

Prove d'esame

anno 1999

***Fondamenti di programmazione
dei calcolatori elettronici***
Corso di laurea in Ingegneria meccanica

Edit by Ermanno Napoli

L'uomo passa attraverso foreste di simboli.
Baudelaire

INDICE

Prova d'esame del 18 gennaio 1999	2
Prova d'esame del 22 febbraio 1999	9
Prova d'esame del 22 marzo 1999	12
Prova d'esame del 18 maggio 1999	18
Prova d'esame del 8 giugno 1999	22
Prova d'esame del 22 giugno 1999	27
Prova d'esame del 13 luglio 1999	31
Prova d'esame del 4 ottobre 1999	35
Prova d'esame del 7 novembre 1999	41
Prova d'esame del 6 dicembre 1999	46

PROVA D'ESAME del 18 gennaio 1999

Siano date in ingresso, da file o da tastiera, le dimensioni ed i valori di due matrici di interi MAT1 avente M righe ed N colonne, e MAT2, avente lo stesso numero M di righe e T di colonne. Si scriva un programma che verifichi se le due matrici hanno una colonna uguale, non necessariamente nella stessa posizione. In caso positivo, si eliminano dalle matrici le colonne individuate. Si stampino, infine, le matrici modificate. Si strutturi il programma in opportuni sottoprogrammi.

ESEMPIO

MAT1	INPUT										MAT2
											3
											4
1	7	3	8	2	5	7	3	2	24		
3	12	6	28	4	7	12	6	10	21		
20	22	4	18	17	12	38	4	8	19		

OUTPUT	MAT1									MAT2
1	7	8	2	5	7	2	24			
3	12	28	4	7	12	10	21			
20	22	18	17	12	38	8	19			

DESCRIZIONE ALGORITMO

Lette le matrici in ingresso, l'algoritmo viene impostato nella seguente maniera: si effettua uno scorrimento della prima matrice per colonne; ogni colonna viene messa a confronto con ognuna delle colonne della seconda matrice. Se le colonne risultano uguali allora si eliminano dalle rispettive matrici e si stampano infine le matrici ottenute. L'algoritmo cessa non appena vengono individuate due colonne uguali o quando, considerate tutte le colonne della prima matrice, nessuna di esse risulta essere uguale a qualcuna delle colonne della seconda matrice.

DOCUMENTAZIONE INTERNA AL PROGRAMMA

1- Costanti adoperate:

Nome	Valore	Tipo	Attributo
CARDR	50	intero	Una delle dimensioni massime del array
CARDC	50	Intero	Una delle dimensioni massime del array

2-Tipi di dato definiti dall'utente:

Nome	Definizione	Descrizione
matrice	Array(1..cardr,1..cardc)of integer	Deve contenere i numeri su cui si effettua l'elaborazione

3-Struttura dei dati:

• Informazioni di ingresso:

Nome	Tipo	Attributo
M	integer	Indica il numero di righe della matrice in ingresso
N	integer	Indica il numero di colonne della prima matrice
T	integer	Indica il numero di colonne della seconda matrice
MAT1	matrice	Indica il generico elemento della prima matrice
MAT2	matrice	Indica il generico elemento della seconda matrice

• Informazioni di uscita:

Nome	Tipo	Attributo
MAT1	matrice	Indica il generico elemento della prima matrice dopo l'elaborazione
MAT2	matrice	Indica il generico elemento della seconda matrice dopo l'elaborazione

• Informazioni di algoritmo:

Nome	Tipo	Attributo
I	integer	Aggiorna il valore della riga corrente della Mat1 nei vari cicli "for" e "while..do"
J	integer	Aggiorna il valore della colonna corrente della Mat1 nei vari cicli "for" e "while..do2"
H	integer	Aggiorna il valore della riga della Mat2 nei vari cicli "for" e "while..do"
K	integer	Aggiorna il valore della colonna della Mat2 nei vari cicli "for" e "while..do"
TROVAT	boolean	Segnala se sono state trovate due colonne uguali
POSIZ1	integer	Indica il numero di colonna della Mat1 su cui si sta effettuando l'elaborazione
POSIZ2	integer	Indica il numero di colonne della Mat2 su cui si sta effettuando l'elaborazione

4- Tabella sottoprogramma:

• Dichiarazione parametri attuali:

Nome	Tipo	Input	Output	Descrizione
N	integer	si	si	Si veda la tabella "Informazioni in ingresso"
M	integer	si	si	Si veda la tabella "Informazioni in ingresso"
CARDR	integer	si	no	Si veda la tabelle "Costanti adoperate"
MAT1	integer	si	si	Si veda la tabella "Informazioni in ingresso"
MAT2	integer	si	si	Si veda la tabella "Informazioni in ingresso"

