103
```

```
ANORM(I8AZA(I)+1)=ANORM(I8AZA(I)+1)*R
103 CONTINUE
K=1
NI=0
VALCST=1.E25
1 ECPIL(I)=1.E25
C SOLUTIE MAI, BUNA (VALCST=RMIN SI NOVP)
2 RMIN=1.E25
DO 3 I=1, NRVL
C *** CIND ESTE ZERO UN COEF. AL EC. PILET? ***
IF(ECPIL(I).LE.1.E-5) GO TO 3
R=COS(I)/ECPIL(I)
IF(RMIN.LE.R) GO TO 3
RMIN=R
NOVP=I
3 CONTINUE
4 FORMAT(/7X, 'LEG. SOL. NEG. ');
IF(RMIN.GE.1.E25) GO TO 14
C *** CIND ESTE COSTUL STATIONAR? ***
C RMIN ESTE FRECVENT ZERO
IF(ABS(VALCST/(RMIN+1.E-35)-1.).LE.1.E-5) GO TO 19
VALCST=RMIN
C DETERMINAREA VAR. DE SCHIMB(NES)
5 RMIN=0.
DO 6 I=1, NR
R=A(NOVP, I)
IF(R.GE.RMIN) GO TO 6
RMIN=R
NES=I
6 CONTINUE
C *** S-A AJUNS LA SCLUTIE ? ****
C *** SOL. (APROAPE) DEGENERATE (SUB NR VAR. >>0) INCURCA
IF(RMIN.GE.-1.E-5) GO TO 29
NI=NI+1
IF(NI.GT.4*NR) GO TO 15
7 DO 8 I=1, NRVL
8 ECREZ(I)=-A(I, NES)/RMIN
ECREZ(NOVP)=1./RMIN
C FLIMINAREA DIN IFCATURI
CALL ELIM(A, ECREZ, NOVP, NR, NRVL)
```

```
J=LIBERE(NCVP)
LIBERE(NCVP)=IBAZA(NES)
IBAZA(NES)=J
C   NI=-1 LA SOLUTIE PE XC.
   IF(NI.LT.0) GO TO 12
C   ELIMINAREA DIN EC. PILOT SI COST
   RMIN=COST(NCVP)
   R=ECPIIL(NCVP)
   DO 9 I=1, NRVL
     ECPIIL(I)=ECPIIL(I)+R*ECREZ(I)
9   COST(I)=COST(I)+RMIN*ECREZ(I)
   COST(NCVP)=RMIN*ECREZ(NCVP)
   ECPIIL(NCVP)=0*ECREZ(NCVP)
C   J=0 NUMAI CIND SE ELIMINA XO
   IF(J.EC.0) GO TO 11
C   ECUATIA NES LA LOCUL EI
   DO 10 I=1, N°VL
     10 A(I,NES)=ECREZ(I)
C   K=0 DUPA ELIMINAREA LUI XO.
   IF(K.EC.0) GO TO 2
   GO TO 1
C   ECUATIA NES DEVINE EC. PILOT
C   ECUATIA NES NU RAMINE IDENTIC NULA
  11 K=0
  12 DO 13 I=1, NRVL
    13 ECPIIL(I)=ECREZ(I)
     A(I,NES)=0.
     IF(NI) 50,50,2
C   ECUATIA PILOT CU TOTI COEF. NEGATIVI
  14 WRITE(3,4)
  15 WRITE(3,30) NI ,
  16 NC=0
  17 RETURN
C   COSTUL FSTE STATIONAR
C   COSTUL STATIONAR
  18 NT=NR+NRVL
   DO 19 I=1, NT
  19 VECHI(I)=0.
     VECHI(LIBERE(NCVP)+1)=R
     R=1./ECPIIL(NCVP)
```

```
DO 20 I=1, NR
20 VECHI(IEAZA(I)+1)=A(NQVP, I)*R
NES=NQVP
DO 26 I=1, NRVL
IF(ECPIL(I).LE.1.E-4) GO TO 29
IF(ABS(COST(I)/ECPIL(I)/(VALCST+1.E-35)-1.).GE.1.E-4) GO TO 28
IF(NES.EQ.I) GO TO 28
DO 21 J=1, NT
21 ECREZ(J)=0.
R=1./ECPIL(I)
ECPIL(LIBERE(I)+1)=R
DO 22 J=1, NR
22 ECREZ(IEAZA(J)+1)=A(I, J)*R
DO 25 J=1, NT
IF(ABS(ECREZ(J)).GT.1.E-4) GO TO 24
IF(ABS(VECHI(J)).LE.1.E-4) GO TO 25
23 IF(ECREZ(J).LE.VECHI(J)) GO TO 26
GO TO 26
24 IF(ABS(VECHI(J)).LE.1.E-4) GO TO 23
IF(ABS(VECHI(J)/ECREZ(J)-1.).LE.1.E-5) GO TO 25
GO TO 23
25 CONTINUE
26 NQVP=I
DO 27 J=1, NT
27 VECHI(J)=ECREZ(J)
28 CONTINUE
GO TO 5
C S-A GASIT SCLUTIA
29 WRITE(3,30) NI
NT=0
30 FORMAT(/7X, 'NR, ITERATII=', I4)
IF(K.EC.0) GO TO 44
C XD N-A FOST ELIMINATA.
IF(LIBERE(NQVP).EC.0) GO TO 46
C XD N-A FOST PRIVILEGIATA(OPTIM NEMARGINIT SAU NUL)
IF(ABS(COST(NQVP)).LT.1.E-5) GO TO 32
WRITE(3,31)
31 FORMAT(/7X, 'OPTIM NEMARGINIT, BAZA:')
GO TO 34
32 WRITE(3,33)
```

```
33 FORMAT(/7X,'OPTIM NUL.BAZA: ')
34 DO 39 I=1,NR
    J=IBAZA(I)
    NUME=ANUME(J)
    IF(J.GE.NC) CALL STORME(NUME,J)
39 WRITE(3,38) NUMB
38 FORMAT(7X,A8)
    J=LIBERE(NDVP)
    NUME=ANUME(J)
    IF(J.GE.NC) CALL STORME(NUME,J)
    WRITE(3,42) NUME
43 FORMAT(7X,A8,' LIBERA')
    GO TO 16
C   SOLUTIE CU X2 ELIMINATA.
44 VALCST=CCST(NDVP)/ECPIL(NDVP)
    DO 45 I=1,NRVL
45 COST(I)=COST(I)-VALCST*ECPIL(I)
C   COST(NDVP) REAL IN VALCST
    GO TO 50
C   SOLUTIE PE X0(ULTIMA VAR.PRIVILEG.)
46 NI=-1
    VALCST=CCST(1)
    DO 47 I=1,NR
    R=A(1,I)
    IF(RMIN.GE.R) GO TO 47
    RMIN=R
    NES=1
47 CONTINUE
    COST(1)=0.
C   COST(1) REAL IN VALCST
C *** SOLUTII(TOATE VAR.) MICI PCT CA FALSURI ***
    IF(RMIN.GE.1.E-5) GO TO 7
    NT=1
C   CALCULUL SOLUTIEI.TOATE VARIANTELE.
50 V(1)=VALCST*ATNE+ANDRM(1)+V(1)
    WRITE(3,51) V(1)
51 FORMAT(/7X,'OPTIMUL=' ,E15.7/)
    K=NC-1
    DO 52 J=1,K
52 ECRF7(J)=1.E-55
```



```
C.   CALCULUL LUI X(LIBERE(NOVPI))
      J=LIBERPE(NOVPI)
      IF((NC-J)*J.LE.0) GO TO 55
      ECREZ(J)=ANR4(I)/(ECPIL(NOVPI)/ANORM(J+1))
      IF(ECPIL(NOVPI).GT.1.E-5) ECREZ(J)=0.
L.   CALCULUL LUI X(IEAZA)
55  DO 58 J=1,NR
      T=IBAZA(J)
      IF((NC-I)*T.LE.0) GO TO 58
      RMIN=A(NOVPI,J)/ECPIL(NOVPI)
      IF(RMIN.LT.1.E-5) RMIN=0.
      ECREZ(I)=RMIN*ANORM(I)/ANORM(I+1)
58  CONTINUE
C.   IPOTEZA: TOATE COMPONENTELE SOL. CIFERA CU 1.E-55
      IF(NLIM.EQ.0) GO TO 59
      DO 61 J=1,NLIM
        DO 61 I=1,K
          IF(ANUME(I).EQ.ANUME(K+J)) ECREZ(I)=H(J)+ECREZ(I)
61  CONTINUE
59  DO 65 J=1,K
      R=ECREZ(J)
      IF(R.EQ.0.) GO TO 65
      IF(R.EQ.1.E-55) GO TO 65
      WRITE(3,64) ANUMF(J),R
64  FORMAT(9X,AR,1='1.E15.7)
65  CONTINUE
C.   CALCULUL SISTEMULUI TUTUROR SOLUTIILOR SI A FC.ECONOMICE.
L.   LIBERE, ECPIL SI A CONDENSATE, XO IN IEAZA( EXCEPTIE:
C.   SOL. PE XO SI SIST.SOL. OMOGENI)
L.   LEG. 1 IN VAP. X(IEAZA(I)) SI X(LIBERE(J)), J=1,...,NI
      NI=0
      WRITE(3,R9)
85  FORMAT(/7X,'FC.ECONOMICA:')
      DO 66 I=1,NRVL
C.   *** CINE O VARIABILA INFLUENTEAZA OPTIMUL? ***
      IF(COST(I).GT.1.E-5) GO TO 85
C.   VAR. DIN SISTEMUL SOLUTIILOR
      LIBERF(NI+1)=LIBERE(I)
L.   DETERMINAREA LUI LIBERE PESTE ECREZ
      LIBERE(NRVL+NI+1)=1
```

```
IF(LIBERE(I).NE.0) NI=NI+1
GO TO 66
C VAR. DIN FC. ECONOMICA
85 J=LIBERE(I)
R=1.
IF(J.GE.NC) GO TO 86
R=ANORM(J+1)
86 R=COST(I)*R*AIND
CALL STDNME(NUME,J)
WRITE(3,90) R,NUME
90 FORMAT(5X,E15.7,'*',A9)
66 CONTINUE
WRITE(3,94) V(I)
94 FORMAT(7X,'T.LIBER=',E15.7)
IF(NI+NT.GT.1) GO TO 68
WRITE(3,67)
67 FORMAT(/7X,'SOL UNICA')
GO TO 17
68 WRITE(3,69)
69 FORMAT(/9X,'SISTEMUL SOL(VAR NENULE):')
K=1
DO 76 I=1,NR
NOVP=IBAZA(I)
IF(NOVP.EQ.0) GO TO 76
DO 71 J=1,NI
R=1.
RMIN=1.
NES=LIBERE(J)
K=K+1
IF(NOVP.GE.NC) GO TO 70
R=ANORM(NOVP+1)
70 IF(NES.GE.NC) GO TO 71
RMIN=ANORM(NES+1)
71 V(K)=A(LIBERE(NRVL+J),I)/R*RMIN
CALL STDNME(NUME,NOVP)
J=K+1-NI
WRITE(3,75) NUME,(V(NES),NES=J,K)
75 FORMAT(4X,A9,'=',(6E15.7))
76 CONTINUE
IF(NT.EQ.1) GO TO 78
```

```
RMIN=1./ANORM(1)
DG 77 I=1,NI
R=ANORM(LIBERE(I)+1)
IF(LIBERE(I).LT.NC) GO TO 77
R=1.
77 ECPIL(I)=ECPIL(LIBERE(I+NRVL))*R*RMIN
WRITE(3,79) (ECPIL(J),J=1,NI)
78 WRITE(3,80)
79 FORMAT(10X,'1.=',(6E15.7))
80 FORMAT(7X,'COEF.VAR:')
DG 82 I=1,NI
J=LIBERE(I)
CALL STCNME(NUMB,J)
82 WRITE(3,38) NUMB
WRITE(3,83)
83 FORMAT(/9X,'TOATE SOL SE OBTIN ADUNIND SOL NENEGATIVE'/
1 9X,'ALE SIST(SI VAR LIPSA) LIMITELE INFERIOARE')
GO TO 17
END
```

```
ELIMS    CSECT    CALL ELIMS(A,ECREZ,NMAX,IP,NC)
          USING *,R15
          BC     15,12(C,15)
          DC     X'07C5D3C9'
          DC     X'D4E24040'
          STM    R2,R7,28(R13)
          L      R2,8(O,R1)
          L      R2,0(C,R2)
          BCT   R2,#+4
          SLL   R2,2    4NMAX-4:R2
          L      R5,16(O,R1)
          L      R5,0(C,R5)
          LR    R7,R5    NC:R7
          BCT   R5,#+4
          SLL   R5,2    4NC-4:R5
          L      R3,4(O,R1)  ADDR CF ECREZ(1):R3
          L      R6,12(O,R1) ADDR OF IR:R6
          M      R6,0(O,R6)  IR*NC:R7
          LR    R0,R6    0:R0
          BCT   R7,#+4
```

```
SLL R7,2 4*IR*NC-4:R7
L R1,0(0,R1) ADDR OF A(1,1):R1
LA R7,0(R1,R7) ADDR OF A(1,1):R7
LA R4,4 4:R4
NRECV LR R6,R0 0:R6
LE F0,0(R2,R1) A(NMAX,J)
ECV LE F2,0(R6,R3) ECREZ(I)
MER F2,F0 A(NMAX,J)*ECREZ(I)
AF F2,0(R6,R1) +A(I,J)
STE F2,0(R6,R1) A(I,J)
BXLE R6,R4,ECV
'BLF R1,R6,NRECV ADDR OF A(1,J+1):R1
LM R2,R7,28(R13)
MVI 12(13),255
BR R14
ELIM USECT CALL ELIM(A,ECREZ,NOVP,NR,NRVL)
USING *,R15
BC 15,12(C,15)
DC X'07C5D3C9'
DC X'D4404040'
STM R2,R7,28(R13)
L R2,8(0,R1)
L R2,0(0,R2)
BCT R2,*+4
SLL R2,2 4NOVP-4:R2
L R5,16(0,R1)
L R5,0(0,R5)
LR R7,R5 NRVL:R7
BCT R5,*+4
SLL R5,2 4NRVL-4:R5
L R3,4(0,R1) ADDR OF ECREZ(1):R3
L R6,12(0,R1) ADDR OF NR:R6
M R6,0(0,R6) NP*NRVL:R7
LR R0,R6 0:R0
BCT R7,*+4
SLL R7,2 4NR*NRVL-4:R7
L R1,0(0,R1) ADDR OF A(1,1):R1
LA R7,0(R1,R7) ADDR OF A(NRVL,NR):R7
LA R4,4 4:R4
NREC LR R6,R0 0:R6
```

```
LE    F0,C(R2,R1)    A(NCVP,J)
EC    LE    F2,0(R6,R3)    ECRESZ(I)
MER   F2,F0    A(NCVP,J)*ECRESZ(I)
AE    F2,0(R6,R1)    +A(I,J)
STC   F2,C(P6,R1)    A(I,J)
BXLF  R6,R4,EC
ME    F0,0(R2,R3)    A(NOVP,J)+ECRESZ(NOVP)
STF   F0,0(R2,R1)    A(NCVP,J)
BXLF  R1,R6,NREC    ADDR OF A(I,J+1):R1
LM    R2,R7,2E(R13)
MVI   12(13),255
BR    R14
STDNME CSECT    CALL STDNME(INAME,I)    I<1000
        USING *,R15
        BC    15,12(C,15)
        DC    X'07E2E3C4'
        DC    X'D5D4C540'
        ST    R7,2B(R13)
        L     R2,4(C,R1)
        L     R2,0(C,R2)
        CVD   R2,PKT    I PACKED:PKT
        UNFK  ZNT,PKT
        OI    ZNT+3,X'F0'
        L     R1,0(C,R1)    ADDR OF NAME:R1
        MVI   ZNT,X'4D'
        MVI   ZNT+4,X'5C'
CL     CLI   ZNT+1,X'F0'
        BNE   MOVE
        MVC   ZNT+1(4),ZNT+2
        B     CL
MOVE   MVC   0(R,R1),INAME    X(NNA) :NAME
        L     R2,2B(R13)
        MVI   12(13),255
        BR    R14
INAME  DC    X'E7'
ZNT    DS    XL4
        DS    X
        DC    X'4040'
PKT    DS    D
FO     EQU   0
```

```

F2      EQU    2
R0      EQU    0
R1      EQU    1
R2      EQU    2
R3      EQU    3
R4      EQU    4
R5      EQU    5
R6      EQU    6
R7      EQU    7
R13     EQU    13
R14     EQU    14
R15     EQU    15
        END
    
```

DRS. DACA FOLOSIREA RUTINELOR ELIMS,ELIM SI STDNME NU RESPECTA REGULA RE-
DISTRELOR DE LEGARE,SE CERE BALR R15,0 IN FATA LUI USING *,R15

DOUA EXEMPLE (DATELE PE CARTELE)

```

3      2
1      -1
    
```

STANDARD

(3E15.7)

MAXIM

```

          E+   -1.      E+30  2.      E-20
1.        E+   1.      E+30  -1.     E-20
-2.       E+  -.5      E+30  1.      E-20
X(2)      1.          E+20
X(2)      0.1        E-50
X(1)      .66666666  E-30
    
```

ENDLIMIT

V

```

8      4
0      0      0      0      0
    
```

STANDARD

(8F10.2)

MINIM

```

          -0.75      20.      -0.5      6.
1.          .25      -8.      -1.      9.
          1000000.   .5      -12.     -0.5      3.
    