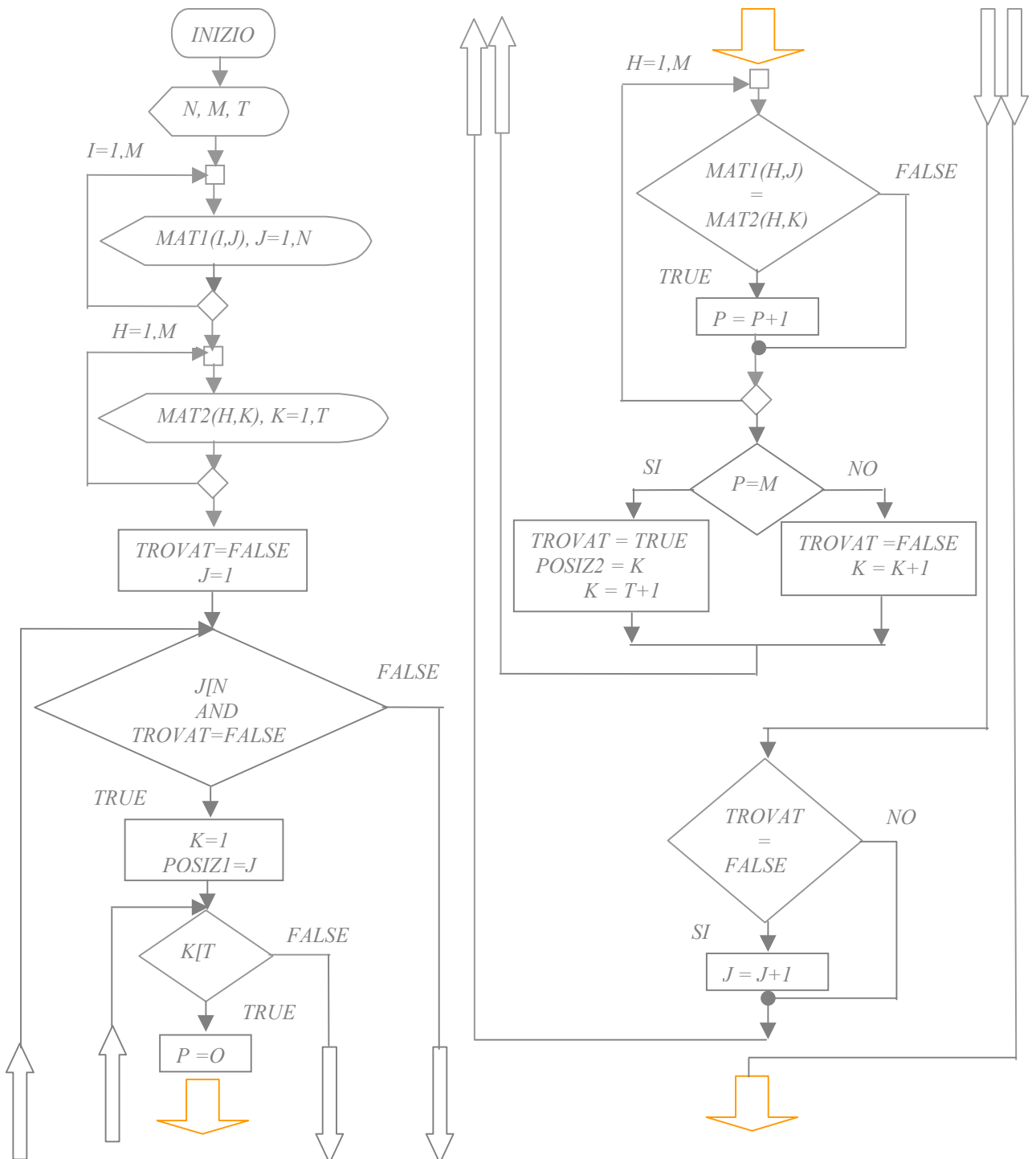
• Dichiarazione parametri formali:

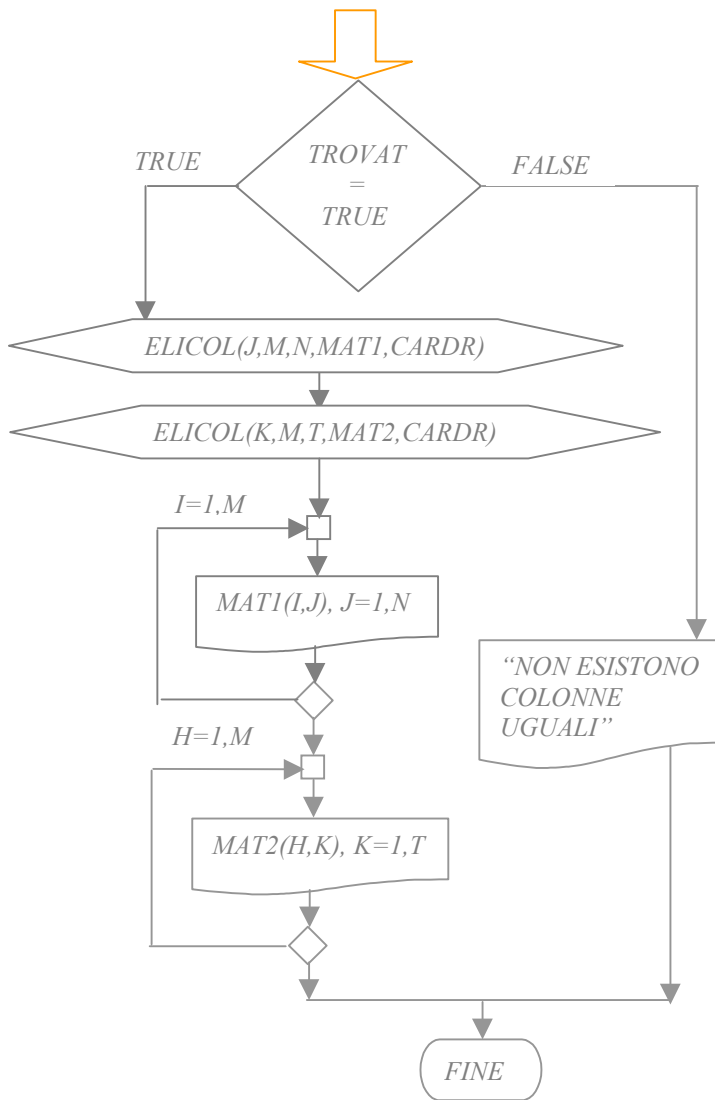
Nome	Tipo	Input	Output	Descrizione
COLON	integer	si	no	Assume il valore di un indice di riga
P	integer	si	no	Assume il valore di numero di righe
S	integer	si	si	Assume il valore di numero di colonne
MAT	integer	si	si	Assume il valore di un elemento generico di un array bidimensionale
CARDR	integer	si	no	Assume il valore di una dimensione massima di un array

• Dichiarazioni variabili locali al sottoprogramma:

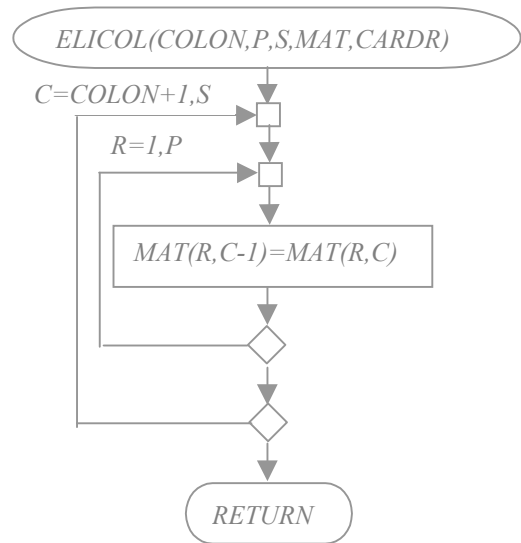
Nome variabile	Tipo	Attributo
R	integer	Variabile che aggiorna il ciclo "do"
C	integer	idem

FLOW CHART
Programma "UGUALC"





FLOW CHART
Sottoprogramma “ELICOL”



CODIFICA FORTRAN

```

PROGRAM UGUALC
c   Date due matrici mat1,mat2, di M,N e M,T
c   righe e colonne; si verifichi se le due
c   matrici hanno una colonna uguale, non
c   necessariamente nella stessa posizione.
c   In caso positivo si eliminino le colonne
c   e si stampino le matrici
c   Autore: Ermanno Napoli
c   Dichiarazione costanti
integer maxrig,maxcol
parameter(maxrig=50,maxcol=50)
c   Dichiarazione tipo
integer mat1(cardr,cardc), mat2(cardr,cardc)
c   Dichiarazione variabili
integer posiz1,posiz2,m,n
integer t,j,h,i,k,p
logical trovat
  
```

```

c Dichirazione sottoprogramma
external elicoll
c Begin
c Acquisizione dati in ingresso
write(*,*)'Dammi il numero di righe delle matrici'
write(*,*)
read(*,1) m
write(*,*)'Dammi il numero di colonne della prima matrice'
write(*,*)
read(*,1) n
write(*,*)'Dammi il numero di colonne della seconda matrice'
write(*,*)
read(*,1) t
write(*,*)'Dammi la prima matrice'
write(*,*)
do 10 i=1,m
10 read(*,2) (mat1(i,j), j=1,n)
continue
write(*,*)'Dammi la seconda matrice'
write(*,*)
do 20 h=1,m
20 read(*,2) (mat2(h,k), k=1,t)
continue
c Inizializzazione variabili: al momento non
c e' stata trovata nessuna colonna uguale
c (trovat=false), e si inizia dalla prima
c colonna della prima matrice
trovat=.false.
j=1
c Inizio ciclo "while..do": fino a quando non
c sono terminate le colonne della prima matrice
c e non e' stata trovata nessuna colonna uguale
c si effettuano le operazioni di seguito riportate
100 continue
if(.not.((j.le.n).and(.not.trovat))) goto 200
posiz1=j
k=1
c Inizio ciclo "while..do": si confronti la colonna
c della prima matrice con tutte le colonne della
c seconda matrice fino a quando non terminino o
c si trovino due colonne uguali
300 continue
if(.not.(k.le.t)) goto 400
p=0
c il controllo tra elementi delle colonne
c viene svolto da un ciclo "do"
do 30 h=1,m
if(mat1(h,j).eq.mat2(h,k)) then
p=p+1
endif
30 continue
c Verifica se le colonne sono uguali
if (p.eq.m) then
trovat=.true.
posiz2=k
k=t+1
else
trovat=.false.
k=k+1
endif
goto 300
400 continue
c Fine "while..do"

```

```

c      Controlla se sono state trovate le colonne
c      uguali in caso negativo si passa alla colonna
c      successiva di mat1
      if(.not.trovat) then
        j=j+1
      endif
      goto 100
200   continue
c      Fine "while..do"
c      Controlla se le colonne sono state trovate
c      in caso affermativo elimina le colonne dalle
c      rispettive matrici e stampa i risultati;
c      altrimenti stampa una frase ammonitrice
      if(trovat) then
c      Si elimina la colonna di mat1 con un sottoprogramma
        call elicol (j,m,n,mat1,cardr)
c      Si elimina la colonna di mat2 con un sottoprogramma
        call elicol (k,m,t,mat2,cardr)
        do 40 i=1,m
          write(*,3) (mat1(i,j), j=1,n)
40     continue
          write(*,*)
          do 50 h=1,m
            write(*,3) (mat2(h,k), k=1,t)
50     continue
          else
            write(*,4)'Non esistono colonne uguali'
          endif
1      format(i2)
2      format(30i3)
3      format(1x,30i3)
4      format(1x,a28)
        stop
      end
c *****
      subroutine elicol(colon,p,s,mat,cardr)
      integer colon,p,cardr
      integer s
      integer mat(cardr,*)
      integer r,c
c      Begin
        do 10 c=colon+1,s
          do 20 r=1,p
            mat(r,c-1)=mat(r,c)
20     continue
10     continue
          s=s-1
          return
        end

```

PROVA D'ESAME del 22 febbraio 1999

Siano dati in ingresso (da file o da tastiera) due vettori V1 e V2 contenenti, rispettivamente, N1 e N2 valori interi. Si scriva un programma che ricerchi gli elementi appartenenti a V2 che siano uguali alla somma degli elementi di V1 e inserisca, al posto di ogni elemento di V2 con tale caratteristica, la sequenza di elementi contenuti in V1. Infine, si produca in uscita il vettore V2 a valle della modifica.

ESEMPIO: Ingresso V1: 2 3 6 1
 V2: 5 12 18 4 12 4 1 12

Uscita V2: 5 2 3 6 1 18 4 2 3 6 1 4 1 2 3 6 1

DESCRIZIONE ALGORITMO

Si sommano tutti gli elementi del vettore V1 e si memorizza tale valore in una variabile (SOM). L'algoritmo consiste nel troncare (una volta individuato con un semplicissimo confronto tra la variabile suddetta e l'elemento corrente di V2, l'elemento secondo le specifiche richieste) il vettore V2 degli elementi successivi a quello caratteristico dopo averli ovviamente memorizzati in un apposito vettore appoggio (vetapp); a questo punto si accodano al vettore V2 troncato gli elementi di V1 e poi si aggiungono quelli del vettore appoggio.

DOCUMENTAZIONE INTERNA AL PROGRAMMA

1- Costanti adoperate:

Nome	Valore	Tipo	Attributo
CARD	100	intero	Dimensione massima array

2-Tipi di dato definiti dall'utente:

Nome	Definizione	Descrizione
vettore	array(1..card) of integer	Contiene i numeri su cui si deve operare

3-Struttura dei dati:

•Informazioni di ingresso:

Nome	Tipo	Attributo
N1	integer	Riempimento del vettore V1
N2	integer	Riempimento del vettore V2
VETT1	vettore	Contiene i numeri interi la cui somma deve essere utilizzata nell'elaborazione
VETT2	vettore	Contiene i numeri interi tra cui cercare la somma degli elementi del vettore V1