```

```

          4CCCCC.          2.          -48.          -2.          12.
-1.          1.          1.          1.
X(4)
ENDLIMIT
R

```

CALCUL INTEGRAL

PROGRAMELE PENTRU CALCULUL INTEGRALELOR SIMPLE PREZENTATE AICI SE CARACTERI-
ZEAZA PRIN ACEIA CA REZULTATUL ESTE CERUT CU O PRECIZIE DORITA. CIND ACEASTA
PRECIZIE NU SE POATE ATINGE, SE OBTINE PRECIZIA MAXIMA POSIBILA.

SE POATE CONSULTA, IN SPECIAL, CARTEA METHODS OF NUMERICAL INTEGRATION DE PH.
DAVIS SI PH. RABINOWITZ, ED. ACADEMIC PRESS, 1975.

ORS. REZULTATELE OBTINUTE CU PUTINE VALORI ALE INTEGRANDULUI SA FIE VERIFI-
GATE (IMPUNIND NR. DE ITERATII SAU SUBDIVIZIND INTERVALUL), CACI AR PUTEA FI
VALORI PARTICULARE CARE VERIFICA CONDITIILE DE IESIRE DIN CALCUL.

C ***** **

FUNCTION RATEX(F,A,B,EPS,NT,NI,ERA,I)

C CALCULUL INTEGRALELOR, PRIN METODA ELLIPSE-STOER, CU O PRECIZIE CERUTA.

C PROGRAMUL NU CONSIDERA ERORILE DE ACUMULARE.

COMMON /RATCM/ TAB,M

C PARAMETRI (EFFECTIVI) DE INTRARE OBLIGATORII: F,A,B,EPS,NT,NI

C PARAMETRI (EFFECTIVI) DE IESIRE OBLIGATORII: ERA,I

C PARAMETRI (PRIN COMMON-UL RATCM) DE IESIRE OPTIONALI : TAB,M

C ACCESUL LA TAB SI M : COMMON /RATCM/ X,L IN PROGRAMUL CARE CHEAMA.

C NOTATIE: VI=VALOAREA INTEGRALEI=RATEX(F,A,....)

C * F= NUMELE INTEGRANDULUI, A SI B LIMITELE(SUP./INF.) DE INTEGRARE.

C PENTRU F SE CERE UN SUBPROGRAM FUNCTION CARE DA VALORILE LUI F(X) IN (A,B).

C * EPS = EROREA RELATIVA DORITA. DACA EPS ESTE DE NEATINS, ETA*TAB VA FI CONSTI-
C DERATA ERORARE ABSOLUTA DORITA.

C * NT= 2 LA 10 - DACA F(X) ARE DERIVATE CE NU SCAD IN (A,B), AR PUTEA FI MAI MUI
C NT MAI MARE (PESTE 10 NU SE POATE). ACOPERITOR, NT=10.

C POTRIVIND NT SE REDUC NR. OPERATIILOR SI ERORILE (DE ACUMULARE).

C CIND F(X) ARE DERIVATE DE ORDIN SUPERIOR F. MARI, TREBUIE MARIT NT.

C PL. VI: TAB MTC (SPRE ETA), NT. DIFERITI POT DA REZULTATE (VI,I,M) DIFERITE.

C CEIA CE INSAMNA CA ERORILE SINT MARI (FATA DE VI).

C * NI=3 LA 19 -NR. MINIM DE ITERATII. STANDAFD, NI=3. DACA UN REZULTAT NU

C PREZINTA INCHEREDERE, SE IMPUNE NR. DE ITERATII. I AR PUTEA INTOARCE ALTA VA-
C LUARE, INSA COMPARIND REZULTATELE(VI) VCM DECIDE DACA SINT BUNE.

C O NOUA ITERATIE CERE (CA 50% NOI VALORI F(X) DE CALCULAT.

C * ERA -ERORAREA ABSOLUTA (CALCULATA DE PROGRAM). DACA EPS E MARE, PRECIZIA
C -DATA DE ERA- POATE FI MAI BUNA DECIT CEA DORITA. DACA EPS E PREA MIC, SE VA
C OBTINE VI OPTIM POSTIBIL. LIMITA INFERICARA A LUI ERA ESTE $ETA * TAB$.
C * I =
L 0 - SE OBTINE PRECIZIA DORITA. ACEST RASPUNS CERE: $EPS * ABS(VI) .GE. FTA *
C TAB$ SAU $ABS(VI) .LE. MIN(ETA, EPS) * TAB$ (CIND VI SE VA CONSIDERA ZERO).
C 1 - SE OBTINE PRECIZIA MAXIMA POSIBILA (NU CEA DORITA). PT. A MARI PRECI-
C ZIA- DACA SE POATE MARI- DIVIZATI (A,B) SAU/SI POTRIVITI NT.
C 2 - SE OBTINE PREC. DORITA, CAR ERA ESTE NESIGURA (MAI ALES PT. VI: TAB
C MARE FATA DE ETA SAU/SI M MIC). SUBDIVIZATI (A,B) SAU/SI POTRIVITI NT.
C 3 - SE OBTINE PRECIZIA MAXIMA POSIBILA, CU ERA NESIGURA. NU CUMVA LUC-
C RATI CU ERORI F. MARI (INTEGRAND F. OSCILANT ETC). O DIVIZARE SAU/SI NT
C BINE POTRIVITE AR PUTEA FI SALVATOARE.
C * TAB - INTEGRALA DIN $ABS(F(X))$, CU ROMBERG MODIFICATA, OPRITA LA PRIMUL TERPEN.
C * M - NR. DE ITERATII CALCULATE (M. LE. 19). NR. DE VALORI F(X) CALCULATE = $3 * 2 * * ($
C $(M-1) : 2) + 1$ SAU $2 * 2 * * (M : 2) + 1$ PT. M IMPAR SAU PAR, RESPECTIV.
C OBS. IN TOATE CAZURILE (I=0,1,2 SAU 3), CIND VI: TAB E F. MIC (SPRE ETA) SAU/SI
C PRECIZIA LUI VI NU CONVINE, MARI TI PRECIZIA DE LUCRU (ETA MAI MIC).
C CIND VREM SA SUBDIVIZAM (A,B), SA-L IMPARTIM IN RAPOARTELE 2/5 SI 3/5. REZUL-
C TATUL (CU ACFLASI EPS) AR PUTEA FI MAI SLAB (DECIT NEDIVIZAT), INSA ERA POA-
C TE DEVENI SIGURA.
C F.F. IMPORTANT: F(X) FARA VALORI PARTICULARE (EGALE FTC) LA CAPETELE DIVIZI-
C UNILOR LUI (A,B) IN 2,3,4,6,8... INTERVALE (IESIRE CU I FALSI). O DIVIZARE
C BUNA PT. (A,B) SAU IMPUNEREA NR. MINIM DE ITERATII POATE EVITA SITUATIA.
C F(X) SA FIE DERIVABILA (DE 3-4 ORI) SI FARA PERIOADA B-A. F(X) SIMPLA, EPS
C MARE (PRECIZIE MICA) SAU B-A MIC NU SE RECOMANDA (TIMP DE CALCUL MARE).
C METODA SLABA PT. POLINOAME DE GRAD MIC SAU COMBINATII LINIARE DE ASTFEL DE
C POLINOAME CU FUNCTII PERIODICE, PE UN INTERVAL DE PERIODICITATE.
C PT. FUNCTII CU $MAX(ABS(F(X)))$ F. MICI, POATE APARE UNDERFLOW NESEMNICATIV.
C LOGICAL ODDM, AS, ACT, PREC, TAS
C TAS/FAS MARCHEAZA INCEPUTUL/SFIRSITUL MONOTONIEI - PE 3 PASI - TESTAT DE AS.
C DIMENSION DT(NT+1), C(2NT+1)
C DIMENSION DT(1), C(21), IDT(1)
C * CALCULATORUL ACCEPTA REPREZENTARILE INTERNE ALE NR. REALE CA NR. INTREGI VA-
C LIOSI SI RELATIILE DE ORDONARE (.LT., .LE., .GT., .GE.) INTRE NR. REALE NENEGATI-
C VE SE MENTIN INTRE ACESTE NR. INTREGI. ALTFEL, REAL * 4 IF NT, IV, IN, IDT, ICOT
C EQUIVALENCE (IENT, ENT), (IV, V), (IW, W), (IDT(1), DT(1)), (IDDT, DDT)
C * FTA - DIN INSTR. DATA, ESTE ERORAREA RELATIVA DE LUCRU IN SIMPLA PRECIZIE.
C MANTISA CU 6 CIFRE HEXAZECIMALE SI NORMALIZARE LA 16 ($ETA = 16 * * (1-6)$).
C DATA ETA / .95E-6/


```
C * D-VE LUNGHE SIMPLA- 'D(1)=9.,7.,D(2)=4.,D(1)=4*D(1-2),I=3,10,D(11)=2.25,
C D(12)=4.,D(1)=4*D(1-2),I=13,21
  DATA 0/1.777778,4.,7.111111,16.,28.444444,64.,113.7778,256.,
1455.11111,1024.,2.25,4.,9.,16.,36.,64.,144.,256.,576.,1024.,2304./
  XEPS=ABS(EPS)
  SM=ETA
  N=2
  NN=3
  NII=NI-3
  BA=B-A
  GR=0.
  T1=0.
  T1A=0.
  T1AB=0.
C T1AB=T2B*ARS(BA) DAR NU E NECESAR.
  V=F(A)
  W=F(B)
  T2A=(V+W)*.5
  T2=T2A
  T2B=(ABS(V)+ABS(W))*5
  T2AB=T2B
  C=T2*BA
  DT(1)=C
  DO 1 I=2,11
1 IDT(I)=0
C DT CONTINE SUMA TRAPEZ. SI I CORECTII V,PT. A OBTINE T(M,I) (I.LE.NT).
  TAS=.FALSE.
C * PREC=TAS CACA EXECUTIA CADE CU VALGARE LOGICA INVALIDA IN INSTRUC. 92.
  ODDM=.TRUE.
C MIN(HM)=(B-A)/2*((M+1)/2)
  DO 13 M=1,19
  HM=BA/M
  IF(ODDM) GO TO 3
  DO 2 I=1,M,6
  W=I*HM
  V=F(A+W)
  D1=F(B-W)
  T1A=T(A+ABS(V)+ABS(D1))
2 T1=T1+V+D1
C T1-SUMA VAL. F(X) PT. N=3+2*K CURENT,CARE NU INTRA IN T2A
```

```
C T2A-SUMA VAL. FIX) PT. N CURENT CARE NU SINT IN T1(LA CAPETE *.5)
  T=(T1+T2A)*HM
C T ESTE VAL. CURENTA A SUMEI TRAP.
  T1AB=(T1A+T2AB)*ABS(HM)
C T1AB-INTEGRALA DIN ABS(FIX)),PT. N CURENT,PRIN TRAPEZE.
  TAB=T1AB+.3*(T1AB-TAB1)
C T2 - SUMA TUTUROR VAL. F(X)-LA CAPETE *.5- PT. N=2**K PRECEDENT
  T2A=T2
  T2AB=T2B
C T2AB- SUMA VAL. ABSOLUTE PT. N=3*2**K URMATOR CARE NU VOR FI IN T1A
  J=11
C J=M*X(NT)+1
  GO TO 3
3 DO 4 I=1,N.2
  W=F(A+I*HM)
  T2=T2+W
4 T2B=T2B+ABS(W)
C T2-SUMA TUTUROR VAL. F(X)-LA CAPETE *.5- PT. N=2**K CURENT.
C T2B-SUMA TUTUROR VAL. ABS(FIX))-LA CAPETE *.5- PT. N CURENT.
  T=T2*HM
  TAB1=T2B*ABS(HM)
C TAB1-INTEGRALA DIN ABS(FIX)),PT. N CURENT,PRIN TRAPEZE.
  TAB=TAB1+1.205714*(TAB1-T1AB)
  J=1
5 DDT=DT(1)
C DDT ESTE VAL. PRECEC. A SUMEI TRAP. ✓
  DT(1)=T
C DT(1) ESTE VAL. ACTUALA A SUMEI TRAP.
  ENT=T
  K=1
  MR=NT+J-1
  IF(M.GT.NT) GO TO 6
  MR=MR+J-1
  ERA=D(MR)
  D(MR)=D(MR+1)
5 DO 9 I=J,MR
  D1=D(I)+DDT
  V=D1-ENT
C URICE NIMITOR NE. 0 E PUN (ETA SI CI POTANASC CORECTIA).
  IF(IV.NE.0) GO TO 7
```

```
IV=U
IENT=0
GO TO 8
7 W=(ENT-DDT)/V
V=W*ENT
T=T+V
C T ESTE IN LINIA M SI COLOANA I-J+2(TABLOUL T).
ENT=W*D1
8 K=K+1
IDDT=IDT(K)
9 IDT(K)=IV
C DT CONTINE NOILE VALORI
ENT=C
C=T
D G-VALOAREA ACTUALA, ENT-VALOAREA PRECEDENTA.
IF(M.GT.MT) GO TO 10
T=T-V
D(MR)=ERA
10 IF(M.LE.MII) GO TO 12
C ENT SI T SINT PE C COLOANA IN LINII SUCCESIVE
T=T-ENT+T
C T=U, ENT=T DIN CRITERIUL DE "BRACKETING"
W=ETA+TAB
IF(ENT.GE.T) GO TO 11
IV=IENT
ENT=T
T=V
C SE LA .LE. SI .GE. PT. FIX) SIMPLE (PCLINAME DE GRAD MIC ETC).
11 ACT=ENT.LE.GR.AND.T.GE.SM
AS=ACT.AND.PREC
IF(AS) TAS=.TRUE.
C TAS=.TRUE. CACA A FOST MONOTONIE PE 3 ITERATII.
IF(.NOT.AS.AND.TAS) GO TO 16
ERA=.5*(ENT-T)
RATEX=.5*(T+ENT)
PREC=ACT
V=XEPS*ABS(C)
C SE CAUTA ER.REL. SCRITA (EPS+ABS(C).GT.ETA+TAB) SAU ER.REL.(ABS.) MINIMA PO-
C SIBILA (ABS(C).LE.ETA+TAB).
ACT=ERA.LG.V
```

```
IF(IV.LT.IV) ACT=ERA.LE.W
IF(AS.AND.ACT) GO TO 14
GR=ENT
SM=T
12 ODDM=.NOT.ODDM
I=N
N=NM
13 NM=I+I
C CUM SE OBTINE PRECIZIA INVESTIGATA/CPTIPA.
I=3
IF(ACT.AND.IV.GE.IV) I=2
GO TO 17
C SE OBTINE PRECIZIA INVESTIGATA.
14 I=0
C ERORAREA RELATIVA DORITA NEATINSA.
IF(IV.LT.IV) I=1
GO TO 17
C SE OBTINE OPTIMUL INVESTIGAT.
16 I=1
C CUM VI=TAB E F.MIC.VI SE CONSIDERA BUNA(FARA MONOTONIE).
17 IF(ABS(RATEX).LE.XEPS*TAB) I=0
IF(ERA.LT.W) ERA=W
RETURN
END

C ***** ** *****
C REAL FUNCTION DRATEX*(F,A,B,EPS,NT,NI,ERR,I)
C CALCULUL INTEGRALELOR,PRIN METODA PILLRSCH-STOER, CU O PRECIZIE CERUTA.
C PROGRAMUL NU CONSIDERA ERORILE DE ACUPULARE.
C IMPLICIT REAL*(A-H,O-Z)
C COMMON /DRATCH/ TAB,M
C PARAMETRI (EFFECTIVI) DE INTRARE OBLIGATORII: F,A,B,EPS,NT,NI
C PARAMETRI (EFFECTIVI) DE IESIRE OBLIGATORII: ERA,I
C ACCESUL LA TAB SI M : COMMON /DRATCH/ X,L IN PROGRAMUL CARE CHEAMA.
C NOTATIE: VI= VALCAREA INTEGRALEI= RATEX(F,A,B,...)
C * F= NUMELE INTEGRANDULUI, A SI B LIMITELE(SUP./INF.) DE INTEGRALE.
C PENTRU F SE CERE UN SUBPROGRAM FUNCTION CARE DA VALORILE LUI F(X) IN (A,B).
C * DRATEX,F(X) A SI B SINT DE TIP REAL*8, EPS,ERR SI TAB DE TIP REAL*4, NT,NI
C SI M DE TIP INTEGER.
C * EPS = ERORAREA RELATIVA DORITA. DACA EPS ESTE DE NEATINSA,ETA*TAB VA FI CONSI-
```

ERATA ERORARE ABSCLUTA DORITA.
C PROGRAMUL FORTEAZA EPS .GE. ETA. PRECIZIA(NR. DE CIFRE SEMNIFICATIVE EXACTE)
C REZULTA DIN NR. DE CIFRE EXACTE IN VALORILE F(X), NR. DE CIFRE SEMNIFICATIVE
C PIERDUTE (INTEGRAND OSCILANT) SI DIN ERORAREA DE ACUMULARE.
C * NT= 2 LA 10 - DACA F(X) ARE DERIVATE CE NU SCAD IN (A,B), AR PUTEA FI MAI BUN
C NT MAI MARE (PESTE 10 NU SE POATE). ACCEPITUR, NT=10.
C POTRIVIND NT SE REDUC NR. OPERATIILOR SI ERORILE (DE ACUMULARE).
C CIND F(X) ARE DERIVATE DE ORDIN SUPERIOR F. MARI, TREBUIE MARIT NT.
C PT. VI: TAB MIC (SPRE ETA), NT DIFERITI POT DA REZULTATE (VI, I, M) DIFERITE,
C CEA CE INSAMNA CA ERORILE SINT MAI (FATA DE VI).
C * NI=3 LA 19 -NR. MINIM DE ITERATII. STANCARO, NI=3. DACA UN REZULTAT NU
C PREZINTA INCREDERE, SE IMPUNE NR. DE ITERATII. I AR PUTEA INTOARCE ALTA VA-
C LOARE. INSA COMPARIND REZULTATELE (VI) VOM DECIDE DACA SINT BUNE.
C O NOUA ITERATIE CERE CCA 50% NOI VALORI F(X) DE CALCULAT.
C * ERR -ERORAREA ABSCLUTA (CALCULATA DE PROGRAM). DACA EPS E MARE, PRECIZIA
C -DATA DE ERR-POATE FI MULT MAI BUNA DECIT CEA DORITA (PATEX CAUTA EPR SIGURA)
C DACA EPS E PEA MIC, SE VA OBTINE VI OPTIM POSIBIL.
C LIMITA INFERIOARA A ERORII ABSCLUTE (ERR) ESTE MAX(EPS*ABS(VI), ETA*TAB).
C * I=
C 0 - SE OBTINE PRECIZIA DORITA. ACEST FASPUNS CERE: EPS*ABS(VI) .GE. ETA*
C TAB SAU ABS(VI) .LE. ETA*TAB (CIND VI SE VA CONSIDERA ZERO).
C 1 - SE OBTINE PRECIZIA MAXIMA POSIBILA (NU CEA DORITA). PT. A MARI PRECI-
C ZIA SE DIVIDE (A,B) SAU/SI SE POTRIVESTE NT.
C 2 - SE OBTINE PREC. DORITA, DAR ERR ESTE NESIGURA (MAI ALES PT. VI: TAB
C MARE FATA DE ETA SAU/SI M MIC). SUBLIVIZATI (A,B) SAU/SI POTRIVITI NT
C 3 - SE OBTINE PRECIZIA MAXIMA POSIBILA, CU ERR NESIGURA. NU CUMVA LUC-
C RATI CU ERORI F. MARI (INTEGRAND F. OSCILANT ETC). C DIVIZARE SAU/SI NT
C BINE POTRIVITE AR PUTEA FI SALVATCARE.
C * TAB - INTEGRALA DIN ABS(F(X)), CU COMPERG MODIFICATA, OPRITA LA PRIMUL TERPEN.
C * M - NR. DE ITERATII CALCULATE (M.LE. 25). NR. DE VALORI F(X) CALCULATE = 3*2*(
C (M-1):2)+1 SAU 2*2*((M:2)+1) PT. M IMPAR SAU PAR, RESPECTIV.
C OBS. IN TOATE CAZURILE (I=0, 1, 2 SAU 3), CIND VI: TAB E F. MIC (SPRE ETA) SAU/SI
C PRECIZIA LUI VI NU CONVINE, MARITI PRECIZIA DE LUCRU (ETA MAI MIC).
C CIND VREM SA SUBDIVIZAM (A,B), SA-L IMPARTIM IN RAPGARTELE 2/5 SI 3/5. REZUL-
C TATUL (CU ACELASI EPS) AR PUTEA FI MAI SLAB (DECIT NEDIVIZAT), INSA ERR POA-
C TE DEVENTI SIGURA.
C F.F. IMPORTANT: F(X) FARA VALORI PARTICULARE (EGALE ETC) LA CAPETELE DIVIZI-
C UNILOR LUI (A,B) IN 2, 3, 4, 6, 8, ... INTERVALE (IESIRE CU I FALSII). C DIVIZARE
C MINA PT. (A,B) SAU IMPUNEREA NR. MINIM DE ITERATII POATE EVITA SITUATIA.
C F(X) SA FIE DERIVABILA (DE 3-4 ORI) SI FARA PERIODADA B-A. F(X) SIMPLA, EPS