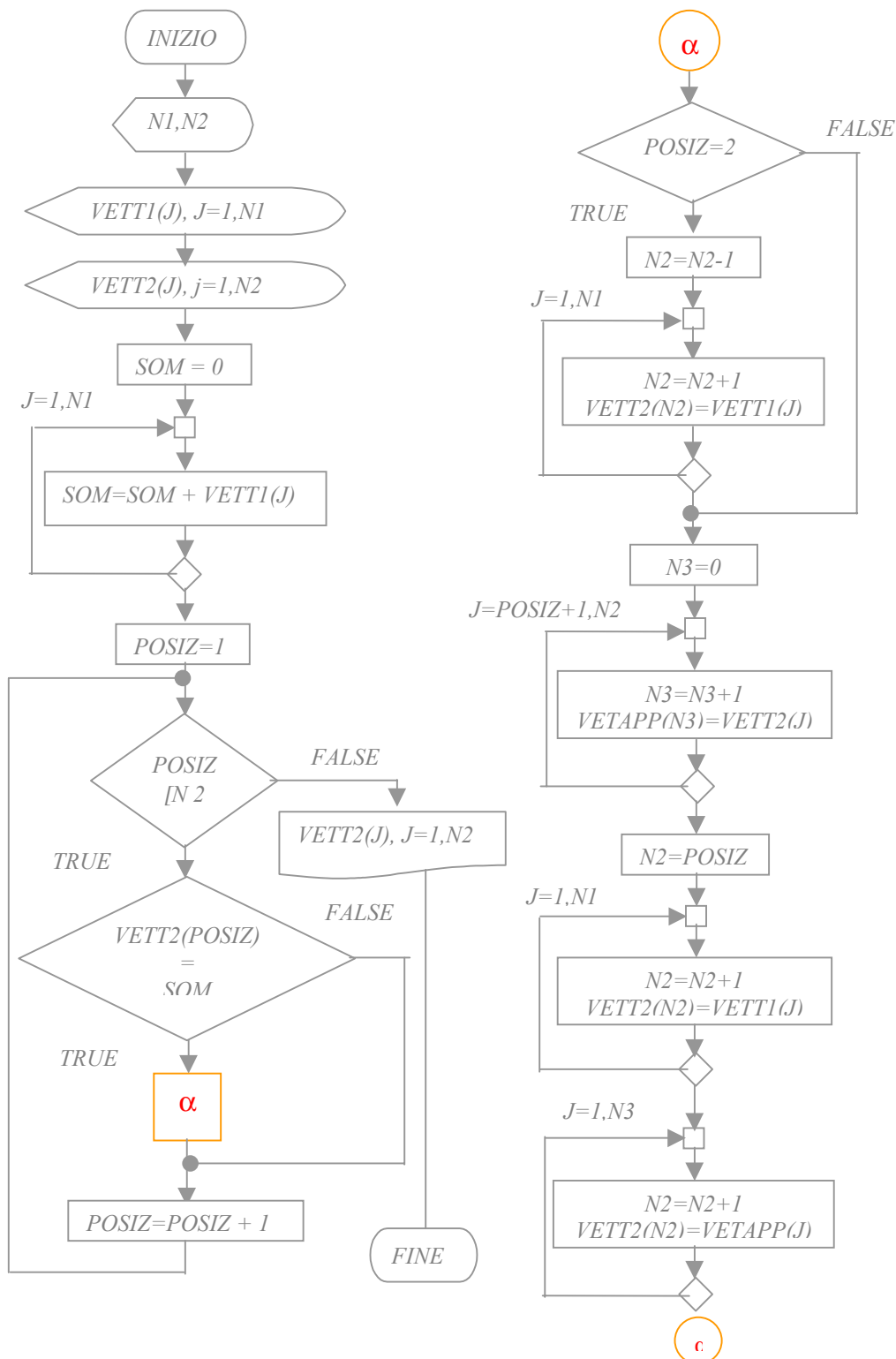
•Informazioni di uscita:

Nome	Tipo	Attributo
VETT2	vettore	Contiene l'insieme di numeri interi a valle delle modifiche richieste dal problema

•Informazioni di algoritmo:

Nome	Tipo	Attributo
POSIZ	integer	Indica la posizione nel vettore V2 dell'elemento corrente
J	integer	Aggiorna il valore della posizione degli elementi dei vettori V1, V2, vetapp
SOM	integer	Assume il valore di somma degli elementi del vettore V1
N3	integer	Riempimento vettore appoggio
VETAPP	vettore	Rappresenta il generico elemento del vettore appoggio

FLOW CHART
Programma "INFSOM"



```

PROGRAM INFOSOM
c   Dati due vettori V1 e V2 di interi
c   e di riempimenti N1 e N2, trovare tra
c   gli elementi di V2 l'elemento di valore
c   pari alla somma degli elementi di V1 e
c   sostituire al suo posto l'intero vettore V1
c   Autore:Ermanno Napoli
c   Dichiarazione costanti
INTEGER CARD
PARAMETER(CARD=100)
c   Dichiarazione tipo
INTEGER VETT1(CARD),VETT2(CARD),VETAPP(CARD)
c   Dichiarazione variabili
INTEGER J,POSIZ,N1,N2,N3,SOM
LOGICAL TROVAT
c   Dichiarazione sottoprogramma
c   NESSUNO
c   Begin
c   Acquisizione dati in ingresso
WRITE(*,*)'Dammi il riempimento del primo vettore:'
READ(*,'(I2)') N1
WRITE(*,*)'Dammi il primo vettore'
WRITE(*,*)
READ(*,'(30I3)')(VETT1(J), J=1,N1)
WRITE(*,*)'Dammi il riempimento del secondo vettore:'
READ(*,'(I2)') N2
WRITE(*,*)'Dammi il secondo vettore'
WRITE(*,*)
READ(*,'(30I3)')(VETT2(J), J=1,N2)
c   Determinazione somma di tutti gli elementi
c   del vettore V1
SOM=0
DO 10 J=1,N1
SOM=SOM+VETT1(J)
10  CONTINUE
c   Inizio "while..do":per ogni elemento del
c   vettore V1 si effettuano le operazioni
c   seguenti
POSIZ=1
100 CONTINUE
IF(.NOT.(POSIZ.LE.N2)) GOTO 200
c   Confronto tra l'elemento corrente di V2
c   e il valore di "som"
IF(VETT2(POSIZ).EQ.SOM) THEN
c   Si verifica se l'elemento cercato si
c   trova in ultima posizione ;in caso positivo
c   si procede come mostrato
IF(POSIZ.EQ.N2) THEN
N2=N2-1
DO 20 J=1,N1
N2=N2+1
VETT2(N2)=VETT1(J)
20  CONTINUE
ENDIF
c   Si memorizzano gli elementi di V2 successivi
c   a vett2(posiz)nel vettore appoggio
N3=0
DO 30 J=POSIZ+1,N2
N3=N3+1
VETAPP(N3)=VETT2(J)
30  CONTINUE

```

```

c      Si tronca il vettore V2 dei termini
c      eccedenti la posizione corrente
      N2=POSIZ
c      Si accodano gli elementi di V1 a V2
      DO 40 J=1,N1
      N2=N2+1
      VETT2(N2)=VETT1(J)
40     CONTINUE
      DO 50 J=1,N3
      N2=N2+1
      VETT2(N2)=VETAPP(J)
50     CONTINUE
c      Si fa assumere a posiz il valore successivo,
c      cioè la posizione dell'elemento a valle
c      dell'inserimento di V1
      POSIZ=POSIZ+N1
      ELSE
      POSIZ=POSIZ+1
      ENDIF
      GOTO 100
200    CONTINUE
      WRITE(*,*)
      WRITE(*,'(1X,30I3)') (VETT2(J), J=1,N2)
      STOP
      END

```

PROVA D'ESAME del 22 marzo 1999

Siano assegnati in ingresso, da file o da tastiera, il riempimento e gli elementi di un vettore di interi. Si scriva un programma che riordini il vettore assegnato in modo tale che i valori negativi precedano quelli positivi ed i primi abbiano ordinamento crescente, mentre i secondi siano ordinati in modo decrescente. Si stampi il vettore ottenuto dopo aver eliminato tutte le coppie di segno opposto e modulo uguale. Si strutturi il programma in opportuni sottoprogrammi.

ESEMPIO	INPUT	8							
		-8	12	-15	23	-50	15	-52	50
	OUTPUT	4							
		-52	-8	23	12				

DESCRIZIONE ALGORITMO

E' necessario individuare nel vettore in ingresso tutte le coppie di segno opposto e modulo uguale; in questo modo non si avranno problemi nelle operazioni successive quali l'ordinamento crescente/decescente degli elementi. L'individuazione di tali coppie si effettua sommando "algebricamente" l'elemento corrente con ciascuno dei rimanenti. La coppia è individuata quando ovviamente la somma è nulla. L'output finale si ottiene come unione di due vettori in ognuno dei quali erano stati posti i numeri negativi e positivi e successivamente ordinati secondo le specifiche del problema.

DOCUMENTAZIONE INTERNA AL PROGRAMMA

1- Costanti adoperate:

Nome	Valore	Tipo	Attributo
CARD	50	intero	La dimensione massima di un array

2- Tipi di dato definiti dall'utente:

Nome	Definizione	Descrizione
vettore	array(1..card) of integer	Deve contenere i numeri relativi su cui si deve operare

3-Struttura dei dati:

• Informazioni di ingresso:

Nome	Tipo	Attributo
N	integer	Riempimento del vettore in ingresso
VETT	vettore	Contiene i numeri interi relativi

• Informazioni di uscita:

Nome	Tipo	Attributo
N3	integer	Riempimento del vettore in uscita
VETT3	vettore	Deve contenere i numeri al termine dell'elaborazione

• Informazioni di algoritmo:

Nome	Tipo	Attributo
K	integer	Aggiorna la posizione dell'elemento corrente nel vettore in input nel ciclo "while..do"
H	integer	Aggiorna la posizione dell'elemento successivo a K
POS1	integer	Segnala l'indice di posizione del primo elemento costituente la coppia di valori uguali ma opposti
POS2	integer	Segnala l'indice di posizione del secondo elemento costituente la coppia di valori uguali ma opposti
VETT1	vettore	Contiene i numeri positivi del vettore in input
VETT2	vettore	Contiene i numeri negativi del vettore in input
N1	integer	Riempimento VETT1
N2	integer	Riempimento VETT2
I	integer	Aggiorna il valore della posizione del generico elemento dei VETT, VETT1, VETT2 nei cicli "do"

4-Tabella sottoprogrammi:

• Dichiarazione parametri attuali:

Nome	Tipo	Input	Output	Descrizione
POS1	integer	si	no	Si veda tabella "Informazioni di algoritmo"
POS2	integer	si	no	Si veda tabella "Informazioni di algoritmo"
N	integer	si	si	Si veda tabella "Informazioni di ingresso"
VETT	vettore	si	si	Si veda tabella "Informazioni di ingresso"
N1	integer	si	si	Si veda tabella "Informazioni di algoritmo"
VETT1	vettore	si	si	Si veda tabella "Informazioni di algoritmo"
N2	integer	si	si	Si veda tabella "Informazioni di algoritmo"
VETT2	vettore	si	si	Si veda tabella "Informazioni di algoritmo"

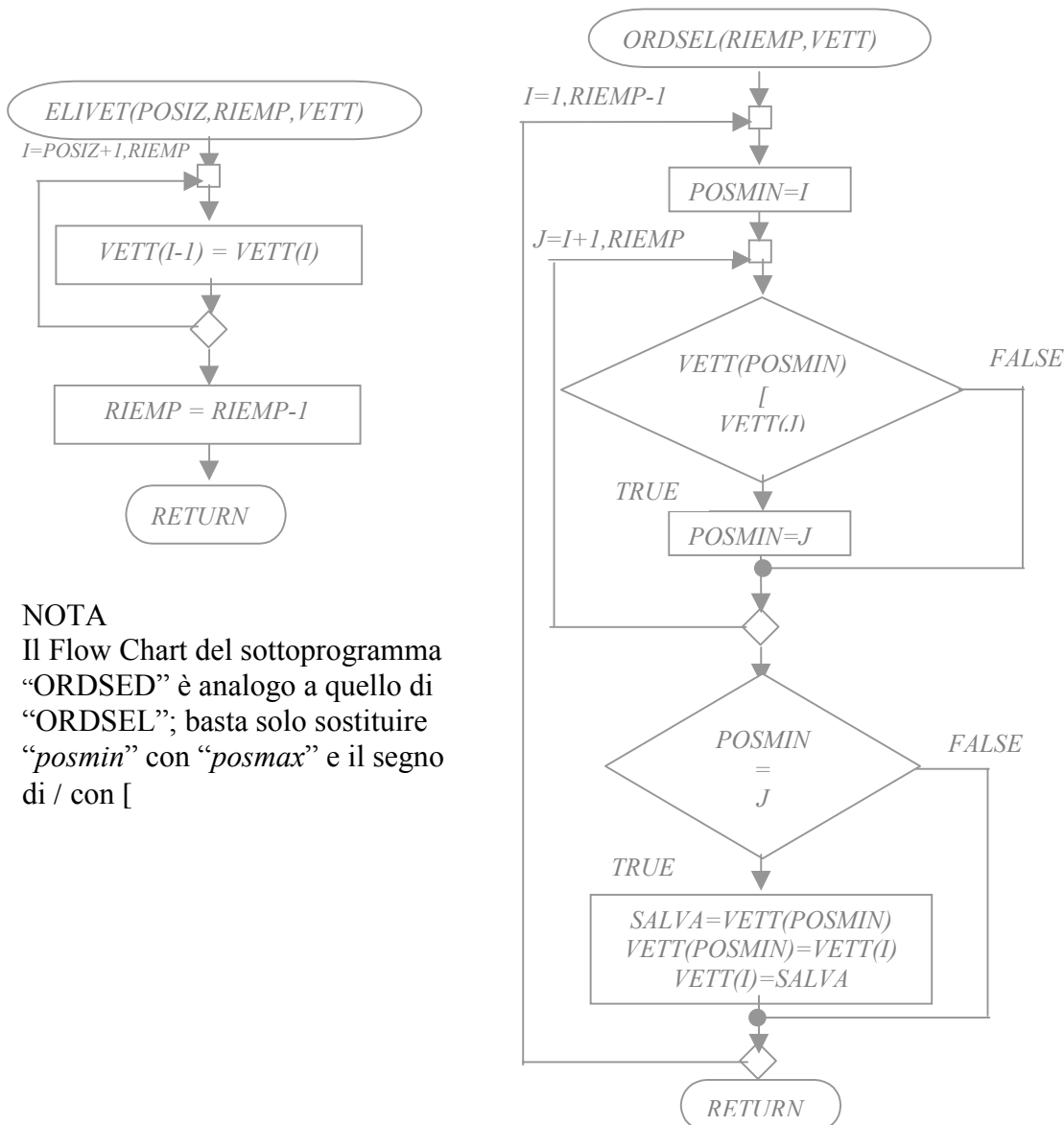
• Dichiarazione parametri formali:

Nome	Tipo	Input	Output	Descrizione
POSIZ	integer	si	no	Indica la posizione di un elemento in un vettore
RIEMP	integer	si	si	Indica il riempimento di un vettore
VETT	integer	si	si	Array monodimensionale

• Dichiarazione variabili locali al sottoprogramma:

Nome	Tipo	Attributo
I	integer	Aggiorna il valore corrente di un elemento di un array nei cicli "do"
J	integer	idem
POSMIN	integer	Aggiorna l'indice dell'elemento di un vettore di valore minimo

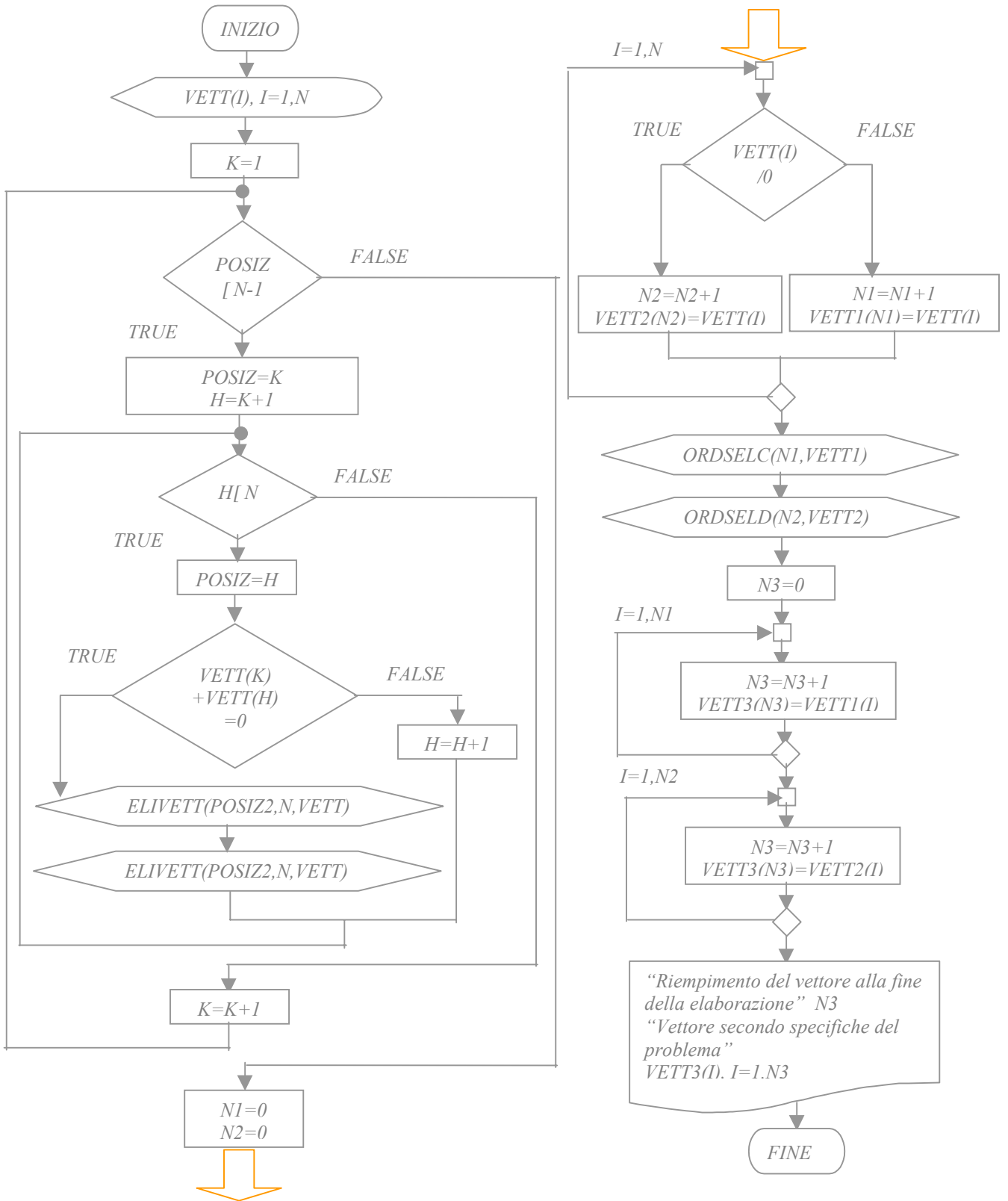
FLOW CHART
Sottoprogramma "ELIVET", Sottoprogramma "ORDSEL"



NOTA

Il Flow Chart del sottoprogramma "ORDSEED" è analogo a quello di "ORDSEL"; basta solo sostituire "posmin" con "posmax" e il segno di / con [

FLOW CHART
Programma "ELIVAL"