```
C MARE (PRECIZIE MICA) SAU R-A MIC NU SE RECOMMANDA (TIMP DE CALCUL MARE).
C METODA SLABA PT. POLINOAME DE GRAD MIC SAU COMBINATII LINIARE DE ASTFEL DE
C POLINOAME CU FUNCTII PERIODICE, PE UN INTERVAL DE PERIODICITATE.
C PT. FUNCTII CU MAX(ABS(F(X))) F.MICI, PCATE APARE UNDERFLOW NESEMNICATIV.
C * CALCULATORUL ACCEPTEA REPREZENTARILE INTERNE ALE NR. REALE CA NR. INTREGI VA
C LIIDE SI RELATIILE DE ORDONARE (.LT.,.LE.,.GT.,.GE.) INTRE NR. REALE NENEGAT
C VE SE PASTREAZA INTRE ACESTE NR. INTREGI, ALTFEL, REAL*4 IENT,IV,IV,IDI
REAL*4 EPS,XEPS,ERR,AERR,TAB,TB,T1A,TAB1,T2A,T1A,VR,VR,CR,HR,
I DIR,ETA
LOGICAL ODDM,AS,ACT,PREC,TAS
DIMENSION DT(11),C(21)
EQUIVALENCE (IENT,ENT),(IDI,DIR,D),(IV,VR,V),(IV,VR,W),
I (HR,HP),(CR,C)
C * ETA -DIN INSTR. DATA,ESTE ERORUL RELATIV DE LUCRU IN PRECIZIE DUBLA.
C MANTISE CU 14 CIFRE HEXAZECIMALE SI NORMALIZARE LA 10: ETA=16*(1-14)
DATA ETA/2.22E-16/
C * D -PE LUNGIME DUBLA- D(1)=7.,7.,D(2)=4.,D(1)=4*(1-2),I=3,10,D(11)=2.25,
C D(12)=4.,D(11)=4*D(1-2),I=13,21
DATA D/1.777777777777778D0,4.,7.111111111111111D0,16.,23.444444
1444444444D0,64.,113.777777777777777D0,256.,455.1111111111111D0,
21024.,2.25D0,9.,16.,36.,64.,144.,256.,576.,1024.,2304./
XEPS=ABS(EPS)
IF(XEPS.LT.ETA) XEPS=ETA
SM=>D)
N=2
NN=3
NI=NI-3
BA=B-4
GR=0.00
T1=).00
T1A=).
T1A=).
V=F(A)
W=F(B)
T2A=(V+W)*.5D0
T2=T2A
TB=(ABS(VR)+ABS(WR))*4.5
T2A=TB
C=T2*BA
C(11)=C
```

1. 1.0) 1 I=2,11
EL DL 1 DT(I)=3.0E0
TAS=.FALSE.
ATIV. 1002A=.TRUE.
EGI VE DO 13 M=1,25
NEGAT MM=BA/M
1 IF(GOCM) GO TO 3
DO 2 I=1,N-6
W=I*MM
V=F(A+W)
DI=F(B-W)
T1A=11A+ABS(V)+ABS(DI)
2 T1=T1+V+D)
T=(T1+T2A)*MM
T1AB=(T1A+T2AB)*ABS(MM)
T2A=T1AB+.P*(T1AB-TAB1)
T2A=T2
T2AB=T2
J=11
GO TO 5
3 DO 4 I=1,N,2
4=F(A+I*MM)
T2=T2+W
4 TR=TB+ABS(MR)
T=T2*MM
TAB1=TB*ABS(MR)
T2B=TAB1+1.28571*(T2B-T1AB)
J=1
5 DDT=DT(I)
DT(I)=T
ENT=1
K=1
MK=NT+J-1
IF(M.G.(NT)) GO TO 6
MK=M+J-1
D(MK)=D(MR)
D(MK)=D(MR+1)
6 DO 9 I=J,MR
DI=D(I)*DDT
V=DI-ENT

```
IF (IV.NE.0) GO TO 7
V=0.00
ENT=0.00
GJ TO 3
7 W=(ENT-DDT)/V
V=W*ENT
I=T+V
ENT=W*D1
8 K=K+1
DDT=DT(K)
9 DT(K)=V
FNT=C
C=T
IF (M.GT.NT) GO TO 10
T=T-V
D(MR)=DRATEX
10 IF (M.LE.NT) GO TO 12
DI=FNT
V=T-FNT
T=T+V
ERR=ABS(VR)
WR=ETA*TAH
IF (ENT.GE.T) GO TO 11
V=FNT
FNT=I
I=V
11 ACT=ENT.LE.GR.AND.T.GE.SM
AS=ACT.AND.PREC
IF (AS) TAS=.TRUE.
IF (.NOT.AS.AND.TAS) GO TO 15
PREC=ACT
VR=XEPS*ABS(CR)
ACT=ERR.LE.VR
IF (IV.LT.IW) ACT=ERR.LE.WR
IF (AS.AND.ACT) GO TO 14
GR=FNT
SM=I
AERR=ERR
12 ODDM=.NOT.ODDM
I=N
```



```
N=NN
13 NN=I+I
    I=3
    IF(ACT.AND.IV.GE.IW) I=2
    GO TO 16
14 I=0
    IF(IV.LT.IW) I=1
    GO TO 16
15 ERR=AERR
    C=D1
    I=1
16 IF(ABS(CR).LE.WR) I=C
17 DRATEX=C
    IF(ERR.LT.WR) ERR=WR
    RETURN
    END
```

```
C ***** ** ** *****
```

FUNCTION PCTMED(F,A,B,EPS,NI,ERR,I)

```
C CALCULUL INTEGRALEROR SIMPLU-ROMBERG MIDPOINT RULE-, CU O PRECIZIE CERUTA,
C PROGRAMUL NU TINE CONT DE ERORILE DE ACUMULARE.
    COMMON /MEDCM/ TAB,K
C F,A,B,EPS,NI SINT PARAMETRI DE INTRARE (ADUC VALORI).
C ERR,I,TAB SI K SINT PARAMETRI DE IESIRE (INTORC VALORI CALCULATE)
C TAB SI K SINT OPTIONALI SI SE FOLCSESC PRIN INSTR. COMMON /MEDCM/ X,L IN
C PROGRAMUL CARE CHEAMA RMIDPT.
C * F,A SI B - INTEGRANDUL SI LIMITELE (SUP./INF.) DE INTEGRARE.
C * EPS - ERORAREA RELATIVA DORITA. DACA EPS ESTE DE NEATINS,ETA*TAB VA FI CONST-
C DERATA ERORARE ABSOLUTA DORITA (SE CAUTA PRECIZIE MAXIMA POSIBILA).
C * NI= 1 LA 10 - NR. MINIM DE ITERATII (INJUMATATIRI DE INTERVALE) DE CALCULAT.
C * ERR - ERORARE ABSOLUTA CALCULATA DE PROGRAM.
C * I=
C 0. SE OBTINE PRECIZIA CERUTA.
C 1. NU SE OBTINE PRECIZIA CERUTA.
C * TAB - INTEGRALA DIN ABS(F(X)),CALCULATA DE PROGRAM PRIN METODA TRAPEZELOR.
C * K - NR. DE ITERATII CALCULATE. NR. DE VALORI F(X) CALCULATE= 2**K+1
    DIMENSION T(11),TM(11),U(10),VT(10)
    DATA ETA/.95E-6/
C U(1)=1/(4**I-1),VT(1)=(4**I/2-1)/(4**I-1)
    DATA U/.3333333,.6666666E-1,.15673015E-1,.3321063E-2,.9775171E-3
```

```
1. .24420024E-3, .6103889E-4, .15259022E-4, .3814712E-5, .9236752E-6/  
VT=VT/.333333, .4666665, .4920634, .4980392, .49951124, .4996719,  
1. .49996948, .49999237, .4999981, .49999952/  
XEPS=ABS(EPS)  
H=(B-A)*.5  
V=F(A)  
W=F(B)  
T(1)=(V+W)*H  
TM(1)=F(.5*(A+H))*H*.5  
TAB=.5*(ABS(H)*(ABS(V)+ABS(W))+ABS(TM(1)))  
NX=4  
DO 1 K=1,10  
H=.5*H  
T(K+1)=.5*(T(K)+TM(K))  
TM(K+1)=0.  
SAB=0.  
DO 1 I=1,NX,2  
V=F(A+I*H)  
SAB=SAB+ABS(V)  
1 TM(K+1)=TM(K+1)+V  
TM(K+1)=H*TM(K+1)*2.  
2 TM(K+1) SI T(K+1) SINT SUMLE TRAPEZELOR PT. DIVIZIUNEA KL  
T(1)=TM(1)+VT(K)*(T(1)-TM(1))  
3 TM(1) SI T(1) SINT PE DIAGONALELE MARI.  
TAB=.5*TAB+ABS(F)+SAB  
J=K  
DO 2 I=1,K  
TM(J)=(TM(J+1)-TM(J))*U(1)+TM(J+1)  
IF(J.EQ.1) GO TO 2  
T(J)=.5*(T(J-1)+TM(J-1))  
2 J=J-1  
3 NX=NX+NX  
IF(K.LT.NI) GO TO 2  
W=EIATAB  
ERA=.5*ABS(T(1)-TM(1))  
PCTMFE=.5*(T(1)+TM(1))  
V=XEPS+ABS(PCTMFE)  
IF(V.LT.W) V=W  
IF(ERA.LE.V) GO TO 4  
5 CONTINUE
```

```
I=1
GO TO 7
C I=0
IF(XEPS*ABS(PCTMED).LT.W) I=1
7 IF(ABS(PCTMED).LE.W) I=0
IF(ERA.LT.W) ERA=W
RETURN
END
```

```
C *****
C FUNCTION BRAKT(F,A,B,EPS,NI,ERA,I)
C CALCULUL INTEGRALELOR (BULIRSCH BRACKETING CRITERION), CU O PRECIZIE CERUTA.
C PROGRAMUL NU TINE CONT DE ERORILE DE ACUMULARE.
C COMMON /BRACM/ TAB,K
C F,A,B,EPS SI NI SINT PARAMETRI DE INTRARE (ADUC VALORI).
C ERA,I,TAB SI K SINT PARAMETRI DE IESIRE (INTORC VALORI CALCULATE)
C TAB SI K SINT OPTIONALI SI SE FOLOSESC PRIN INSTR. COMMON /BRACM/ X,L IN
C PROGRAMUL CARE CHEAMA ROMBG.
C * F,A SI B - INTEGRANDUL SI LIMITELE (SUP./INF.) DE INTEGRARE.
C * EPS - ERORAREA RELATIVA DORITA. DACA EPS ESTE DE NEATINS,ETA*TAB VA FI CONSI-
C DERATA ERORARE ABSOLUTA DORITA (SE CAUTA PRECIZIE MAXIMA POSIBILA).
C * NI = 4 LA 12 - NR. MINIM DE ITERATII (INJUMATATIRI DE INTERVALE) DE CALCULAT.
C * ERA - ERORARE ABSOLUTA CALCULATA DE PROGRAM.
C * I =
C 0. SE OBTINE PRECIZIA CERUTA.
C 1. SE OBTINE REZULTATUL OPTIM POSIBIL (NU CEL DORIT),
C 2. SE OBTINE PRECIZIA CERUTA ,DAR ERA ESTE NESIGURA.
C 3. SE FAC TOATE ITERATIILE, NU SE OBTINE PRECIZIA SI ERA E NESIGURA.
C * TAB - INTEGRALA DIN ABS(F(X)),CALCULATA DE PROGRAM PRIN METODA TRAPEZELOR.
C * K - NR. DE ITERATII CALCULATE. NR. DE VALORI F(X) CALCULATE= 2**K+1.
C PENTRU DETALII DE FOLOSIRE VEGETI PROGRAMUL RATEX.
C DIMENSION TD(NI+1),V(NI)
C DIMENSION TD(12),V(12)
C LOGICAL AS,ACF,PREC,TAS
C * ETA - ERORAREA RELATIVA DE LUCRU (MAXIMA).
C DATA ETA/.95E-6/
C V(1)=1/(4**I-1) SA FIE CU PRECIZIA DE LUCRU.
C DATA V/.3333332,.6666666E-1,.15673015E-1,.39215686E-2,.9775171E-3,
C 1.,24420024E-3,.6103892E-4,.15259022E-4,.3914712E-5,.9536752E-6,
C 2.,2554126E-6,.59605E-7/
```

```
XEPS=ABS(EPS)
NII=NI-3
IF(NII.LT.1) NII=1
TAS=.FALSE.
C * PREC=TAS  DACA EXECUTIA CADU CU VALCARE LOGICA INVALIDA (ISTR. 40)
H=B-A
S=F(A)
SS=F(B)
TD(1)=.5*H*(S+SS)
C TD(1) ESTE PRIMUL TERMEN DIN TABLOUL TD.
TAB=ABS(H)*(ABS(S)+ABS(SS))*0.5
SM=.5
GR=0.
NX=1
C * LIMITA SUP. PT. K SA NU DEPASEASCA DIMENSIUNEA LUI V.
DO 1 K=1,12
H=H*.5
S=0.
SS=0.
NA=NX+NX
DO 2 I=1,NX,2
U=F(A+H*I)
SS=SS+ABS(U)
2 S=S+U
C S ESTE SUMA VAL. F(X) PT. NOILE PUNCTE ALE DIVIZIUNII
TD(K+1)=.5*TD(K)+H*S
C TD(K+1) ESTE SUMA TRAPEZELOR (PRIMA CORDANA DIN TD).
TAB=TAB*.5+SS*ABS(H)
DO 3 J=1,K
I=K+1-J
3 TD(I)=TD(I+1)+(TD(I)-TD(I+1))*V(J)
IF(K.LE.NII) GO TO 1
U=TD(2)+(1.+2.*V(K))*(TD(2)-TDP)
IF(U.GE.TDP) GO TO 4
S=U
U=TDP
TDP=S
4 ACT=TDP.GE.SM.AND.U.LE.GR
C MONITONIE IN U SI TD, PE 3 ITERATII SUCCESIVE.
C U SI TDP LIMITEAZA (EPARTE) LARG APROXIMATIA ROMBERG.
```

```
AS=ACT.AND.PREC
IF(AS) TAS=.TRUE.
SS=ETA*TAB
IF(.NOT.AS.AND.TAS) GO TO 6
PREL=ACT
ERA=.5*(U-TDP)
BRAKT=.5*(U+TDP)
S=XEPS*ABS(BRAKT)
ACT=ERA.LE.S
IF(S.LT.SS) ACT=ERA.LE.SS
IF(AS.AND.ACT) GO TO 5
SM=TDP
GR=U
1 TOP=TD(1)
I=3
IF(ACT.AND.S.GE.SS) I=2
GO TO 7
5 I=0
IF(S.LT.SS) I=1
GO TO 7
6 I=1
7 IF(ABS(BRAKT).LE.SS) I=0
IF(ERA.LT.SS) ERA=SS
RETURN
END
```

```
C **** * ***** ****
C FUNCTION ROMBG(F,A,B,EPS,NI,ERA,I)
C CALCULUL INTEGRALELOR, PRIN METODA ROMBERG, CU O PRECIZIE CERUTA.
C PROGRAMUL NU TINE CONT DE ERORILE DE ACUMULARE.
C COMMON /ROMCM/ TAB,K
C F,A,B,EPS SI NI SINT PARAMETRI DE INTRARE (ADUC VALORI).
C ERA,I,TAB SI K SINT PARAMETRI DE IESIRE (INTORC VALORI CALCULATE)
C TAB SI K SINT OPTIONALI SI SE FOLOSESC PRIN INSTR. COMMON /BRACH/ X,L IN
C PROGRAMUL CARE CHEAMA ROMBG.
C * F,A SI B - INTEGRANDUL SI LIMITELE (SLP./INF.) DE INTEGRARE.
C * EPS - ERORAREA RELATIVA DORITA. DACA EPS ESTE DE NEATINS,ETA*TAB VA FI CONSI-
C DERATA ERORARE ABSOLUTA DORITA (SE CAUTA PRECIZIE MAXIMA POSIBILA).
C * NI= 2 LA 12 - NR. MINIM DE ITERATII (INJUMATATIRI DE INTERVALE) DE CALCULAT.
C * ERA - ERORARE ABSOLUTA CALCULATA DE PROGRAM (PE 3 ITERATI SUCCESIVE DEPE DIA-
```

```
C (GONALA MARE). ERA NU ESTE FOARTE SICURA.
C * I =
C 0. SE OBTINE PRECIZIA CERUTA.
C 1. NU SE OBTINE PRECIZIA CERUTA.
C * TAB - INTEGRALA DIN ABS(F(X)), CALCULATA DE PROGRAM PRIN METODA TRAPEZELOR.
C * K - NR. DE ITERATII CALCULATE. NR. DE VALORI F(X) CALCULATE = 2**K+1.
C PENTRU DETALII DE FOLOSIRE VEDETI PROGRAMUL RATEDX.
C DIMENSION C(N+1),V(1)
  DIMENSION Q(13),V(12)
C * ETA - ERORAREA RELATIVA DE LUCRU (MAXIMA).
  DATA ETA/.95E-6/
C V(I) = 1/(4**I-1) SA FIE CU PRECIZIA DE LUCRU.
  DATA V/.3333333,.6666666E-1,.15873015E-1,.3921586E-2,.775171E-3
  1..24420024E-3,.6103935E-4,.15259022E-4,.3819712E-5,.9536752E-6,
  2..2384186E-6,.59605E-7/
  XEPS=ABS(EPS)
  H=H-A
  S=F(A)
  SS=F(R)
  Q(1) = .5*H*(S+SS)
C Q(1) ESTE PRIMUL TERMEN DIN TABLOUL T.
  TDP=Q(1)
  TAB=ABS(H)*(ABS(S)+ABS(SS))*0.5
  NX=1
* LENTA SUP. PT. K (PR PUTEA FI PARAMETRU) SA NU DEPASEASCA DIMENS. LUI V.
  DO 1 K=1,12
    H=.5*H
    S=0.
    SS=0.
    NX=NX+NX
  DO 2 I=1,NX+2
    ERA=F(A+I*H)
    SS=SS+ABS(ERA)
  2 S=S+FKRA
C S ESTE SUMA VALORILOR F(K) PT. PCT. NCI ALE DIVIZIUNII.
  Q(K+1) = .5*Q(K)+H*S
C Q(K+1) ESTE SUMA TRAPEZELOR (PRIMA CULOANA A TABLOULUI T).
  TAB=TAB*.5+SS*ABS(H)
  DO 3 J=1,K
    I=K+1-J
```

```
3 U(I)=U(I+1)+I*(I+1)-O(I), V(J)
   ERA=.5*(ABS(O(I))-TDP)+ABS(TDP-TDAP)
C   ERA ESTE SEMISUMA DIFERENTELOR A 3 ITERATII SUCCESIVE.
   IF(K.LT.NI) GO TO 4
   SS=ETA*TAB
   S=XEPS*ABS(O(I))
   IF(SS.GT.S) S=SS
   IF(ERA.LE.S) GO TO 5
4 TDAP=TDP
  1 TDP=O(I)
  I=1
  GO TO 6
5 I=0
  IF(XEPS*ABS(O(I)).LT.SS) I=1
6 IF(ERA.LT.SS) ERA=SS
  IF(ABS(O(I)).LE.SS) I=0
  ROMAG=C(I)
  RETURN
END
```

IRADIAREA JUMPEILOR MALIGNE.

DESCRIEREA PROBLEMEI IN CAZUL MEDIULUI OMOCEN ESTE FACUTA IN REVUE ROMANIAI DE PHYSIQUE, TOM 20, NO. 1, 1975, UNDE SE DA SI O BIBLIOGRAFIE.

PREZENTUL PROGRAM ADMITE EXISTENTA UNOR ZONE NEOMOGENE SI CALCULEAZA DOZELE INTA-UN PLAN-SECTIUNE. POT FI CEL MULT 5 ZONE NEOMOGENE CU SECTIUNI ELIPSE, CU DEFINIREA NEOMOGENITATII PRIN MATRICEA TARM. ELIPSELE SE DAU PRIN 5 PUNCTE. FIXAREA SECTIUNII DE CALCULAT SE FACE PE UN SISTEM DE AXE RECTANGULARE XCY, ALTE DUA DORINDTA, DE PREFERAT INSA CU ORIGINA IN ZONA PUTERNIC IRADIATA.

I. MEDIU OMOCEN

SE INTRODUC PE CARTELE, CONFORM CU FORMATELE INDICATE:

1) OMOCEN .FORMAT(AE)

2) NSURSE, NRELPS, NCJ, YMX, YMN .FORMAT(2I2, 2F4, 1)

NSURSE=NR. DE SURSE, NRELPS=0, NCJ= 1 DACA ESTE ULTIMUL PLAN DE CALCULAT SAU 0 DACA NU ESTE ULTIMUL PLAN DE CALCULAT, IN ACELASI JOB, YMX=ORDONATA MAXIMA A DOMENIULUI IN CARE SE CALCULEAZA DOZE, YMN=ORDONATA MINIMA A DOMENIULUI DE CALCUL

3) PASX, PASY .FORMAT(2F4, 2) RETEAUA PT. CARE SE CALCULEAZA DOZE. PASX POATE FI 0.5 SAU 0.25 CM. PASY POATE FI ORICIT, DAR NEGATIV (SE INCEPE CU YMAX).

4) SURSELE, CONFORM CU DESCRIEREA DE MAI JOS, C SUPSA/CARTELA.

II. MEDIU NEOMOCEN

SE ADMIT CEL MULT 3 SURSE PARALELE (RAZA CENTRALA) CU AXA CY SI CEL MULT UNA

CARE FACE UN UNGHII DE 45 GRADE CU ACESTEA (PARALELA PRIMEI BISECTIONARI). DACA NU EXISTA SURSA NEPARALELA CU DY, SE SCHIMBA INSTRUCIUNEA IF(I-1) IN IF(I-5), ASA CUM SE APATA IN PROGRAM (DELIMITAT CU ASTERISCURII).

1) NEMOGEN, FORMAT(4F)

2) TARM(27,5) ,FORMAT(10F5,3); CITE 10 PE CARTELA, ULTIMA CONTINE 5 ELEMENTE.

3) NSURSE, NREPLS, MCJ, YPX, YMN, , FORMAT(3I2,2F4,1), CA LA MEDIU OMOGEN, DAR NREPLS ESTE DIFFERIT DE ZERO.

4) CAPK(I), I=1, NSURSE, ,FORMAT(4F5,2). SINT COORDONATELE IN ORIGINA ALE LINIILOR PERPENDICULARE RAZELOR CENTRALE ALE FIECAREI SURSE, IN PUNCTUL DE INTRARE IN ORGANISM. (LINIILE AU ECUATIILE $Y=X+C$ SAU $Y=C$). ORDINEA SA FIE ACEASI CA A SURSELOR INTRODUSE LA PUNCTUL 6+NREPLS ETC.

5) 5 PUNCTE IN PLANUL XOY, CARE DEFINESC O ELIPSA, CU FORMAT(10F5,2)

.....

6) NREPLS) 5 PUNCTE, CARE DEFINESC ULTIMA ELIPSA, CU FORMAT(10F5,2).

ESTE INDIFERENTA ORDINEA ELIPSELOR.

7) NREPLS) PASX, PASY, ,FORMAT(2F4,2). ANALOG CU MEDIUL OMOGEN.

8) NREPLS) SURSA NEPARALELA CU DY, CIND EXISTA, CU FORMAT(8F4,1)

.....

9) NREPLS+NSURSE) ULTIMA SURSA, CU FORMAT(8F4,1)

OBS. DACA INTR-UN JOB SE REZOLVA MAI MULTE PROBLEME (SAU MAI MULTE PLANE PT. ACEASTI PROBLEMA), SE REPETA TOATE DATELE. ATENTIE LA LOCUL SURSEI NEPARALELE.

SURSELE

DEP=DISTANTA FOCUS-PIELE, IN CM

ALAT=DESCHIDERA (LATIMEA) CIMPLULUI, IN CM

ADC=ADINCINEA CIMPLULUI, IN CM

RC, FI= COORDONATELE POLARE ALE PUNCTULUI DE INCIDENTA AL RAZEI CENTRALE CU ORGANISMUL

TETA=UNGHIIUL DINTRE AXA OX+ SI RAZA CENTRALA (ORIENTATA FOCUS-PIELE)

ALFA=UNGHIIUL DINTRE RAZA CENTRALA SI NORMALA LA PIELE IN PUNCTUL DE INCIDENTA, ALES ASA FEL INCIT C-ASUPRA (SAU IN DREAPTA) RAZEI CENTRALE DOZA SE AMPLIFICA CU $EXP(-M_1U+L*TANG(ALFA))$

PSI=UNGHIIUL INTRE OX+ SI SUPRAFATA PJELII (ORICARE DINTRE ELE)

TOATE ACESTE MARIMI SE INTRODUC PE O CARTELA DE FIECARE SURSA, IN ORDINEA:

DEP, ALAT, ADC, RC, FI, TETA, ALFA, PSI CU FORMAT(8F4,1)

OBS. UNGHIIURILE SE DAU IN GRADE S-XACESIMALE.

DIMENSION DOZA(4), IABSC(2), DATE(32), RIND(8), SNA(4), CSA(4), SNT(1-4), CST(4), SNF(4), CSF(4), SNP(4), CSP(4), CT(4), FT(4), CP(4), FP(4), CFT(2-1), CFT2(4), CSTP(4), SNTP(4), SCT(4), XI(4), YI(4), APLDG(4)
DIMENSION PI(10), L(5), MI(5), VMATR(5,6), AM(5,5), EL(5,5), ELA(6), CAPK(1-14), SURSE(8), ARM'C(4)


```

E). DACA          COMMON VDIST(27),VARIE(5),TARM(27,5)
N IF(I=5),        DATA DMCG/'DMCG'/
                  C=.017453
                  111 READ 120,NEOMG
LEMONTE.          120 FORMAT(A4)
DAR NPELPS        IF(NEOMG.EQ.DMCG) GO TO 100
                  VDIST(1)=0.
                  VDIST(2)=.5
E LINIT-          400 VDIST(1)=1-2
INTRAPE          DO 400 I=3,22
ASI CA A          VDIST(23)=22.
                  C S-A INTRODUS SI ADINCIMEA ZERO
                  VDIST(24)=24.
                  VDIST(25)=26.
                  VDIST(27)=30.
                  C VARIE, INCEPTIUNEA CU B/E(SI ADINCIMEA ZERO)
                  VARIE(1)=64.
                  VARIE(4)=225.
                  VARIE(5)=400.
PLANE PT.        READ 401,TARM
ARALELF.          401 FORMAT(10F5,3)
                  100 READ 1,NSURSE,NREL,MCJ,VMX,VMN
                  VARIE(2)=100.
                  VARIE(3)=144.
                  VDIST(26)=28.
ALE CU DR-        1 FORMAT(2I2,2F4,1)
                  N=NSURSE
                  PRINT 1008,N,NREL,MCJ,VMX,VMN
INCIDENTA,        1008 FORMAT(2I5,2F10,2//)
AMPLIFICA        IF(NEOMG.EQ.DMCG) GO TO 500
                  READ 1000,(CAPRT(I),I=1,N)
LINEA:            1000 FORMAT(10F5,2)
                  PRINT 1007,(CAPRT(I),I=1,N)
                  1007 FORMAT(5X,'ORDONATELE:           ',4F7,2)
                  PRINT 500
(                500 FORMAT(//'           ELIPSELE'//)
FT              C LACULUL COEF. TUTUROR ELIPSELOR EL(5,5)
PR              DO 1005 NR=1,NREL
                  READ 1000,PI
                  PRINT 1009,PI

```

```

1001 F=PI*(10X+10F7.2/)
C     CALCUL VMATR. 5/6.PTR. 0 ELIPSA
      DO 1002 I=1.5
        X=PI(2*I-1)
        Y=PI(2*I)
        VMATR(I,1)=X*Y
        VMATR(I,2)=X*Y
        VMATR(I,3)=Y*Y
        VMATR(I,4)=X
        VMATR(I,5)=Y
1002 VMATR(I,6)=1.
C     CALCUL COEFFICIENTS OF ELIPSE
      DO 1004 I=1.5
        DO 1003 J=1.5
          JMATR=J
          DO 1005 I=1.5
            EL(I,J,K)=JMATR=J+I
C     ELEMENTS OF MATRIX K OF ELIPSE 75
1003 AX(I,J)=VMATR(I,JMATR)
          CALL SINV(AM,5,D,L,M)
1004 EL(I,K)=(-1)**(N+I)**K
1005 FL(II,K)=EL(I,K)*X**2/E-10**3
      DO 1 INDC=1
        READ(1,67) P,SY,T,SY
      67 FORMAT(2F9.2)
        WRITE(3,2)
      2 FORMAT(///50X,'COEFFICIENTS')
        WRITE(3,3)
C     PASK PULTE FI.5 SAU 25. PASY DEATE FI.001E(VEGATIV)
      DO 1005 I=1.5
      3 FORNATI(//50X,'EIMPUL DEF ALBT ACC RC FI TET/ ALF
        WRITE(4,2)
      2 FORMAT(12X,'I',1.4X,'O',1.4X,'P',1.4X,'X',1.4X,'Y',1.4X,'M',1.4X,'E',1.4X,'A',1.4X,'R',1.4X,'E',1.4X,'A',1.4X,'S',1.4X,'I',1.4X,'O',1.4X,'N')
14      PSI = ' ')
1400 5.5X,'GRAD',F.Y,'COS',1.7)
      DO 4 I=1,N
        READ(1,5) (RIND(J),J=1.5)
      5 FORMAT(8F9.1)
      DO 5 M=1,3
        K=(I-1)*2+M

```

```
6 DATE(K)=RIND(*)
  WRITE(3,7) 1,(FIND(J),J=1,8)
7 FORMAT(52X,I2,F6.1,F6.1,F5.1,F6.2,F6.1,F5.1,F5.1,F11.1)
  SNA(I)=SIN(RIND(7)*C)
  CSA(I)=COS(RIND(7)*C)
  SNE(I)=SIN(RIND(6)*C)
  CSE(I)=COS(RIND(6)*C)
  SNE(I)=SIN(RIND(5)*C)
  CSE(I)=COS(RIND(5)*C)
  SNE(I)=SIN(RIND(4)*C)
  CSE(I)=COS(RIND(4)*C)
  CT1=SIN((RIND(5)-RIND(6))*C)
  CT(I)=RIND(4)*CT1
  CT2=SIN((RIND(5)-RIND(3))*C)
  CP(I)=RIND(4)*CT2
  IF(SNT(I)+0.01) 101,102,102
101 FT(I)=-1.
  GO TO 104
102 IF(RIND(6)-0.01) 101,103,103
103 FT(I)=1.
104 IF(CT2) 105,42,106
105 FP(I)=-1.
  GO TO 107
106 FP(I)=1.
107 CFT1(I)=-RIND(4)*SNT(I)*CT1
  CFT2(I)=RIND(4)*CST(I)*CT1
  CSTP(I)=CST(I)**2
  SNTP(I)=SNT(I)**2
  SCT(I)=SNT(I)*CST(I)
  XI(I)=RIND(4)*CSE(I)
  YI(I)=RIND(4)*SNE(I)
  A=RIND(2)*RIND(3)
  ARMIC(I)=A
  SURSE(2*I-1)=XI(I)-RIND(1)*CST(I)
  SURSE(2*I)=YI(I)-RIND(1)*SNT(I)
  P=2.*(RIND(2)+RIND(3))
  IF(A/P-2.162) 8,9,9
8 CONST=0.
  AP=A/P
  GO TO 10
```

7 11 11

5 11 11 11

```
9  C0NST=0.
  AP=A/P/3.142
10 T=(AP-1.)/(AP+1.)
  JPLUS(I)=0.66304*T+0.76415*T**3+CONST
  CONTINUE
  KL=1.+36.01*PASK
  IABSC(I)=I-KL
  DO 15 I=2,KL
15  IABSC(I)=IABSC(I-1)+1
  IF(PASK-0.49) 81,82,83
81  WRITE(3,82) (IABSC(I),I=1,KL)
83  FORMAT(//7X,13.9112/)
  GO TO 23
82  WRITE(3,1.) (IABSC(I),I=1,KL)
16  FORMAT(//7X,13.1616/)
  GO TO 23
18  IABSC(I)=0
  DO 19 I=2,KL
19  IABSC(I)=IABSC(I-1)+1
  IF(PASK-0.49) 84,85,85
84  WRITE(3,85) (IABSC(I),I=1,KL)
85  FORMAT(////////IX,13.9112/)
  GO TO 23
85  WRITE(3,17) (IABSC(I),I=1,KL)
17  FORMAT(////////IX,13.1616/)
23  YP=YPA
24  XP=IABSC(I)
C 30  CALL M1, FUNCTELE YP=CONSTANT
  DO 30 I=1,37
  D174=0.
C 31  HUGIA D1PA N4, SLPSELP
  DO 31 I=1,N
  C1=XP*CSTP(I)+YP*SCT(I)+CF1(I)
  C2=XP*SCT(I)+YP*SNTP(I)+CF2(I)
  ELP=(XP*SNT(I)-YP*(SP(I)+CF(I)))*F(I)
  IF(ELP-0.05) 81,82,82
88  IF((C1*SP(I)-C2*(SP(I)+CF(I)))*F(I)-0.05) 81,26,26
26  ER=C*(XP*SNT(I)-YP*CST(I)+CF1(I))*F(I)
  K=(I-1)*P+1
  A0=SURT((X1(I)-C1)**2+(Y1(I)-C2)**2)
```

```
IF(LD=20) 61,61,62
61 LOGZ4(TL)=0
GO TO 43
62 Z=SQRT((XP-C1)**2+(YP-C2)**2)
DEP=U*TR(K)
ALMTC=Z*CFP/(CFP+AD)
X=(-0.15/CFP-0.27)*AD+(0.0123*AD-C.0004)*APL6(1)+2.0457
IF(1.-X) 24,27,27
27 FACTOR=1.
28 ANLOG=1.+1.14*Z2+X+0.477-0.23*X*X+0.20003*X**3+0.1200080*X**4
ANLOG=FACTOR*ANLOG**2
GO TO 32
29 IF(X-2.) 31,33,30
30 X=X-2.
FACTOR=100.
GO TO 29
31 X=X-1.
FACTOR=20.
GO TO 29
32 ALAT=CATE(X)+0.5
IF(AL-ALAT) 25,33,33
33 S=(ALMTC/ALAT-1.)/C.24041
FACTOR=-0.5
34 RS=(1-(0.070100+S+C.000972)*S+C.230309)*S+C.275303)*S+1.
RS=1.-1./RS**34
GO TO 36
35 S=(1.-ALMTC/ALAT)/C.24041
FACTOR=0.5
GO TO 34
36 PK=0.5+FACTOR*RS
COEF=X*(0.064)*ALMTC*SNA(1)/CSA(1)
IF(EAC) 71,72,72
71 RFZ=PK*ANLOG*COEF
GO TO 73
72 RFZ=PK*ANLOG/COEF
73 IF(RFZ-100.) 45,49,43
46 RFZ=100.
49 CONTINUE
IF(INEOMG,EC,EMFC) GO TO 37
C CALCULATE AREA TO RIVFLU P
```

```
      CD=CAPRT(I)
      XS=SURSF(2*I-1)
      YS=SURSF(2*I)
C  ****
C  SURSA CU LINTA PTELII NEPARALELA LA CA PE LOCUL I
C  CIND NU EXISTA ACEASTA SURSA SE VA SCHIMBA 1 IN 5
C  IF(I-5) 301,302,301
C  ****
      IF(I-1) 301,302,301
301  XPP=YP-YS
      XPP=(XP-XS)*(CD-YP)/XPP+XS
      YPP=CD
      GO TO 303
302  XPP=(CD-YP)*(XP-XS)+XS*(YP-YS)/(YP-YS+XS-XP)
      YPP=XPP+CD
303  PS=SQRT((XP-XS)**2+(YP-YS)**2)
      ARIE=PS/SQRT((XPP-XS)**2+(YPP-YS)**2)*ARIE(I)
C  BUCLA DUPA NR. ELIPSELOR
      DO 2000 NE=1,NREL
C  INTERSECTAREA LINIEI PS CU ELIPSA
      J=NE
      IF(ABS(XP-XS).GT..01) GO TO 1016
      X1=XS
      X2=XS
      E1=EL(J,1)*XS+EL(J,3)
      E=(E1**2-4.*EL(J,2)*(EL(J,3)*XS+XS**2+EL(J,5)))
      IF(E) 2000,2000,1013
1013  Y1=(E1+SQRT(E))/(-EL(J,2))*0.5
      Y2=(E1-SQRT(E))/(-EL(J,2))*0.5
      GO TO 310
1016  AL=(YP-YS)/(XP-XS)
      BE=YP-XP*AL
      AR=1.+(EL(J,1)+EL(J,2)*AL)*AL
      B=(BE*(EL(J,1)+2.*AL+EL(J,2))+EL(J,3)+EL(J,4))*AL
      CE=(EL(J,2)*BE+EL(J,4))*BE+EL(J,5)
      DT=B**2-4.*AR*CE
      IF(DT) 2000,2000,1015
1015  RAD=SQRT(DT)
      X1=(RAD-B)/AR*0.5
      X2=(-(RAD+B)/AR*0.5
```

```
Y1=AL*X1+BE
Y2=AL*X2+BF
310 SP1=SQRT((XS-X1)**2+(YS-Y1)**2)
    SP2=SQRT((XS-X2)**2+(YS-Y2)**2)
    DISTMN=AMIN1(SP1,SP2)
    TESTP=XP*XP+XP*YP+EL(J,1)+YP*YP*EI(J,2)+XP*EI(J,3)+YP*EI(J,4)+EL(J
1,5)
    IF(TESTP) 340,330,320
C   P INTERIOR
340 DIST=PS-DISTMN
341 F=1./TAR(ARIE,DIST)**.65
    GO TO 350
C   P PE FLIPSA
330 IF(PS-DISTMN) 2000,2000,331
331 DIST=SQRT((X1-X2)**2+(Y1-Y2)**2)
    GO TO 341
C   P EXTERIOR
320 IF(PS-DISTMN) 2000,2000,321
321 DIST1=PS-DISTMN
    DIST2=PS-AMAX1(SP1,SP2)
    F=(TAR(ARIE,DIST2)/TAR(ARIE,DIST1))**.65
350 REZ=F*REZ
2000 CONTINUE
37 DOZA=REZ+DOZA
    IDOZA(I)=DOZA
40 XP=XP+PASX
    IF(INDC) 50,51,51
50 WRITE(3,44) (IDOZA(I),I=1,37),YP
54 FORMAT(1X,37I3,F6.2)
    GO TO 39
51 WRITE(3,52) YP,(IDOZA(I),I=1,37)
52 FORMAT(1X,F6.2,37I2)
59 YP=YP+PASY
    IF(YP-YMN) 41,24,24
41 INDC=-INDC
    IF(INDC) 18,42,42
52 IF(1-MCJ) 43,43,111
43 STOP
    END
C
```

C

```

FUNCTION TAR(A,D)
COMMON VDIST(27),VARIE(5),TARM(27,5)
DO 1 I=1,40
  IF(VDIST(I)-D) 1,2,3
2 ID=I
  GO TO 3
1 CONTINUE
3 DO 4 J=1,7
  IF(VARIE(J)-A) 4,5,5
5 IA=J
  GO TO 6
4 CONTINUE
6 EXPR=(A-VARIE(IA-1))/(VARIE(IA)-VARIE(IA-1))
  TARI=TARM(ID-1,IA-1)+(TARM(ID-1,IA)-TARM(ID-1,IA-1))*EXPR
  TAR2=TARM(ID,IA-1)+(TARM(ID,IA)-TARM(ID,IA-1))*EXPR
  TAR=TARI+(TAR2-TARI)*(D-VDIST(ID-1))/(VDIST(ID)-VDIST(ID-1))
RETURN
END

```

EXAMPLE (CATTLE PL. CONTROL)

NO. PUGEN

10	1.0291	017	581	555	521	384	311	272	260
	722	551	545	610	577	544	515	485	450
	403	365	344	303	268	235	210	1.0361	029
1.002	971	936	772	864	825	785	746	707	671
	635	602	570	539	511	484	459	432	409
	324	287	256	223	1.0431	0371	014	983	952
	918	881	844	804	766	728	693	656	624
	564	535	507	480	455	431	407	385	364
	246	1.0521	0451	022	556	566	525	495	462
	826	789	755	719	686	652	624	593	564
	510	484	460	437	414	396	382	2711	1.061
1.0371	0361	022	962	952	921	887	852	819	787
	754	722	690	659	629	601	576	550	525
	459	433	411	387	365				
05050116	05-01								
-12	01425	-1							
-0.3	2.2	-10.9	9.9	-6.9	14.7	-4.5	13.1	-1.8	9.4
-2.5	13.7	-3.3	15.7	-6.8	15.7	-1.1	14.7	-1.8	13.6


```
2.2 5.4 4.4 6.5 6.2 5.6 8. 2.8 4.9 3.7
7.8 3.0 0.4 13.5 5.3 9.4 8. 14.5 11.7 5.9
5.4 13.4 4. 13.5 3.2 14.5 2.5 15.0 5. 12.7
0050-05
50. 8. 10. 11. 03545 135.0000045.
50. 10. 15. 13.5 90.270.
050.010.015.+1.0270.050.
```

CITIREA MATRICILOR RARE (SPARSE).