PROGRAM ELIVAL

```

c      Data : 20/12/99
c      Versione : 1.0
c      Autore: Ermanno Napoli
c      Motivazione globale :
c      assegnato un vettore di interi relativi
c      si riordini il vettore assegnato in modo
c      tale che i valori negativi precedano i
c      positivi ed i primi abbiano ordinamento
c      crescente, mentre i secondi siano ordinati
c      in senso decrescente. Si stampi il vettore
c      avendo eliminato le coppie eventuali di
c      valori di segno opposto e modulo uguale
c      Dichiarazione costanti
      INTEGER CARD
      PARAMETER(CARD=50)
c      Dichiarazione tipo
      INTEGER VETT(CARD),VETT1(CARD)
      INTEGER VETT2(CARD),VETT3(CARD)
c      Dichiarazioni variabili
      INTEGER H,K,POS1,POS2,N,N1,N2,N3
c      Dichiarazione sottoprogramma
      EXTERNAL ELIVET,ORSELC,ORSELD
c      Begin
c      Acquisizione dati in ingresso
      WRITE(*,*) 'Dammi il riempimento:'
      READ(*,'(I2)') N
      WRITE(*,*) 'Dammi il vettore:'
      READ(*,'(30I3)') (VETT(I), I=1,N)
c      Inizio ciclo "while..do": si cercano le coppie di
c      valori di segno opposto e modulo uguale
      K=1
100    CONTINUE
      IF(.NOT.(K.LE.N-1)) GOTO 200
      POS1=K
      H=K+1
c      Inizio ciclo "while..do": fissato un elemento
c      del vettore si ricerca tra i rimanenti il suo
c      opposto
300    CONTINUE
      IF(.NOT.(H.LE.N)) GOTO 400
      POS2=H
      IF(VETT(K)+VETT(H).EQ.0) THEN
c      Si eliminano gli elementi del vettore che sono
c      di segno opposto e modulo uguale
      CALL ELIVET(POS2,N,VETT)
      CALL ELIVET(POS1,N,VETT)
      H=N
      ELSE
      H=H+1
      ENDIF
      GOTO 300
400    CONTINUE
c      Fine ciclo "while..do"
      K=K+1
      GOTO 100
200    CONTINUE
c      Fine ciclo "while..do"

```

```

c      Si inseriscono i valori negativi e positivi
c      nei vettori vett1,vett2
c      Inizializzazione variabile riempimento vett1,vett2
      N1=0
      N2=0
      DO 10 I=1,N
      IF(VETT(I).GE.0) THEN
      N2=N2+1
      VETT2(N2)=VETT(I)
      ELSE
      N1=N1+1
      VETT1(N1)=VETT(I)
      ENDIF
10    CONTINUE
c      Si ordinano gli elementi negativi
c      in senso decrescente
      CALL ORSELC(N1,VETT1)
c      Si ordinano gli elementi positivi
c      in senso crescente
      CALL ORSELD(N2,VETT2)
c      Si inseriscono gli elementi dei vettori
c      vett1,vett2 in un unico vettore come
c      richiesto da specifica del problema
c      Inizializzazione variabile riempimento vett3
      N3=0
      DO 20 I=1,N1
      N3=N3+1
      VETT3(N3)=VETT1(I)
20    CONTINUE
      DO 30 I=1,N2
      N3=N3+1
      VETT3(N3)=VETT2(I)
30    CONTINUE
c      Stampa dei risultati
      WRITE(*,'(1X,A38,I2)')'Riempimento alla fine dell''algoritmo:',N3
      WRITE(*,*) 'Vettore secondo la specifica del problema'
      WRITE(*,'(1X,30I3)') (VETT3(I), I=1,N3)
      STOP
      END
c *****
      SUBROUTINE ELIVET(POSIZ,RIEMP,VETT)
      INTEGER VETT(*)
      INTEGER POSIZ,RIEMP,I
      DO 10 I=POSIZ+1,RIEMP
      VETT(I-1)=VETT(I)
10    CONTINUE
      RIEMP=RIEMP-1
      RETURN
      END

```

```

SUBROUTINE ORSELC (RIEMP,VETT)
INTEGER VETT(*)
INTEGER RIEMP,I,J, POSMIN,SALVA
C Begin
DO 10 I=1,RIEMP-1
  POSMIN=I
  DO 20 J=I+1,RIEMP
    IF (VETT (POSMIN) .GT.VETT (J)) THEN
      POSMIN=J
    ENDIF
  20 CONTINUE
  IF (POSMIN.NE.I) THEN
    SALVA=VETT (POSMIN)
    VETT (POSMIN)=VETT (I)
    VETT (I)=SALVA
  ENDIF
  10 CONTINUE
RETURN
END
C *****
SUBROUTINE ORSELD (RIEMP,VETT)
INTEGER VETT(*)
INTEGER RIEMP,I,J, POSMAX,SALVA
C Begin
DO 10 I=1,RIEMP-1
  POSMAX=I
  DO 20 J=I+1,RIEMP
    IF (VETT (POSMAX) .LT.VETT (J)) THEN
      POSMAX=J
    ENDIF
  20 CONTINUE
  IF (POSMAX.NE.I) THEN
    SALVA=VETT (POSMAX)
    VETT (POSMAX)=VETT (I)
    VETT (I)=SALVA
  ENDIF
  10 CONTINUE
RETURN
END

```

PROVA D'ESAME del 18 maggio 1999

Siano assegnati in ingresso, da file o da tastiera, le dimensioni M ed N ed i valori di una matrice di interi. Si scriva un programma che, per ogni coppia di righe consecutive della matrice, verifichi se esistono almeno due valori in comune tra le righe e, in caso positivo, le scambi. Si stampi, infine, la matrice modificata.

ESEMPIO

INPUT

M=5

N=6

```

1 7 5 8 2 6
3 2 9 8 4 10
20 2 3 18 7 13
4 6 2 5 1 8
5 8 22 7 30 14

```

OUTPUT

```

3 2 9 8 4 10
1 7 5 8 2 6
20 2 3 18 7 13
5 8 22 7 30 14
4 6 2 5 1 8

```

DESCRIZIONE ALGORITMO

Letti i numeri di righe e colonne fornite in input e completata la lettura dei singoli elementi costituenti la matrice, si effettua uno scorrimento della matrice per righe non mediante un ciclo “for”, ma utilizzando un costrutto iterativo del tipo “while....do”. Infatti il problema richiede che siano controllate ogni riga con la successiva (consecutiva) e quindi tale controllo deve essere terminato alla penultima riga. La verifica dei valori uguali per ogni coppia di righe considerate, si effettua con un costrutto di selezione “if...then”, mettendo bene in luce con un ciclo “for” il fatto che ogni elemento della generica riga deve essere confrontato con ogni elemento della riga consecutiva. Verificatasi la condizione posta dal problema essa è messa in luce nell’algoritmo mediante una variabile ‘contatore’ che indica se esistono almeno i due valori richiesti. Si effettuano lo scambio tra le righe e infine si stampano i risultati.