```
SUBROUTINE SPARSE(A,M)
C CITIRE MATR. RARE, CU FORMAT(2I5,F15.7), PT. I, J SI A(I,J)
C OPRESTE EXECUTIA PT. CEL PUTIN UN INDICE GRESIT
C UN ELEMENT ARE VALOAREA ULTIMA, CINE SE DAU MAI MULTE
C A ESTE DATA CU INDICIII INTRE 1 SI M
  DIMENSION A(1)
  IND=1
  K=M*M
  DO 1 I=1,K
  1 A(I)=0.
  DO 5 I=1,K
  READ(1,2,END=6) I,J,P
  2 FORMAT(2I5,E15.7)
  IF(I*(M+1-I).LE.0) GO TO 3
  IF(J*(M+1-J).LE.0) GO TO 3
  A((J-1)*M+I)=P
  GO TO 5
  3 PRINT 4
  4 FORMAT(5X,'ELEMENT GRESIT:')
  IND=-1
  5 PRINT 2,I,J,P
  6 IF(IND.EQ.-1) STOP
  RETURN
END
```

SIMETRIZAREA SUBMATRICII PATRATICE MAXIMA CU CONTROL IN A(K,L).

```
SUBROUTINE SMTRY(A,M,K,L)
C A ESTE MATRICE PATRATICA DE TIP M/M
C INLOCUIESTE A(I,J), A(2K-I,2L-J) CU MATEA LCA
  DIMENSION A(M,M)
```

```
J2=M-L
IF(J2+1.GT.L) J2=L-1
I2=M-K
IF(I2+1.GT.K) I2=K-1
JLMT=L-J2
IF(J2.GT.I2) JLMT=L-I2
I1=K-I2
I2=K+I2
J2=L+J2
K2=K*2
L2=L*2
KL=K-L
DO 2 J=JLMT,J2
  ILMT=J+KL
  IF(ILMT.GT.I2) ILMT=I2
  DO 1 I=I1,ILMT
    A(I,J)=(A(I,J)+A(K2-I,L2-J))*0.5
  1 A(K2-I,L2-J)=A(I,J)
  2 CONTINUE
  RETURN
END
```

SISTEME ALGEBRICE LINIARE COMPLEXE.

SUBROUTINE CGELG(R,A,M,N,EPS,IERR)

C CGELG FOLOSESTE ROUTINA DE REZOLVARE A SISTEMELOR LINIARE CU COEFICIENTI
C REALI GELG DIN SCIENTIFIC SUBROUTINE PACKAGE(SSP)-IBM.
C PARAMETRII M,N,EPS SI IERR CA IN GELG DIN SSP
C K(M,N) SINT TERMENII LIBERI (COMPLEXI). SOLUTIA SE OBTINE IN R.
C A(M*2M) IN DIMENSION DIN MAINPGM.A(M,M)= MATRICEA
C SISTEMULUI,COMPLEXA,IN PRIMELE M COLONAE.

DIMENSION R(1),A(1)

M2=M*2

M22=M*M2

M1=M-1

DO 1 I=0,M1

M2I=M2*I

M2I1=M2I-1

KI=M22+M2I

KR=KI-1

DO 1 J=2,M2,2

```
A(KR+J)=-A(M2I+J)
1 A(KI+J)=A(M2I+J)
CALL GELG(R,A,M2,N,EPS,IERR)
M1=N-1
DO 1 I=0,M1
M2I=M2*I
M2I1=M2I+M
DO 2 J=1,M
J2=2*J
A(J2-1)=R(M2I+J)
2 A(J2)=R(M2I1+J)
DO 3 J=1,M2
3 R(M2I+J)=A(J)
RETURN
END
```

RUTINE ASSEMBLER

AVANTAJELE PRINCIPALE ALE PROGRAMELOR SCRISE IN (LIMBAJUL) ASSEMBLER, FATA DE CELE DIN (LIMBAJUL) FORTRAN SINT: TAMP DE CALCUL MAI MIC SI MEMORIE INTERNA MAI MICA. DEZAVANTAJELE: DIFICULTATEA SCRIERII DE PROGRAME ASSEMBLER SI DIFERENTIA ACESTUI LIMBAJ FATA DE LIMBAJELE ANALOAGE DE LA ALTE CALCULATOARE (FELIX, CDC ETC).

RECOMANDARE.

ESTE BINE CA PROGRAMELE CARE FOLOSESC RUTINELE DE MAI JOS, SA RESPECTE CONVENTIILE (IBM) DE FOLOSIRE A REGISTRELOR (GENERALE) DE LEGARE:

REGISTRUL 13 CONTINE ADRESA ARIEI (72 BYTES) DE SALVARE A PROGRAMULUI CARE CHEAMA; REG. 14 CONTINE ADRESA (LOCUL) INTOARCIERII IN PROGRAMUL CARE CHEAMA; REG. 15 CONTINE ADRESA DE INCEPERE A EXECUTIEI PROGRAMULUI CHEMAT; REG. 0 INTOARCE DIN PROGRAMUL CHEMAT REZULTATUL (VALOAREA) FUNCTIEI (DACA ESTE UN SUBPROGRAM FUNCTIE); REG. 1 VA TRANSMITE PROGRAMULUI CHEMAT ADRESA LISTEI DE PARAMETRI. DACA PROGRAMUL CHEMAT ESTE O FUNCTIE DE TIP REAL, VALOAREA ACESTUIA VA FI PUSA IN REG. DE VIRGULA MOBILA 0.

UNELE DIN RUTINELE PREZENTATE AICI ALTEREAZA UNELE DIN REGISTRELE DE LEGARE SI UNELE DE VIRGULA MOBILA. DEASEMENA, NU S-A PREVAZUT POSIBILITATEA MESAJELOR CU TRACEBACK.

OBS. PENTRU PEDANTERIE, IN RUTINELE ASSEMBLER SE POATE INTRODUCI INSTR. LA 15.0(15), INAINTEA INSTR. USING *,15.

BIBLIOGRAFIE

1. IBM SYSTEM/370 PRINCIPLES OF OPERATION
2. DOS (AND TOS) ASSEMBLER LANGUAGE
3. DOS SUPERVISOR AND I/O MACROS
4. INTRODUCTION TO IBM DIRECT-ACCESS STORAGE DEVICES AND ORGANIZATION METHODS
5. IBM SYSTEM/360 AND SYSTEM/370 FORTRAN IV LANGUAGE
6. IBM SYSTEM/370 DISK OPERATING SYSTEM
7. DOS SYSTEM CONTROL AND SERVICE
8. DOS FORTRAN IV PROGRAMMER'S GUIDE

SUPRIMAREA INTERRUPTIILOR DE EXPONENT-UNDERFLOW.

```
OUTUND CSECT  USAGE: CALL OUTUND
      BALR 15.0
*      LA 15.0(15)
      USING *.15
      BALR 0.0
      X 0,MAS
      SPM 0
      BR 14
      DS OF
MAS    DC X'02000000'
      END
```

FACE INTERRUPTIILE DE EXPONENT-UNDERFLOW

```
INUND  CSECT  USAGE: CALL INUND
      BALR 15.0
      USING *.15
      BALR 0.0
      D 0,MASCA
      SPM 0
      BR 14
      DS OF
MASCA  DC X'02000000'
      END
```

INTERRUPTIILE DE ABNORMAL-END

```
SETAR  CSECT  USAGE: CALL SETAR
* SETAR DA (SUPERVIZORULUI) PE ABEND CA RUTINA ABNORMAL-END
* RUTINA ABENA SA FIE INCHESIATA CU STOP (ECU IN ASSEMBLER).
      SAVE (14,12)
```

```
BALR 10.0
USING *,10
EXTRN SAVERR, ABEND
STXIT AB, ABEND, SAVERR
RETURN (14, 12)
END
```

INTERRUPTURILE DE PROGRAMM-CHECK

```
CALL STXIT
C STXIT DA (SUPERVIZORULUI) PE PCRUȚ CA RUTINA PROGRAMM-CHECK
  A=2.E+50**2
  A=5.E-50**2
  A=J.
  A=1./A
  STOP
  END
```

ICTL 1.79.2.

```
STXIT CSECT
  SAVE (14, 12)
  BALR 10.0
  USING *,10
  EXTRN SAVERR
  STXIT PC, PCRUȚ, SAVERR
  RETURN (14, 12)
* RUTINA PCRUȚ SUPRIMA EXPONENT-UNDERFLOW SI LASA CELELALTE INTERRUPTURILE
* PERI IN SEAMA RUTINEI ILFFINT.
PCRUȚ BALR 3.0
  USING *,3
* REFERIREA UNEI MARIMI (SAVERR) DIN ALTA RUTINA CERE URMATORILE 2
* INSTRUCȚIUNI (SI EXTRN SAVERR)
  L 4.=AISAVERR)
  USING SAVERR,4
  MVN COD,SAVERR+3
  CLI COD,13
  BNE ALTPC
  EXIT PC
ALTPC L 15.=V(ILFFINT)
  B 0(15)
  COD DC X'00'
```

END

DATA RULARII UNLI PROGRAM

```
DATE      CSECT  USAGE: CALL DATE(I,J,K,D)
*  D ESTE D ARIE DE E BYTES
*  I,J,K CONTIN DATA DE INTREGI,C CONTINE DATA IN CARACTERE
      BALR 15,0
      USING *,15
      STM 0,3,REG
      LR 2,1
      CCMRG
      MVC 0,0(1)
      L 3,12(2)
      MVC 0(P,3),D
      XI D,C
      PACK DD,D(2)
      CVB 0,DD
      L 3,0(2)
      ST 0,0(3)
      PACK DD,D+3(2)
      CVB 0,DD
      L 3,4(2)
      ST 0,0(3)
      PACK DD,D+6(2)
      CVB 0,DD
      L 3,8(2)
      ST 0,0(3)
      LM 0,3,REG
      BR 14
REG      DS 4F
J        DS D
DD       DS D
C        DC X'0030610030610030'
      END
```

SUPRIMAREA ASIGNAREA TEMPORARA A UNITATII LOGICE SYSOON

```
RESET    CSECT  USAGE: CALL RESET('SYSOON')
      BALR 15,0
      USING *,15
```

```
STM 0.2.REG
L 1.0(1)
LA 2,R+R
MVC 0(3.2),3(1)
R   RELEASE (SYS000)
LM 0.2.REG
BR 14
REG DS 3F
END
```

ULTIMUL BYTE OCUPAT DE PROGRAM (MICCARE DIN HARTA DE LINCARE)

```
PGMEND CSECT  USAGE: CALL PGMEND(1)
      BALR 15.0
      USING *.15
      STM 0.3.REG
      LR 2.1
      CGMRG
      L 3.0(2)
      MVC 0(4.3),40(1)
      LM 0.3.REG
      BR 14
REG DS 4F
END
```

INCARCA SI EXECUTA FAZA XXXXXXXX (ULTIMII X POT FI BLANCURI)

```
FETCH CSECT  USAGE: CALL FETCH('XXXXXXXXX')
* FAZA XXXXXXXX ESTE IN CORE IMAGE LIBRARY (CLB)
      L 1.0(1)
      FETCH (1)
      END
```

SECUNDA EXECUTIEI LUI GETIME

```
GETIME CSECT  USAGE: CALLGETIME(1)
* I V-A INTOARCE SECUNDA EXECUTIEI LUI GETIME
      BALR 15.0
      USING *.15
      STM 0.2.REG
      L 2.0(1)
      GETIME BINARY
```

```
ST 1.0(2)
LM 0.2,REG,
BR 14
REG DS 3F
END
```

DUCE CARTELE DEPE CARD-FEEDER PE BANDA (SYS004), PANA LA /

```
CARUTP CSECT
      BALR 15,0
      USING *,15
      LA 2,RCCB
RC     EXCP RCCB
      WAIT RCCB
      TM 4(2),1
      BU EQU
      EXCP TCCB
      WAIT TCCB
      B RC
      EQU
      EQU 0,4
RCCB   CCW SYSDPT,RCCW
RCCW   CCW 2,IOAREA,32,80
TCCB   CCB SYS004,TCCW
TCCW   CCW 1,IOAREA,X'20',80
      DS 80C
      END
```

FIȘIER PE BENZI MAGNETICE.

SE POT SCRIE PROGRAME FORTRAN PT. FIȘIERE PE B.M. DE CUPOL TIE (IMPL-
MISTRARI RECUNOSCUTE MULTI-FILE-VOLUME SAU MULTI-VOLUME-FILE). PENTRU
ACEȘI SCOP EXISTA TREI FACILITĂȚI:

1. UNICE OPERATIE POSIBILA PRIN INSTR. // MTC ... LA NIVEL DE JOB-
CONTROL. SE POATE FACE SI IN FORTRAN CU INSTRUCȚIUNEA:

CALL FSR/FSE/BSR/PSE/WTM/ERG/REW/RUN(IOSRN)

PARAMETRUL IOSRN (DATA DT REFERENCE NUMBER) POATE FI NR. INTREG
SAU VARIABILA INTREGA DE LUNGIME 4. DECI SE FACE ASIGNAREA:

// ASSCN SYSNNN.X'23M'

unde NNN=IOSRN-3 SI M=0 SAU 1

OBS. IN OPERATIA FSR/PSR, INREGISTRAREA END-OF-FILE (TAPL-MARK SAU TM) CONTEAZA CA INREGISTRARE DE DEPASIT/DEWMSAT. OPERATIA FSE/MSF PIZITIONEFAZA BANCA DUPA PRIMUL TM CARE URMEAZA/PRECEDE (FAZA DE LOCUL IN CARE SE AFLA INAINTE DE OPERATIE).

OBS. CU OPERATIA RUN SE PLATE PEELORA (IN ACEASI EXECUTIE) UN FISIER MULTIVOLUM. SE VA CERE OPERATORULLI (INAINTEA LUI CALL RUN(ESRN)) BANCA DE MONTAT. ACEST LUCRU SE POATE FACE CU SECVNTA:

WRITE(15,1)

I FORMATIX. LA FORMATAREA DESCARCARE A Uu. 200/281 MONTATI BAN-
I CA X)

II. SE POATE CITI ORICE INREGISTRARE (MAI MICA DE 32K BYTES) CU :
CALL RDTPEY(A,N,ICSRN)

A = ARIA DIN MEMORIE IN CARE SE ADUCE INREGISTRAREA

N = NR. DE BYTES AI INREG. (INTORS DE RDTPEY CA REZULTAT). CIND SE
CITESTE UN TM(END-OF-FILE), N ESTE NEGATIV.

ICSRN = ACEASI SEMNIFICATIE CA MAI SUS

CITINEA CU RDTPEY NU FACE IN PLUS NICI-O ALTA MISCARE (CITESTE IN-
REGISTRAREA URMATOARE SI LASA BANCA IMEDIAT DUPA EA).

III. SE POATE SCRIE ORICE INREGISTRARE (MAI MICA DE 32K BYTES) CU:
CALL WRTPHY(A,N,ICSRN)

A = ARIA DIN MEMORIE DIN CARE SE IAU INFORMATIILE DE SCOS PE P.M.

N = NR. DE BYTES AI INREGISTRARII (PRIMI A DIN A)

ICSRN = ACEASI SEMNIFICATIE CA MAI SUS

OBS. TM SE POATE SCRIE CU OPERATIA WTM DE LA I

OBS. REAMINTIM CA PRIMA CITIRE/SCRIERE CU INSTR. FORTRAN ASUPRA U-
NUI FISIER PE BANCA, VA REBUBINA INAINTE DE A CITI/SCRIE. FOLCIND
ACEASI UNITATE LOGICA (ACELASI ICSRN) PT. TOATE FISIERELE DEPE O BAN-
CA, SE EVITA REBUBINAREA LA PRIMA OPERATIE PE UN NOU FISIER.

<u>RUN</u>	CSECT	USAGE:CALL PUN(ICSRN), (ICSRN=INTEGER*4
*	BALR	15,0 EACA NU SE RESPECTA REGULA REG. DE LEGARE
	USTIG	*.15
	L	1,0(0,1)
	L	1,0(0,1)
	S	1,=F*3'
	STC	1,FCCB*7
	EXCP	FCCB
	NR	14
	LTOPG	
FCCB	CCH	SYS000,FCCW

FCCW CCM 15.1.32.1

FSR CSECT USAGE: CALL FSR(DSRN),DSRN=INTEGER*4
USING *0,15
L 1.0(0.1)
L 1.0(1.1)
S 1.=F'3'
STC 1.FCCB1*5
EXCP FCCB1
R# 14
LTORG
FCCB1 CCB SYS000.FCCW1
FCCW1 CCM 55.1.32.1

WTM CSECT USAGE: CALL WTM(DSRN),DSRN=INTEGER*4
USING *0,15
L 1.0(0.1)
L 1.0(1.1)
S 1.=F'3'
STC 1.FCCB2*7
EXCP FCCB2
R# 14
LTORG
FCCB2 CCB SYS000.FCCW2
FCCW2 CCM 31.1.32.1

FSF CSECT USAGE: CALL FSF(DSRN),DSRN=INTEGER*4
USING *0,15
L 1.0(2.1)
L 1.0(1.1)
S 1.=F'3'
STC 1.FCCB3*7
EXCP FCCB3
R# 14
LTORG
FCCB3 CCB SYS000.FCCW3
FCCW3 CCM 43.1.32.1

BSF CSECT USAGE: CALL BSF(DSRN),DSRN=INTEGER*4
USING *0,15

```
      I      1,0(10,1)
      L      1,0(10,1)
      S      1,=F'3'
      STL    1,FCCP4+7
      EXCP   FCCP4
      BR     14
      LTORG
FCCM4    LCB   SYSJ.C.FCCM4
FCCM4    CCM   47.1.32.1
```

```
MSR    CSECT  USAGE=(CALL RSM(DSRN),CSRN=INTEGER*)
        USING  *0,15
        I      1,0(10,1)
        L      1,0(10,1)
        S      1,=F'3'
        STL    1,FCCP5+7
        EXCP   FCCP5
        BR     14
        LTORG
FCCM5    LCB   SYSJ.C.FCCM5
FCCM5    CCM   39.1.32.1
```

```
MSR    CSECT  USAGE=(CALL RSM(DSRN),CSRN=INTEGER*)
*      MSR POZITILNEAZA IN FATA TM-ULUI DIN CAPUL COMZII,FARA A FACE SI-
*      MIL ALTCEVA,INCHERENT IN CE MOMENT E CAT. COMANDA
        USING  *0,15
        I      1,0(10,1)
        L      1,0(10,1)
        S      1,=F'3'
        STL    1,FCCP4+7
        EXCP   FCCP5
        BR     14
        LTORG
FCCM6    LCB   SYSJ.C.FCCM6
FCCM6    CCM   7.1.32.1
```

```
MSR    CSECT  USAGE=(CALL CRG(DSRN),CSRN=INTEGER*)
        USING  *0,15
        I      1,0(10,1)
        L      1,0(10,1)
```

```
S      1.=F'3'  
STC   1.FCCB7+7  
EXCP  FCCB7  
BR    14  
LTORG  
FCCB7 CCB  SYS000.FCCW7  
FCCW7 CCW  23.1.32.1  
END
```

WRTPBY CSECT USAGE: CALL WRTPBY(A,N,DSRN)

- * SCRIE DIN MEMORIA PRINCIPALA PE BANDA MAGNETICA (SYSDSRN,DSRN='-L)
- * A =ARIE IN MEMORIA PRINCIPALA, N= NR.BYTES DIN A DE SCOS, DSRN=DATA
- * SET REF. NUMBER (FORTRAN)

```
BALR 15.0  
USING *,15  
MVC WRCCW+1(3),1(1) ADDR OF A: CCW  
L 0.8(0.1) ADDR OF DSRN : PO  
L 1.4(0.1)  
MVC WRCCW+6(2),2(1) N : CCW  
LR 1.0  
L 1.0(0.1)  
S 1,TREI  
STC 1,WRCCW+7  
LA 1,WRCCW  
EXCP (1)  
WAIT (1)  
BR 14  
CNOP 0.4  
WRCCW CCE SYS000,WRCCW  
TREI DC F'3'  
WRCCW CCW 1.1.0.1  
END
```

RDTPHY CSECT USAGE: CALL RDTPHY(A,N,DSRN)

- * LITESTE DEPE BANDA MAGNETICA (SYSDSRN) IN MEMORIE
- * READ THE FOLLOWING PHYSICAL RECCRD ON TAPE
- * A = RECCRD ADDRESS IN MAIN STORAGE
- * N=NUMBER OF BYTES IN RECCRD (0,LT. 32K),OUTPUT PARAMETER
- * N= NEGATIVE FOR A TAPE MARK
- * DSRN=DATA SET REFERENCE NUMBER(FORTRAN,INTEGER*4)

```
BALR 15.0
USING *.15
MVC RDCCW+1(3),1(1)
L 0,4(0.1)
L 1,8(0.1)
L 1,0(0.1)
S 1,TREI
STC 1,RDCCB+7
LA 1,RDCCB
EXCP (1)
WAIT (1)
LR 1,C
LH C,RDCCB
LCR 0,0
TM RDCCB+4,1
BO RET
AH 0,RDCCW+6
RET ST 0,C(0.1)
BR 14
CNDP 0,4
RDCCB CCH SYS000,RDCCW
TREI DC F'3'
RDCCW CCM 2,1,32,32767
END
```

FISIERE (MONO SI MULTI-EXTENT) PE DISCURI MAGNETICE.

OBS. RUTINELE PREZENTATE MAI JOS PERMITS LUCRUL CU FISIERE PE DISC. FARA UTILIZAREA PROGRAMELOR DE PEFORMARE, PRELUCRARI DE ORICE FEL. (SERIE, ACCES DIRECT, CREARE DE FISIER COMBINATA CU PUNERE LA ZI ETC.) SI IN TIMP DE RULARE MAI MIC. IN SCHIMB, SCRIEREA PROGRAMELOR DE LUCRU (CHIAR SI IN FORTRAN) ESTE MAI GREU.

NEWFILE CSFCT. USAGE: CALL NEWFILE

* NEWFILE CREAZA FISIERUL (FILE ID-UL, EXTENT-URI SI RETENTIE NOI)

```
SKCCW BALR 15.0
USING *.15
OPEN DFILE
MVC SKCCW+2(4),SALT
BR 14
SALT DC X'47F0F016'
```

DFILE DTFPH TYPEFILE=OUTPUT,CCWADDR=SKCCW,DEVICE=2314,MCUNTED=ALL
END

WRITED CSECT USAGE: CALL WRITED(A,NB,NC,NP,NR,NTEST)

- * WRITED SCRIE INREGISTRARI NOI PE FISIERUL DFILE.
- * A ESTE ARIA DIN MEMORIE DIN, CARE SE SCOT NH BYTES PE CILINDRUL NC,
- * PISTA NP SI INREGISTRAREA NP. NTEST INTOARCE VALORILE: NR. DE BYTES
- * SCRISI 0 PT. NB NEPOZITIV SI -1 PT. NP GRESIT
- * NC SAU/SI NP GRESIT CANCELAZA JOB-LL

```
BALR 15.0
USING *.15
STM 1.3,REG
L 2.8(0.1)
MVC CC.2(2)   NC:CC
MVC 0.2(2)   NC:C
L 2.12(0.1)
MVC NH.2(2)  NH:HH
MVC H.2(2)  NH:H
L 2.16(0.1)
L 2.0(0.2)
STC 2.RL   NP:RL
RCTR 2.0
STC 2.R   NR-1:R
L 2.4(0.1)
L 3.0(0.2)
LA 0.0
LTR 3.3
BNP RET
S 3.0LMAX
BP RET
A 3.0LMAX
STM 3.DL   NB:DL
L 0.0(0.1)
LA 1.8(0.2)
STM 1,WRDATA+6   2+NB:CCW
LA 2.REC
MVCL 2.0   A:I/O AREA
OPEN DFILE
MVC DFILE+6(2),DFILE+30   SYSOON:CCW
MVC A+2(4),SALT
```

```
LA 1.DFILE
EXCP (1)
WAIT (1)
LM 0.0L
SM 0.DFILE
TM DFILE+3.8 NO RECDR FOUND TEST
BZ RET
SR 0.0
BCTR 0.0
RET L 1,REG
L 2,20(0.1)
ST 0,0(0.2)
LM 2,3,REG+4
BR 14
SALT DC X'47F0F084'
DFILE DTFPH TYPEFLE=INPUT,CCWACDR=SKCCW.DEVICE=2314,MOUNTED=ALL
SKCCW CCM 7,SEEK,64,6
CCM 49,CC,64,5 SEARCH PE ID-LL INREG. PRECEDENTE
LCW 8,*-8,0,0
WRDATA CCM 29,C,0,0 WRITE COUNT,KEY & DATA
LTORG
DLMAX DC F'7800'
SEEK DC H'0'
CC DS H
HH DS H
R DS C
REG DS 3F
C DS H
H DS H
PL DS C
AL DC X'00'
DI DS H
REC DS 7800C
END
```

UPDD CSECT USAGE: CALL UPDD(A,N,AC,AM,AR,NTEST)

- * UPDD ACTUALIZEAZA (UPDATE) INREGISTRARI VECHI IN FISIERUL CFILE.
- * N =NR. DE BYTES DIN A DE SCOS PE DISC IN SA NU INTRFAÇA VECHEA
- * INREGISTRARE. NTEST ARATA CITI BYTES AU FOST SCOSI.
- * CELELAITE INREG. DEPE PISTA SINT LASATE INTACTE

```
* NC=0-199,NH=0-19,NN=1-7C
  BALR 15,0
  USING *,15
  STM 1,2,REG
  MVC WRCCW+1(3),1(1)  ADDR OF A : CCW
  L 2,R(0,1)
  MVC CC,2(2)  NC:CC
  L 2,12(0,1)
  MVC HH,2(2)  NF:HH
  L 2,16(0,1)
  MVC R,3(2)  NR:R
  L 2,4(0,1)
  SR 0,0
  L 1,0(0,2)
  LTR 1,1
  BNP RET
A  MVC WRCCW+6(2),2(2)  N:CCW
  OPEN DFILE
  MVC DFILE &(2),DFILE+30
  MVC A+6(4),SALT
  LA 1,DFILE
  FACP (1)
  WAIT (1)
  SR 0,0
  BCTR 0,0
  TM DFILE+3-8
  BO RET  BOM PE NO RECORD FOUND
  LH 0,WRCCW+6
  SH 0,DFILE
RET L 1,REG
  L 2,20(0,1)
  ST 0,0(0,2)  NR,BYTES SCRISI:NTST
  L 2,REG+4
  RR 14
  DS 2F
DFILE DTFPH TYPEFILE=INPUT,CCWADDR=SKCCW,DEVICE=2314,MOUNTED=ALL
SEEK  DC H'0'
CC  DS H
HH  DS H
R  DS C
```



```
SKCCW  CCW 7. SEEK.64.6  SEEK PE BRCCH  
      CCW 49.CC.64.5  SEARCH PE IC-UL INREG. CITITE  
      CCW 8.*-8.0.C  
WRCCW  CCW 5.C.0 0  WRITE DATA  
      LTORG  
SALT  DC X'47F0F058'  
      END
```

READD CSECT USAGE: CALL READD(A,N,NC,NH,NR,NTEST)

- * READD CITESTE O INREGISTRARE DIN FISIERUL DFILE
- * A= ARIA DIN MEMORIE IN CARE SE ADUCE INREGISTRAREA.
- * N= NR. DE BYTES DIN INREG. PE CARE VREM SA-I CITIM
- * NC=NR. CILINDRULUI CU INREG.(NC=0 PT. PRIMUL)
- * NH=NR. CAPULUI DE ACCES(NH=0 PT. PRIMUL)
- * NR=NR. DE ORDINE AL INREG. PE PISTAI (IN FISIER)

```
BALR 15.0  
USING *.15  
STM 1,2,REG  
L 2,8(0.1)  
MVC CC,2(2)  NC:CC  
L 2,12(0.1)  
MVC HH,2(2)  NH:HH  
L 2,16(0.1)  
MVC R,3(2)  NR:R  
MVC RDCCH+1(3),1(1)  ASEP CF A :CCW  
L 2,4(0.1)  ADDR OF N : 2  
SR 0.0  
L 1,0(0.2)  
LTR 1.1  
BNP RET  
* MVC RDCCH+6(2),2(2)  N:CCW  
OPEN DFILE  
MVC DFILE+6(2),DFILE+30  
MVC A+6(4),SALT  
LA 1,DFILE  
EXCP (1)  
WAIT (1)  
SR 0.0  
RCTR 0.0  
TM DFILE+3.8
```

LL

```
NO RET    PCH FF NO RECORD FOUND
LM 0,RDCCW+6
SH 0,DFILF
RET      L 1,REG
        L 2,20(0,1)
        ST 0,0(0,2)  NR.BYTES CITED:INTEST
        L 2,REG+4
        BR 14
REG      DS 2R
DFILE    DTFPH TYPEFILE=INPUT,CCWADDR=SKCCW,DEVICE=2314,MOUNTED=ALL
SEEK     DC H'0'
CC       DS H
MH       DS H
R        DS C
SKCCW    CCW 7,SEEK,64,6      SEEK UNCLFF
        CCW 49,CC,64,5      SEARCH CCHFR,PE ID-UL INREG, CITITE
        LCW 8,+8,0,0        TRANSFER IN CHANAL
RDCCW    CCW 6,0,0,0  READ DATA
        LTORG
SALT     DC X'47F0F058'
        END
```

GETCPC CSECT USAGE: CALL GETCPC(NC,NH,NR,NBL)

- * GETCPC CITESTE INREGISTRAREA DE CAPACITATE A PISTEI
- * CITESTE NR. DE BYTES LIBERI (NBL) PE CILINDRUL NC, PISTA NH SI NR. DE
- * INREG. EXISTENTE PE PISTA (NR)

```
BALR 15,C
USING *.15
STM 1,2,REG
L 2,0(0,1)
MVC C,2(2)  NC=C
L 2,4(0,1)
MVC H,2(2)  NH=H
MVI R,C
OPEN DFILF
MVC DFILF+8(2),DFILE+30
MVC A+4(4),SALT
LA 1,DFILE
EXCP (1)
WAIT (1)
```

```
L 1,REG
L 2,8(0,1)
SR 0,0
ST 0,0(0,2)
MVC 3(1,2),RU    FU:NR
L 2,12(0,1)
ST 0,0(0,2)
MVC 2(2,2),NB    NB:NBL
L 2,REG+4
BR 14
REG    DS 2F
SALT   DC X'47F0F038'
DFILE  DTFPH TYPEFLE=INPUT,CCWADDR=RBSK,DEVICE=2314,MOUNTED=ALL
RBSK   CCW 7,B,64,6
        CCW 49,C,64,5    SEARCH IC EC.
        CCW 8,*-8,0,0
        CCW 6,CL,0,8    READ RD DATA AREA
B       DC H'0'
C       DS H
H       DS H
R       DS C
CL      DS H
HD      DS H
RU      DS C
NB      DS CL3
        END
```

PUTCPC CSECT USAGE: CALL PUTCPC(INC,NH,AR,NBL)

- * PUTCPC ACTUALIZEAZA INREGISTRAREA DE CAPACITATE
- * SCRIE INREG. DE CAPACITATE INR=NR. DE INREG. EXISTENTE SI NBL=NF.!
- * DE BYTES NEFOLOSITI PE CILINCRUL NC. PISTA (CAPUL) NH

```
BALR 15,0
USING *,15
ST 2,REG
L 2,0(0,1)
MVC CC,2(2)    NC:CC
MVC CL,2(2)    NC:CL
L 2,4(0,1)
MVC HH,2(2)    NH:HH
MVC HD,2(2)    NH:HC
```

```
L 2.8(0.1)
MVC RU.3(2) NR:RU
L 2.12(0.1)
MVC NB.2(2) NEL:NR
OPEN DFILE
MVC DFILE+6(2).DFILE+30
MVC A+6(4).SALT
LA 1.DFILE
EXCP (1)
WAIT (1)
L 2.REG
BR 14
SALT DC X'47F0F054'
DFILE DTFPH TYPEFILE=INPUT.CCWADDR=MSK.DEVICE=2314.MOUNTEO=ALL
BB DC H'0'
CC DS H
HH DS H
DC X'00'
MSK CCM 7.BB.64.6
CCM 49.CC.64.5 SEARCH ID EQ
CCM 8.*-8.0.0
CCM 5.CL.0.8 WRITE DATA
LTORG
REG DS F
CL DS H
HD DS H
RU DS C
NB DS CL2
DC X'00'
END
```

FOLOSIREA RUTINELOR NEWFLE,WRITED,READD,UPDD,GETPC SI PUTCP.

```
DIMENSION R(2000),L(2000)
CALL NEWFLE
L(1)=1
L(1250)=1250
C 5000+1978 BYTES/PISTA(CAPACITATEA STANDARD)
CALL WRITED(L,5000,0,1,1,NTEST)
L(1)=-1
L(494)=-2
```

```
      CALL WRITED(L,1978,0,1,2,NTEST)
C   SE CITESC CELE DOUA INREGISTRARI
      CALL READD(K,8000,0,1,1,NTEST)
      PRINT 1,NTEST,K(1),K(1250)
1   FORMAT(4I5)
      CALL READD(K,2000,0,1,2,NTEST)
      PRINT 1,NTEST,K(1),K(494)
C   SE ACTUALIZEAZA O INREGISTRARE VECE
      I=-5
      CALL UPDD(1,4,0,1,1,NTEST)
      PRINT 1,NTEST
C   SE CITESC PRIMII 200 BYTES AI INREG. PRECEDENTE
      CALL READD(K,200,0,1,1,NTEST)
      PRINT 1,NTEST,K(1),K(2)
C   CITESTE, SCRIE SI IAR CITESTE INREG. DE CAPACITATE(CIL,0,PISTA 3)
      CALL GETCPC(0,3,NR,NBL)
      PRINT 1,NR,NBL
      CALL PUTCPC(0,3,2,0)
      CALL GETCPC(0,3,NR,NBL)
      PRINT 1,NR,NBL
C   DEPASIREA (CU MAX. 506 A) CAPACITATII STANDARD)(7294)
      CALL WRITED(L,7800,C,8,1,NTEST)
      PRINT 1,NTEST
C   CITIREA INREG. PRECEDENTE AR CANCELA JCR-LL
C   INCRCAM O INREG. MAI SCURTA(CU 20 BYTES)
      N=NTEST-20
      I=N/4
      L(1)=-9
      L(I)=-I
      CALL WRITED(L,1,0,8,1,NTEST)
C   DACA N=NTEST, INREG. FACUTA ESTE BLNA
      IF(N,NE,NTEST) GO TO 2
      CALL READD(K,9000,0,8,1,NTEST)
      PRINT 1,NTEST,K(1),K(1)
C   UTILIZAMI GRESITE ALE RUTINELCR
2   CALL WRITED(I,0,0,1,1,NTEST)
      PRINT 1,NTEST
      CALL WRITED(I,-1,0,1,1,NTEST)
      PRINT 1,NTEST
      CALL WRITED(I,4,0,9,90,NTEST)
```

```
PRINT I.NTEST
CALL WRITED(I,SCCC,C.9,I.NTEST)
PRINT I.NTEST
CALL UPDD(I.0.7.1.I.NTEST)
PRINT I.NTEST
CALL UPDD(I.-1.C.1.I.NTEST)
PRINT I.NTEST
CALL UPDD(I.4.C.1.9C.NTEST)
PRINT I.NTEST
CALL READD(I.0.C.1.I.NTEST)
PRINT I.NTEST
CALL READD(I.-1.C.1.I.NTEST)
PRINT I.NTEST
CALL READD(I.4.C.1.9C.NTEST)
PRINT I.NTEST
C   NC SAU NP GRESIT (IN AFARA EXTENT-UPILOR) CANCELEAZA JOB-UL
    STOP
    END
```

```
/*
// IOLYP MSD(4)
// EXEC LINKED
// ENUN DE FILE, 'MULTI-EXTENT FILE', 1.DA
// EXTENT SYS001....1.2
// EXTENT ....2.1
// EXTENT ....3.1
// EXTENT ....4.3
// EXEC
```

CITIREA PRIMEI INREGISTRARI DEPE PRIMA PISTA A FIECARUI CILINDRU
DEI FISIERUL SYSLNK SI TIPARIREA LA IMPRIMANTA.

```
// OPTION LINK
// EXEC PFORTRAN
    LOGICAL L*(17294)
C   CREAREA FISIERULUI SYSLNK IN VTDC (LINKEDITORUL L-A STEP)
    CALL NEWFLB
    DO 1 NC=55.09
    DO 1 NH=1.1
    NHC=NH-1
    DO 1 NR=1.1
    CALL READD(L,N,NC,NHC,NR,NTEST)
```

```
1 PRINT 2,NC,NHC,MR,N,((LI),I=1,N)
2 FORMAT(1X,I3,1X,I2,1X,I2,1X,I4,1),117A1/(1X,I32A1)
STOP
END
```

```
/*
// EXEC LINKEDT
// DDL OF FILE, 'SYSTEM LINK FILE X', 0
// EXTENT BYSLAK ...
// EXEC
```

CALCUL MATRICIAL

OBS. ROUTINELE CE URMEAZA SINT PERFECT COMPATIBILE CU CELE DIN SSP.

```
* SUBROUTINE LOC
*      COMPUTE A VECTOR SUBSCRIPT FOR AN ELEMENT IN A MATRIX OF
*      SPECIFIED STORAGE MODE
*      USAGE (FORTRAN): LOC(I,J,IR,N,M,MS)
*      I - ROW NUMBER OF ELEMENT
*      J - COLUMN NUMBER OF ELEMENT
*      IR - RESULTANT VECTOR SUBSCRIPT
*      N - NUMBER OF ROWS IN MATRIX
*      M - NUMBER OF COLUMNS IN MATRIX
*      MS - ONE DIGIT NUMBER FOR STORAGE MODE OF MATRIX
*           0 - GENERAL
*           1 - SYMMETRIC
*           2 - DIAGONAL
*      MS=0 SUBSCRIPT IS COMPUTED FOR A MATRIX WITH N*M ELEMENTS
*           IN STORAGE(GENERAL MATRIX)
*      MS=1 SUBSCRIPT IS COMPUTED FOR A MATRIX WITH N*(N+1)/2
*           ELEMENTS IN STORAGE(UPPER TRIANGLE OF SYMMETRIC
*           MATRIX).IF ELEMENT IS IN LOWER TRIANGULAR PORTION,
*           SUBSCRIPT IS CORRESPONDING ELEMENT IN UPPER TRIANGLE.
*      MS=2 SUBSCRIPT IS COMPUTED FOR A MATRIX WITH N ELEMENTS IN
*           STORAGE(DIAGONAL ELEMENTS OF DIAGONAL MATRIX).
*           IF THE ELEMENT IS NOT ON DIAGONAL(THEREFORE NOT IN
*           STORAGE),IR IS SET TO ZERO.
LOC      CSECT      USAGE: CALL LOC.(I,J,IR,N,M,MS)
*      LOC ALTERS GENERAL REGISTERS RC,R1 AND R15
*           USIAG #.R15
```

```
LR R0,R14 RETURN ADDR:R0
L R14,20(0,R1)
L R14,0(0,R14)
S R14,=F'1' MS-1:CC,R14
BNM SMDG BRANCH TO SYMM/DIAG MATRIX
L R15,4(0,R1)
L R15,0(0,R15)
AR R15,R14 J-1:R15
L R14,12(0,R1) ADDR OF N:R14
M R14,0(0,R14) N-(J-1):R15
L R14,0(0,R1) ADDR OF I:R14
A R15,0(0,R14) I+N(J-1):R15
L R1,8(0,R1) ADDR CF IR:R1
ST R15,0(0,R1) IP
LR R14,R0
BR R14
SMDG BR2 SM BRANCH FOR SYMMETRIC MATRIX
LR R14,R0 RETURN ADDR :R14
L R15,4(0,R1)
L R0,0(0,R15) J:R0
L R15,0(0,R1) ADDR OF I:R15
CL R0,0(0,R15) J-1:CC
L R1,8(0,R1) ADDR CF IR:R1
LA R15,0
ST R15,0(0,R1) 0:IR
ANE 0(0,R14)
ST R0,0(0,R1) J:IR
BR R14
SM STM R2,R3,REG SAVE R2,R3
LR R14,R0 RETURN ADDR:R14
L R2,4(0,R1)
L R2,0(0,R2) J:R2
L R3,0(0,R1)
L R3,0(0,R3) I:R3
CLR R2,R3 J-1:CC
L R1,8(0,R1) ADDR CF IR:R1
BL UB BRANCH FOR ELEMENT UNDER THE DIAGONAL
LR R0,R3 I:R0
LR R3,R2
BLTR R3,R0 J-1:R3
```



```

MR      R2,R2
SRL     R3,1          J+(J-1)/2:R3
AR      R3,R0
ST      R3,0(0,R1)   I+(J-1)/2:IR
LM      R2,R3,REG
RR      R14
UD      ACT     R3,#+4      I-1:R3
        LN      R0,R2      J:R0
        LA      R2,1(0,R3)   I:R2
        MR      R2,R2      I(I-1):R3
        SRL     R3,1
        AR      R3,R0      J+1(I-1)/2:R3
        ST      R3,0(0,R1)   IR
        LM      R2,R3,REG
        RR      R14
REG     US      2F
R0      EQU     0
R1      EQU     1
R2      EQU     2
R3      EQU     3
R13     EQU     13
R14     EQU     14
R15     EQU     15
        END

```

* SUBROUTINE GMPRC

```

*      MULTIPLY TWO GENERAL MATRICES TO FORM A RESULTANT GENERAL
*      MATRIX
*      USAEF (FORTRAN): CALL GMPRC(A,B,R,N,M,L)
*      A - NAME OF FIRST INPUT MATRIX
*      B - NAME OF SECOND INPUT MATRIX
*      R - NAME OF OUTPUT MATRIX
*      N - NUMBER OF ROWS IN A AND R
*      M - NUMBER OF COLUMNS IN A AND ROWS IN B
*      L - NUMBER OF COLUMNS IN B AND R
*      ALL MATRICES MUST BE STORED AS GENERAL MATRICES
*      MATRIX R CANNOT BE IN THE SAME LOCATION AS MATRICES A OR B
*      NUMBER OF COLUMNS OF MATRIX A MUST BE EQUAL TO NUMBER OF
*      ROWS OF MATRIX B
*      THE M BY L MATRIX R IS PREMULTIPLIED BY THE N BY M MATRIX A

```

* AND THE RESULT IS STORED IN THE M BY L MATRIX R

GMPRD CSECT USAGE:CALL GMPRD,(A,F,R,N,M,L)

* GMPRD ALTERS GENERAL REGISTERS R0 AND R1

USING *,R15

SAVE (2,10)

L R2,0(0,R1) ADDR OF A :R2

L R3,4(0,R1) ADDR OF B :R3

L R4,8(0,R1) ADDR OF R :R4

L R5,12(0,R1)

L R0,0(0,R5) N :R0

L R7,16(0,R1)

L R7,0(0,R7)

SLL R7,2

MR R6,R0

BGT R7,*,+4 4MN-1 : R7

LR R6,R0

SLL R6,2 4N : R6

L R9,20(0,R1)

L R9,0(0,R9)

MR R8,R6

BGT R9,*,+4 4NL-1:R9

LR R8,R6 4N : R8

SUR F0,F0

LA R7,0(R2,R7) A(1,1)+4MN-1:R7

LA R9,0(R4,R9) R(1,1)+4NL-1:R9

MATR LR R1,R3 REG. FOR B:R1

LR R5,R4 REG. FOR R :R5

ROW LR R10,R2 REG. FOR A :R10

LER F2,F0 0 : F2

ELMT LE F4,0(0,R10) A(I,J)

ME F4,0(0,R1) B(J,K)

AER F2,F4

LA R1,4(0,R1)

BXLE R10,R6,ELMT

STE F2,0(0,R5) R(I,K)

BXLE R5,R8,ROW

LA R2,4(0,R2)

LA R4,4(0,R4)

LA R7,4(0,R7) A(1,1)+4NM-1:R7

LA R9,4(0,R9) R(1,1)+4NL-1:R9

```
      RCT   RO, MATR
      RETURN (2, 10)
F0     EQU   0
F2     EQU   2
F4     EQU   4
R0     EQU   0
R1     EQU   1
R2     EQU   2
R3     EQU   3
R4     EQU   4
R5     EQU   5
R6     EQU   6
R7     EQU   7
R8     EQU   8
R9     EQU   9
R10    EQU   10
R15    EQU   15
      END
```

```
MPRD   USECT   USACF:CALL MPRD,(A,B,R,N,M,MSA,MSB,L)
```

```
*      MPRD ALTERS NO GENERAL REGISTERS
*      MULTIPLY TWO MATRICES TO FORM A RESULTANT MATRIX. A,B,R,
*      N,M AND L AS IN GMPRD. MSA=0 FOR A GENERAL MATRIX,1 FOR A
*      SIMMETRIC, 2 FOR A DIAGONAL. MSB SAME AS MSA,BUT FOR B.
*      FOR GENERAL MATRICES,GMPRD IS BETTER THAN MPRD
      USING *,R12
      SAVE (14,12)
      LR   R12,R15    LOCAL BASE REG
      L    R5,20(0,R1)
      L    R5,0(0,R5)   MSA:R5
      L    R6,24(0,R1)
      L    R6,0(0,R6)   MSB:R6
      LA   R7,0(R5,R6)  MSA+MSB:R7
      LA   R10,4        4:R10
      L    R2,0(0,R1)   ADDR OF A(1,1):R2
      L    R3,4(0,R1)   ADDR OF B(1,1):R3
      L    R4,8(0,R1)   ADDR OF R(1,1):R4
      L    R11,12(0,R1)
      L    R11,0(0,R11)
      ST   R11,N        N:N
```

```
SLL R11.2
BCT R11,*+4      4N-1:R11
CLR R7,R10
HNE NDGS BRANCH WHEN A AND B NOT DIAGONAL
LA R1.0 0:R1
LOOP LE F2.0(R1,R2)
ME F2.0(R1,R3)
STE F2.0(R1,R4)
BXLE R1,R10,LOOP
RETURN RETURN (14,12)
NDGS ST R5,MSA MSA:MSA
ST R6,MSB MSB:MSB
L R6.16(0,R1)
L R6.0(0,R6)
ST R6,M M:M,R5
L R5.28(0,R1)
L R5.0(0,R5)
ST R5,L L:L,R5
ST R13,REG13
SR R2,R10 (ADDR OF A(1,1))-4:R2
SR R3,R10 (ADDR OF B(1,1))-4:R3
LA R13.1 1:R13
ST R13,K 1:K
MATH LA R8.0 0:R8
SI R13.1 1:I
CLN ST R13,J 1:J
LR R7,R6 M:R7
SUR FO,FO 0:F0
ELMT CALL LOC.(I,J,IA,N,M,MSA)
L R9,IA
LTR R9,R9
BZ CONT BRANCH:A(I,J) NOT ON DIAG. OF DIAG.MATRIX A
SLL R9.2 4IA:R9
LE F2.0(R9,R2) A(I,J):F2
CALL LOC.(J,K,JB,M,L,MSB)
L R9,JB
LTF R9,R9
BZ CONT BRANCH:B(J,K) NOT ON DIAG. OF DIAG.MATRIX B
SLL R9.2 4JB:R9
ME F2.0(R9,R3) A(I,J)*B(J,K)
```

```

      AER  F0.F2
CONT  L    R9.J
      LA  R9.1(0.R9)
      ST  R9.J      J+1:J
      BCT R7.ELMT
      (STE, F0.0(R8.R4)  F(I,K)
      L    R9.I
      LA  R9.1(0.R9)
      ST  R9.I      I+1:I
      BXLE R8.P10.CLN
      LA  R4.0(R8.R4)  ADDR OF R41,K+1):R4
      L    R9.K
      LA  R9.1(0.R9)
      ST  R9.K      K+1:K
      BCT R5.MATR
      L    R13.REG13
      B    RETURN

```

```

REG13 DS F
I      DS F
J      DS F
IA     DS F
N      DS F
M      DS F
MSA    DS F
K      DS F
IB     DS F
L      DS F
MSB    DS F
R1     EQU 1
R2     EQU 2
R3     EQU 3
R4     EQU 4
R5     EQU 5
R6     EQU 6
R7     EQU 7
R8     EQU 8
R9     EQU 9
R10    EQU 10
R11    EQU 11
R12    EQU 12

```

```

R13 EQU 13
R14 EQU 14
R15 EQU 15
F0 EQU 0
F2 EQU 2
END

```

DMPRO ESTE MPRO IN PRECIZIE CUBLA.

DMPRO CSECT USAGE:CALL DMPRO.(A,B,R,N,M,MSA,MSB,L)

* DMPRO ALTERS NC GENERAL REGISTERS

```

        USING *,R12
        SAVE (14,12)
        LR R12,R15      LOAD BASE REGISTER
        L R5,20(0,R1)
        L R5,0(0,R5)    MSA:F5
        L R6,24(0,R1)
        L R6,0(0,R6)    MSB:F6
        LA R7,9(R5,R6)  MSA+MSB:R7
        LA R10,8        B:R10
        L R2,0(0,R1)    ADDR OF A(1,1):R2
        L R3,4(0,R1)    ADDR OF B(1,1):R3
        L R4,8(0,R1)    ADDR OF R(1,1):R4
        L R11,12(0,R1)
        L R11,0(0,R11)
        ST R11,N        N:N
        SLL R11,3
        BGT R11,#+4     3N-1:R11
        SLL R7,1
        CLR R7,R10
        BNE NDGS        BRANCH:A AND B NOT DIAGONAL
        LA R1,0         0:R1
LOOP    LD R2,0(R1,R2)
        MD R2,0(R1,R3)
        STD R2,0(R1,R4)
        BXLE R1,R10,LOOP
RETURN  RETURN (14,12)
NDGS   ST R5,MSA        MSA:MSA
        ST R6,MSB        MSB:MSB
        L R6,16(0,R1)
        L R6,0(0,R6)

```

```
ST R6,M M:M,R6
L R5,28(O,R1)
L R5,0(O,R5)
ST R5,L L:L,R5
ST R13,REG13
SR R2,R10 (ADDR OF A(1,1))-E:R2
SR R3,R10 (ADDR OF B(1,1))-E:R3
LA R13,1 1:R13
ST R13,K 1:K
MATK LA R8,0 C:R8
ST R13,I 1:I
CLN ST R13,J 1:J
LR R7,R6 M:R7
SMR F0,F0
ELMT CALL LOC.(I,J,IA,N,M,MSA)
L R9,IA
LTR R9,R9
BZ CONT BRANCH:A(I,J) NOT ON DIAG.OF DIAG.MATRIX A
SLL R9,3 DIA:R5
LD F2,0(R9,R2) A(I,J):F2
CALL LOC.(J,K,IB,M,L,MSB)
L R9,IB
LTR R9,R9
BZ CONT BRANCH:B(J,K) NOT ON DIAG.OF DIAG.MATRIX B
SLL R9,3 RIB:R9
MD F2,0(R9,R3) A(I,J)*B(J,K)
ADR F0,F2
CUNT L R9,J
IA R9,1(O,R9)
ST R9,J J+1:J
BCT R7,ELMT
S(L) F0,0(R8,R4) R(I,K)
L R9,I
LA R9,1(O,R9)
ST R9,I I+1:I
HXLE R8,R10,CLN
LA R4,0(R8,R4) ADDR OF R(I,K+1):R4
L R9,K
LA R9,1(O,R9)
ST R9,K K+1:K
```

```
      BCT   R5, MATR
      L     R13, REG13
      B     RETURN
REG13 DS    F
I      DS    F
J      DS    F
IA     DS    F
N      DS    F
M      DS    F
MSA    DS    F
K      DS    F
IB     DS    F
L      DS    F
MSB    DS    F
R1     EQU   1
R2     EQU   2
R3     EQU   3
R4     EQU   4
R5     EQU   5
R6     EQU   6
R7     EQU   7
R8     EQU   8
R9     EQU   9
R10    EQU   10
R11    EQU   11
R12    EQU   12
R13    EQU   13
R14    EQU   14
R15    EQU   15
FO     EQU   0
F2     EQU   2
      END
```

* SUBROUTINE TPRD

* TRANSPOSE A MATRIX AND POSTMULTIPLY BY ANOTHER TO FORM
* A RESULTANT MATRIX

* USAGE (FORTRAN): CALL TPRD(A,B,P,N,M,MSA,MSB,L)

* A - NAME OF FIRST INPUT MATRIX

* B - NAME OF SECOND INPUT MATRIX

* P - NAME OF OUTPUT MATRIX

* N - NUMBER OF ROWS IN A AND B
* M - NUMBER OF COLUMNS IN A AND ROWS IN R
* MSA - ONE DIGIT NUMBER FOR STORAGE MODE OF MATRIX A
* 0 - GENERAL
* 1 - SYMMETRIC
* 2 - DIAGONAL
* MSB - SAME AS MSA EXCEPT FOR MATRIX B
* L - NUMBER OF COLUMNS IN B AND R
* MATRIX R CANNOT BE IN THE SAME LOCATION AS MATRICES A OR B
* FOR A AND B GENERAL MATRICES, GTPRD IS BETTER THAN TPRD
* SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED
* LOC(MADE WITHOUT SAVE AREA FROM CALLING PROGRAMM)
* THE STORAGE MODE OF OUTPUT MATRIX IS ALWAYS GENERAL, EXCEPT
* FOR A AND B DIAGONAL WHEN STORAGE MODE OF R IS 2(DIAGONAL)
* MATRICES A AND B NOT CHANGED

TPRD CSECT USAGE: CALL TPRD, (A,B,R,N,M,MSA,MSB,L)

* TPRD ALTERS NO GENERAL REGISTERS

```
USING *.R12
SAVE (14,12)
LR R12,R15 LOAD BASE REG
L R5,20(0,R1)
L R5,0(0,R5) MSA:R5
L R6,24(0,R1)
L R6,0(0,R6) MSB:R6
LA R7,0(R5,R4) MSA+MSB:R7
LA R8,4 4:R8
L R2,0(0,R1) ADDR OF A(1,1):R2
L R3,4(0,R1) ADDR OF B(1,1):R3
L R4,8(0,R1) ADDR OF R(1,1):R4
L R9,12(0,R1)
L R9,0(0,R9)
ST R9,N M:N
SLL R9,2
BCT R9,++4 4N-1:R9
CLR R7,R8
BNE NDGS BRANCH WHEN A AND B NOT DIAGONAL
LA R1,0 (:R1)
LOOP LE F2,0(R1,R2)
ME F2,0(R1,R3)
STE F2,0(R1,R4)
```

```

      SXEF  R1,R8,LOCP
RETURN  RETURN (14,12)
NDGS   L    R11,16(O,P1)
      L    R11,0(O,S11)
      ST   R11,M    M:M,P11
      SLL  R11,2
      BCT  R11,7+4    4M-1:R11
      ST   R5,MSA    MSA:MSA
      ST   R6,MSB    MSB:MSB
      L    R5,28(O,P1)
      L    P5,0(O,R5)
      ST   R5,1    L:L,R5
      LK   R10,R8    4:R10
      ST   R13,REG13
      SR   R2,R10    (ADDR OF A(1,1))-4:R2
      SR   R3,R10    (ADDR OF B(1,1))-4:R2
      LA   R13,1    1:R13
      ST   R13,K    1:K
      L    R6,N    N:R5
MAIN   LA   R8,0    C:R8
      ST   R13,J    1:J
CLN    ST   R13,I    1:I
      LR   R7,R6    N:R7
      SUR  F0,F0    C:F0
FLM1   CALL LOC,(I,J,IA,N,M,MSA)
      L    R9,IA
      LTR  R9,R9
      BZ   CONT    FRANCH:A(1,J) CUT OF THE DIAG.,A DIAG.MATRIX
      SLL  R9,2    4IA:R9
      LE   F2,0(R9,F2)    A(I,J)
      CALL LOC,(1,K,IB,N,L,MSB)
      L    R9,IB
      LTR  R9,R9
      BZ   CONT    FRANCH:B(1,K) CUT OF THE DIAG.,B DIAG.MATRIX
      SLL  R9,2    4IB:R9
      ME   F2,0(R9,F3)    A(I,J)*B(I,K)
      AER  F0,F2
CCAT   L    F9,I
      LA   R9,11(O,R9)
      ST   R9,I    I+1:I
```

```

      BCT  R7,ELMT
      STE  F0,0(R0,R4)      R(J,K)
      L    R9,J
      LA   R9,1(0,R9)
      ST   R9,J      J+1:J
      RXLc R8,R10,CIN
      LA   R4,0(18,F4)      ADDR OF R(1,K+1):F4
      L    R9,K
      LA   R9,1(0,R9)
      ST   R9,K      K+1:K
      BCT  R5,MATR
      L    R13,REG13
      B    RETURN
REG13 DS    F
I      DS    F
J      DS    F
IA     DS    F
N      DS    F
M      DS    F
MSA    DS    F
K      DS    F
IP     DS    F
L      DS    F
MSB    DS    F
R1     EQU   1
R2     EQU   2
R3     EQU   3
R4     EQU   4
R5     EQU   5
R6     EQU   6
R7     EQU   7
R8     EQU   8
R9     EQU   9
R10    EQU   10
R11    EQU   11
R12    EQU   12
R13    EQU   13
R14    EQU   14
R15    EQU   15
F0     EQU   0

```

F2 EQU 2
END

```

*      SUBROUTINE MATA
*      PREMULTIPLY A MATRIX BY ITS TRANSPOSE
*      USAGE (FORTRAN): CALL MATA(A,P,A,P,MSA)
*      A - NAME OF INPUT MATRIX
*      P - NAME OF OUTPUT MATRIX
*      N - NUMBER OF ROWS IN A
*      M - NUMBER OF COLUMNS IN A, ALSO NUMBER OF ROWS AND NUMBER
*           OF COLUMNS IN R (R IS M BY M)
*      MSA - ONE DIGIT NUMBER FOR STORAGE MODE OF MATRIX A
*           0 - GENERAL
*           1 - SYMMETRIC
*           2 - DIAGONAL
*      MATRIX R CANNOT BE IN THE SAME LOCATION AS MATRIX A
*      MATRIX R IS ALWAYS WITH STORAGE MODE=1, EXCEPT FOR MATRIX A
*      DIAGONAL, WHEN STORAGE MODE OF R=2
*      SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED
*      LOC(MADE WITHOUT SAVE AREA FROM CALLING PROGRAM)
*      THE MATRIX A IS NOT CHANGED.
MATA CSECT  USAGE:CALL MATA(A,P,A,P,MSA)
*      MATA ALTERS NO GENERAL REGISTERS
      USING =.R12
      SAVE (14,12)
      LR   R12,R15   LOAD BASE REG
      L    R3,0(0,P1)  ADDR OF A(1,1):R3
      L    R4,4(0,P1)  ADDR OF P(1,1):R4
      L    R7,8(0,P1)
      L    R7,0(0,P7)   N:R7
      L    R6,16(0,P1)
      L    R10,0(0,P6)  MS:R10
      LA   R2,1        1:R2
      LA   R6,4        4:R6
      CTR  R2,R10
      BNE  ADG  BRANCH FOR A NOT DIAGONAL
      LA   R11,0      0:R11
      STI  R7,2
      SCL  R7,#+4     4A-1:R7
      CODE  R3,0(P1,R3)  A(1,1)

```

```

MER  F0.F0  A(I,J)**2
STE , F0.0(R(I,R4)  R(I,I)
BXLE R11.R6.LOOP
RETURN RETURN (14.12)
NOG  ST  R7.N  N:N
      ST  R10.MS  MS:MS
      L   R5.12(0.R1)
      L   R5.0(0.R5)
      ST  R5.N  M:P.R5
      SR  R3.R6  ACER OF A(1.1)-4:R3
      ST  R2.K  1:K
      LR  R8.R2
MATR, ST  R2.J  1:J
CLM  ST  R2.I  1:I
      LR  R6.R7  N:R6
      SUR F0.F0  C:FC
ELMT CALL LOC.(I,J,IA,N,M,MS)
      L   R9.IA
      SLL R9.2  4T:RS
      LE  F2.0(P9.R3)  A(I,J)
      CALL LOC.(I,K,IA,N,M,MS)
      L   R9.IA
      SLL R9.2  4T:RS
      ME  F2.0(R9.R3)  A(I,J)*A(I,K)
      AER F0.F2
      L   R9.I
      LA  R9.1(0.P9)
      ST  R9.I  I+1:I
      OCT P6.ELMT
      STE F0.0(0.R4)  R(J,K)
      LA  R4.4(0.R4)
      L   R9.J
      LA  R9.1(0.R9)
      ST  R9.J  J+1:J
      CLR R9.P8
      MOV CLM
      LA  R9.1(0.P8)
      L   R9.K
      LA  R9.1(0.P9)
      ST  R9.K  K+1:K

```

```
      ACT   R5, MATR
      B     RETURN
I      DS   F
J      DS   F
IA     DS   F
N      DS   F
M      DS   F
MS     DS   F
K      DS   F
R1     EQU  1
R2     EQU  2
R3     EQU  3
R4     EQU  4
R5     EQU  5
R6     EQU  6
R7     EQU  7
R8     EQU  8
R9     EQU  9
R10    EQU 10
R11    EQU 11
R12    EQU 12
R15    EQU 15
FJ     EQU  0
F2     EQU  2
      END
```

PROGRAM PRINCIPAL ASSEMBLER 31 SUBPROGRAM FORTRAN.

```
// OPTION LINK
// EXEC ASSEMBLY
      CSECT
      BALR 10,0
      USING *,10
      LA 13,SAVAR
      L 15,=V(ILFIBCOM)
      BAL 14,64(15)
      CALL SIN,(ARC)
      STE 0,RE7
      CALL WRITE,(ARG,RE2)
      EQU
ARG     DC E'1.57079'
```

```
REZ      DS F
SAVAR    DS 90
        END
```

/*

```
// EXEC FFORTRAN
        SUBROUTINE WRITE(X,Y)
        PRINT 1,X,Y
        1 FORMAT(2E17.7)
        RETURN
        END
```

/*

```
INCLUDE ILFSSCN
// EXEC LNKEDT
// EXEC
```

RUTINA WRITE ARE NEVOIE DE ILFIBCCM, CIA CARE CAUZA ACEASTA RUTINA TREBUIE INITIALIZATA. INCLUDE ILFSSCN ESTE NECESARA DE DARECE IN PROGRAMUL ASSEMBLER AM CHEMAT MODULUL ILFSSCN PRINTR-UN ENTRY (SIN). IN ACEST PROGRAM SE VEDE FOLCSIREA REGISTRELOR DE LEGARE (LINKAGE REGISTERS). DACA ILFIBCCM NU ESTE NECESARA (SAU NU ESTE DORITA), NU SE CERE INITIEREA

SPATIEREA LA IMPRIMANTA.

```
SPACE   CSECT   CALL SPACE, (NRLINES)
*  NRLINES—NR. DE LINII SARITE (SPATIATE) LA IMPRIMANTA
        STM    2,4,12(13)   6F SAVE AREA
        BALR   4,0
        USING *,4
        LA    2,0
        L     3,0(0,1)
        L     3,0(0,3)
        LTR   3,3
        BM    END
        D     2,TREI
        LTR   3,3
        BZ    CNT
        LA    1,CCB3
LOOP3    EXCP  (1)
        HCF  3,LOOP3
CNT      LTR   2,2
        BZ    END
        LA    1,CCB1
```

```
LOOP1  EXCP  (1)
      BGT  2.LOOP1
END    LM 2+4.12(13)
      BK   14
      CNUF 0.4
CCB1   OCB  SYSLSY.CCW1
CCB3   CC'  SYSLSY.CCW3
TREI   DC   F'3'
CCW1   CCW  11.TREI.0.1
CCW3   CCW  27.TREI.0.1
      END
```

TIPARTREA CARTELELOR CU IMPRIMANTA.

CSECT

- * TIPARESTE CARTELE DEPE CARD-READER (SI CELE DE JOB-CONTROL), PINA LA
- * PRIMA CARTELA /6 SAU // JOB SI NUMERCTEAZA PAGINILE TIPARITE. O CAR-
- * TELA CONTININD PAGEND INCEPE O PAGINA NOUA. NR. DE LINII(LL) PE O
- * PAGINA (UN RIND TIPARIT CU 2 SPATII CERE 2 LINII, UNUL CU UN SPATIU
- * CERE O LINIE IN LL) SI NR. DE ORDINE AL PRIMEI PAGINI(PPP) SE DAU
- * IN PRIMA CARTELA (LLPPP DIN PRIMA COLCANA, EXEMPLU, 50010)
- * SCRIEREA LA 2 RINDURI CERE IN PLUS O INSTRUCIUNE SI MODIFICAREA
- * UNUI CCW, MARCATE PRIN COMENTARII CU 5 ASTERISCURI

```
BALR  5.0
USING *.5
LA 1.CARDCCB
EXCP  (1)
LA 13.SAVEAR
CALL SPACE.(1)
L     2.NTL
WAIT  (1)
NC NZ.IOAREA
PACK  ND.NZ
CVR   1.ND
ST    1.N
NC    I(3).IOAREA+2
PACK  NRPG.I(3)
S     2.N
RCTR  2.0
RCTR  2.0
ST    2.M
```



```
NEWPAGE MVC I.M
AICI CALL SPACE,(I)
MVC BLNK+1(80).BLNK
UNPK IOAREA+32(3).NRPG
DC IOAREA+34(1).EF
TM IOAREA+32.15
BNZ PRT
MVI IOAREA+32.64
TM IOAREA+33.15
BC 7.PRT
MVI IOAREA+33.64
PRT LA 1.PRNTCCB
EXCP (1)
AP NRPG.UNU
L 1C.N
WAIT (1)
READCARD LA 1.CARDCCB
EXCP (1)
WAIT (1)
CLC PEND.IOAREA
BNE CONTA
A 10.M
ST 10.I
B AICI
CONTA TM 2(1).64
SO EOJ
EOF LA 1.PRNTCCB
EXCP (1)
BCTR 10.0 PT. SCRIERE LA 2 RINDURI(LINII) *****
WAIT (1)
RCT 10.READCARD
B NEWPAGE
FOJ CLC SLASH.IOARFA
BE EOF
A 10.M
ST 10.I
CALL SPACE,(I)
EOJ
SLASH DC XL2'615C'
CNDP 0.4
```

CARDCCB CCB SYSIPT.CARDCCB
PRNTCCB CCB SYSLST.PRNTCCW
NRPG DS XL3
EF DC X'FO'
CARDCCW CCW 2.IOAREA,0.80
PRNTCCW CCW 17.IOAREA,0.80 7/17 PT. SCRIERE LA 1/2 RINDURI ****
L DC F'65' NR. LINII SARITE LA INCEPUT, CU // OPTION LINK
NTL DC F'66' 96/72 PT. 8 SAU 5 LINII/INCH
N DS F
M DS F
I DC XL4'FFFFCFFF'
UNU DC P'1'
N7 DC XL2'FFCF'
BLNK DC C' '
IOAREA DS 80G
SAVEAR DS 6F
ND DS D
PEND DC CL6'PAGEND'

COMPLETARI LA SISTEMUL DE OPERARE CCS.

BIBLIOTECA RELOCATABILA.

UN MODUL DIN BIBLIOTECA RELOCATABILA (RLB-MODUL) POATE CONTINE INSTRUCIUNI DE LINKEDITOR (ACTION, PHASE, INCLUDE MCONAME SI ENTRY) SAU/SI O UNITATE DE PROGRAM (MAINPGM SAU SUBPROGRAM).

NUMELE RLB-MODULULUI ESTE FORMAT DIN 1-8 CARACTERE (FARA BLANC SI PRIMUL SAU NU FIE ASTERISC), DECI POATE FI ALTUL DECIT NUMELE UNITATII DE PROGRAM PE CARE AR CONTINE-O.

MODURILE PREZENTATE PT. CATALOGARE POT FI IN INSTR. SURSA (FORTRAN, ASSEMBLER) SAU MODULE OBJECT (COMPILETE/ASAMBLATE ANTERIORI), PE CARTELE SAU BENZI.

RLB-MODULUL DE TIP CLASIC (ARE ACELASI NUME-RLB CU NUMELE SUBPROGRAMULUI CONTINUT) SE FOLOSESTE IN MODUL CUNOSCUT: RLB-UL IN CARE SE GASESTE VA FI ASIGNAT IN MOMENTUL LINKEDITARII (INAINTE DE // EXEC LNKEDT) SAU MODULUL A FOST ADUS PE SYSLNK (VEZI OPICITE RLB-URI ..) MODULUL CU ALT NUME-RLB DECIT AL UNITATII CONTINUTE IN EL SE FOLOSESTE CU INCLUDE NUMEMODUL SI ASIGNIND RLB-UL IN CARE SE GASESTE EL SI TOATE, EVENTUAL, RLB-MODULE CHEMATE IN PROGRAM (RECURSIV, PRIN INCLUDE DIN ACESTE RLB-MODULE), PINA LA ADINCIMEA ADMISA (6 NIVELI).

UBS. NUMELE RLH-MODULELOR SINT UNICE PE BIBLIOTECA SI INTR-O LINKEDIT

ROUTINE PENTRU CATALOGARI IN BIBLIOTECA RELOCATABILA.

DEPE BANDA SYS004,UNDE SINT MODULE-OBJECT,TTCATLR IA FIECARE MODUL SI,
PRECEDAT DE INSTR. CATALR NUME, IL DUCE PE BANDA SYS005. NUME ESTE
NUMELE PRIMII CS (CONTROL SECTION) DIN FIECARE MODUL.

```
TTCATLR  CSECT
          BALR 15,0
          USING *.15
          LA 2,TCCB
RT        EXCP TCCB
          WAIT TCCB
          TM 4(2),1
          BO EOJ
          MVC CSNAME,IOAREA+17
          EXCP TCTCCB
RTU       EXCP TWCCB
          WAIT TWCCB
          CLC END,IOAREA+2
          BE RT
          EXCP TCCB
          WAIT TCCB
          TM 4(2),1
          BND RTU
EOJ       EOJ
          CNOP 0,4
TCCB      CCB SYS004,TCCW
TCCW      CCM 2,IOAREA,X'20',81
TCTCCB    CCB SYS005,TCTCCW
TCTCCW    CCM 1,CTLRCB,X'20',81
TWCCB     CCB SYS005,TWCCW
TWCCW     CCM 1,IOAREA,X'20',81
IOAREA    DS 84C
CTLRCB    DS 0C
          DC X'0240C3C1E3C1D3D940'
CSNAME    DS CL8
          DC 67C' '
END       DC CL3'END'
          END
```

TTCATLR FACE ACELASI LUCRU CA TTCATLR, DAR NUME ESTE LUAT DEPE
SYSIPT (CARD-READER). (DIN PRIMA COLONANA). CU TTCATLR SI CU TTD
CATER SE CATALOGHEAZA IN BIBLIOTECA RELOCATABILA, PASTRIND SAU NU A-
CELASI NUME IN BIBLIOTECA PELOCATABILA CU CEL DIN PROGRAMUL SURSA.
DACA NUMELE-RLR DIFERA DE CEL DIN PROGRAMUL SURSA, SE CERTE INSTE.
INCLUDE NUMERLR, PENTRU A FI ADUS AUTOPAT IN PROGRAM.

TTCATLR CSECT

```
      BALR 15.0
      USING =.15
      LA 2, TCCB
      LA 3, CDCCB
RC     EXCP CDCCB
      WAIT CDCCB
      TM 4(3).1
      BU EOJ
      EXCP TCTCCB
RT     EXCP TCCB
      WAIT TCCB
      TM 4(2).1
      BO EOJ
      EXCP TWCCB
      CLC END, IOAREA+2
      BE RC
      WAIT TWCCB
      B RT
EOJ    EOJ
      CNOP 0.4
CDCCB  CCB SYSIPT, CDCCW
CDCCW  CCW 2, MODNAME, X'20', 8
TCTCCB CCB SYS005, TCTCCW
TCTCCW CCW 1, CTLRCD, X'20', 81
TCCB   CCB SYS004, TCCW
TCCW   CCW 2, IOAREA, X'20', 81
TWCCB  CCB SYS005, TWCCW
TWCCW  CCW 1, IOAREA, X'20', 81
IOAREA DS 84C
CTLRCD DS 0C
      DC X'0240C3C1E3C1D3D940'
MODNAME DS 0L8
      DC 67C' '
```

END DC CL3'END'
END

CU ACESTE PUTINE (TTCATL R SI TTDCATLR) SI CU CITEVA PROCEDURI (CATALOGATE IN BIBLIOTECA DE PROCEDURI SAU INCLUSE IN JOB), SE POT CATALOGA IN BIBLIOTECILE RELOCATABILE MODULE DE ORICE TIP SI ADUSE SUB DIFERITE FORME (INSTR. SURSA PE DIFERITE MEDII, MODULE OBIECT ETCI)

ORIGINE BIBLIOTECI RELOCATABILE FOLosite DE UN PROGRAM.

// PAUSE ROG PANCA LUCRU PE 28)

// OPTION LINK

// EXEC PROC=RLBMATL

// EXEC PROC=\$\$RLB

PUNCH DELKE,ABCD,CLEBS

/*

// EXEC PROC=RLB130

// EXEC PROC=\$\$RLB

PUNCH TPRD.LOC,CLEBS

/*

// EXEC FFORTRAN

REAL*8 X,DE,DK

DIMENSION A(3,3),B(3,3)

DATA A/9*1.0/

CALL TPRD(A,A,B,3,3,0,0,3)

PRINT 1,4

1 FORMAT(3E15.5)

X=.5

CALL DELKE(X,DE,DK)

PRINT 1,DE,DK

STOP

END)

/*

// EXEC PROC=RLBIRNE

// EXEC LNKEDT

// EXEC

S-AM FOLOSIT 3 BIBLIOTECI RELOCATABILE IN CARE SE CAUTA MODULELE DELKE,ABCD, CLEBS (RLBMATL), TPRD.LOC,CLEBS (RLB130), SI ORICE MODUL DIN RLBIRNE.

PROCEDURELE FOLOSITE MAI SUS REALIZEAZA URMATOARELE OPERATII:

RLBMATL,RLB130 SI RLBIRNE ASIGNEAZA CEFLE TREFI BIBLIOTECI. \$\$RLB CAUTA

MODULELE CERUTE PRIN COMANDA PUNCH IN BIBLIOTECA ASIGNATA IN ACEL MOMENT SI LE PUNE PE FISIERUL SYSLNK.

ACESTE PROCEDURI POT FI CATALOGATE IN BIBLIOTECA DE PROCEDURI SAU POT APARE DIRECT IN JOB-UL DE CATALOGARE.

FISIERELE DE LUCRU ALE SISTEMULUI DE OPERARE DOS.

SISTEMUL DOS FOLOSESTE CITEVA FISIERE (PE UNITATILE LOGICE SYSC01, SYSC02, SYSC03, SYSLNK ETC) PENTRU A COMPILA/ASAMBLA/LINKEDITA ETC. SPATIUL ACESTOR FISIERE POATE FI UTILIZAT DE PROGRAMELE UTILIZATORULUI CA FISIERE DE LUCRU. FISIERELE PE SYSC01 - SYSC03 SE POT FOLCSI DIRECT, FARA NICI-O ACTIUNE IN PLUS (SE FAC SCRIERI/CITIRI CU NUMEREL DE REFERIRE DE FISIER-DATA SET REFERENCE NUMBER- 4,5 SI 6). SPATIUL FISIERULUI SYSLNK POATE FI SI EL FOLOSIT. MAI MULT, INTREG SPATIUL ACESTOR FISIERE SE POATE FOLOSI IN PROGRAMELE FORTRAN/ASSEMBLER, CU ORICE NUMERE DE REFERIRE, ORGANIZATE ORICUM, DACA SE CERE ELIBERAREA ACESTUI SPATIU CU URMATOARELE INSTRUCIUNI (PUSE IN CAPUL JOB-ULUI):

```
// OPTION LINK
// EXEC ASSEMBLY
      EOJ
      END
```

/*

```
// EXEC LNKEDT
```

DACA JOB-UL PROGRAMATORULUI FACE SI EL ASAMBLARE SI LINKEDITARE, NICI NU SE CER ACESTE INSTRUCIUNI. PENTRU A FOLOSI ACEST SPATIU DUPA DORINTA PROGRAMATORULUI, EL TREBUIE SA STIE LOCALIZAREA PRECISA A ACESTOR SPATII, PENTRU A SCRIE EXTENT-URILE SI ASSGN-URILE RESPECTIVE. CEL MAI BINE ESTE SA FORMEZE PROCEDURI PE CARE SA LE FOLOSEASCA ORICE DORITOR. DAR FISIERELE DE LUCRU CREATE PE SPATIUL FISIERELOR DE SISTEM TREBUIE SA AIBA RETENTIA ZERO, ALTFEL APAR NEPLACERI. DIN ACESTE CAUZE, ASTFEL DE PROCEDURI SE FAC NUMAI DE PROGRAMATORII DE SISTEM.

CONTINUT

<u>PROGRAME FORTRAN</u>	PAG.
PROGRAMARE LINIARA	1
CALCUL INTEGRAL	29
IRADIAREA TUMORILOR MALIGNE	45
CITIREA MATRICILOR RARE	55
SIMETRIZAREA SUBMATRICILOR UNEI MATRICI	55
SISTEME ALGEBRICE LINIARE COMPLEXE	56
<u>PROGRAME ASSEMBLER</u>	57
SUPRIMAREA INTRERUPERILOR DE EXPONENT-UNDERFLOW	58
INTRODUCEREA INTRERUPERILOR DE EXPONENT-UNDERFLOW	58
RUTINA PENTRU ABNORMAL-END	58
PRELUCRAREA INTRERUPERILOR DE PROGRAM-CHECK	59
DATA RULARII PROGRAMULUI	60
SUPRIMAREA UNEI ASIGNARI TEMPORARE	60
ULTIMUL BYTE OCUPAT DE UN PROGRAM	61
INCARCAREA SI EXECUTIA UNUI PROGRAM DIN CLB	61
SECUNDA EXECUTIEI PROGRAMULUI	61
TRECEREA CARTELELOR DEPE CARD-READER PE BANDA MAGNETICA	62
FISIERE PE BENZI MAGNETICE	62
FISIERE (MONO SI MULTI-EXTENT) PE DISCUPI MAGNETICE	67
CALCUL MATRICIAL	67
SUBPROGRAMELE FORTRAN IN ASSEMBLER	92
SPATIAREA IMPRIMANTEI	93
TRECEREA CARTELELOR DEPE CARD-READER PE IMPRIMANTA	94
COMPLETARI LA SISTEMUL DE OPERARE DCS	96