DOCUMENTAZIONE INTERNA AL PROGRAMMA

1- Costanti adoperate :

Nome	Valore	tipo	Attributo
cardr	20	intero	Dimensione array
Cardc	20	intero	Dimensione array

2- Tipi di dato definiti dall’utente:

Nome	Definizione	Descrizione
matrice	array(1..cardr,1..cardc)of integer	Insieme in cui effettuare la ricerca

3- Struttura dei dati:

• Informazioni di ingresso:

Nome	Tipo	Attributo
M	integer	Numero di righe della matrice in ingresso
N	integer	Numero di colonne della matrice in ingresso
Mat	matrice	Matrice di ingresso

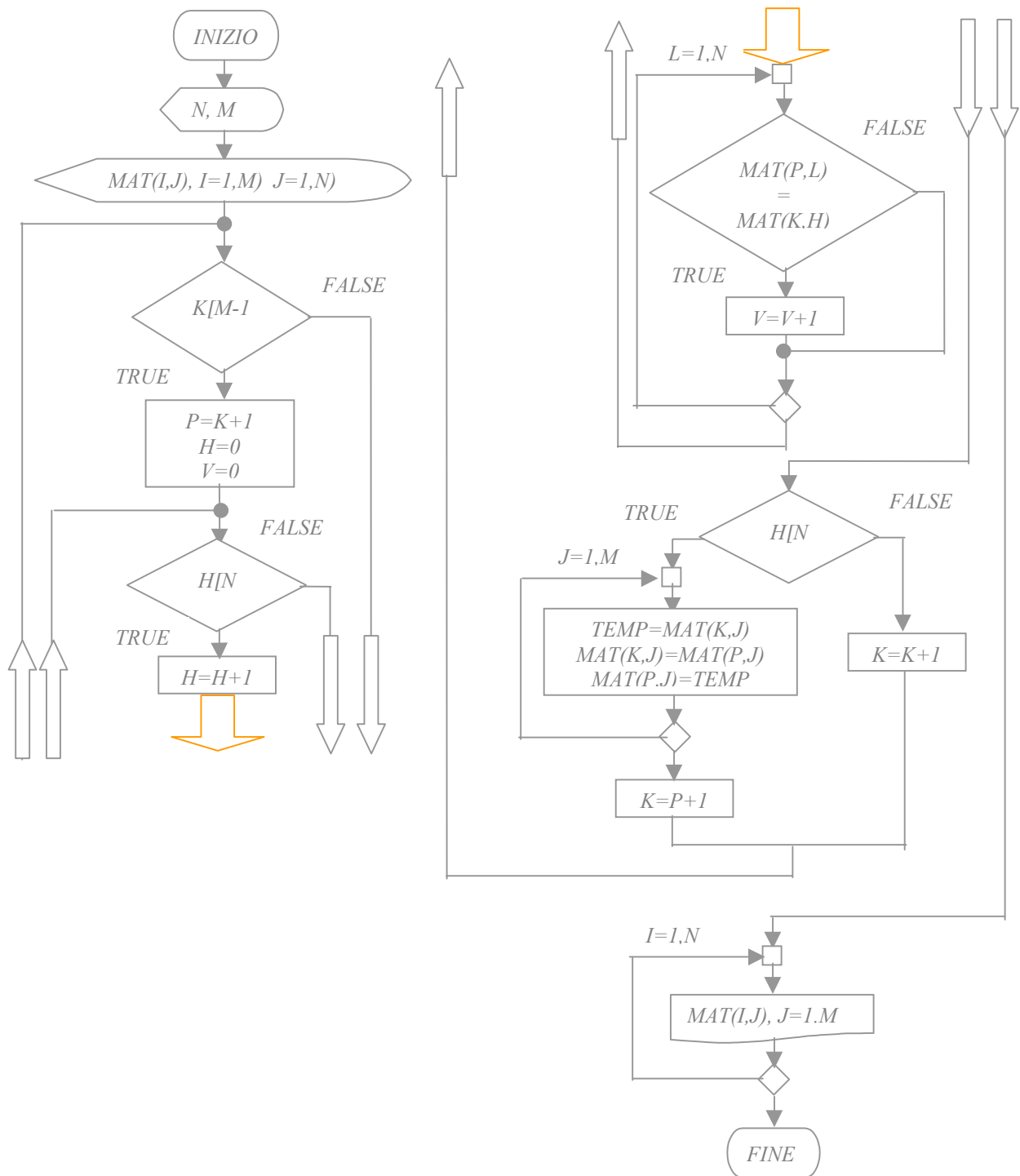
• Informazioni di uscita:

Nome	tipo	Attributo
M	integer	Numero di righe della matrice in uscita
N	integer	Numero di colonne della matrice in uscita
Mat	matrice	Matrice modificata secondo le specifiche del problema

• Informazioni di algoritmo:

Nome	Tipo	Attributo
K	integer	Indica il numero di riga corrente
P	integer	Indica la riga consecutiva alla riga corrente
H	integer	Indica il numero di colonna corrente
V	integer	Aggiorna il numero di volte per cui si hanno valori uguali
I	integer	Aggiorna i valori del ciclo “do”
J	integer	Idem
L	integer	idem

FLOW CHART
Programma "CONSRI"



CODIFICA FORTRAN

```

      program consri

c      Data : 20/12/99
c      Versione : 1.0
c      Autore: ERMANNO NAPOLI
c      Motivazione globale :
c      Data una matrice di N righe e M colonne
c      si verifichi se esistono almeno due valori
c      in comune tra le righe e in caso positivo,
c      le scambi.
c      Dichiarazione costanti
      INTEGER CARDR,CARDC
      PARAMETER (CARDR=20,CARDC=20)
c      Dichiarazione tipo
      INTEGER MAT(CARDR,CARDC)
c      Dichiarazione variabili
      INTEGER N,M,I,J,H,K,P,V,L,TEMP
c      Begin
c      Acquisizione dati in ingresso
      WRITE(*,*) 'DAMMI IL NUMERO DI RIGHE:'
      READ(*,'(I2)') N
      WRITE(*,*) 'DAMMI IL NUMERO DI COLONNE:'
      READ(*,'(I2)') M
      WRITE(*,*) 'DAMMI GLI ELEMENTI DELLA MATRICE PER RIGHE.'
      WRITE(*,*)
      DO 10 I=1,N
      READ(*,'(20I3)') (MAT(I,J),J=1,M)
10    continue
c      Inizio ciclo "while..do": controlla ogni riga con
c      la successiva
c      Inizializzazione indici riga "corrente"
c      e riga "consecutiva"
      K=1
      P=0
100   CONTINUE
      IF(K.LE.(N-1)) GOTO 200
      P=K+1
      H=0
      V=0
c      Inizio ciclo "while..do": cerca valori in comune
c      fra le righe
300   CONTINUE
      IF(.NOT.(H.LE.M)) GOTO 400
      H=H+1
      DO 20 L=1,N
      IF(MAT(P,L).EQ.MAT(K,H)) THEN
      V=V+1
      ENDIF
20    CONTINUE
      GOTO 300
400   continue
c      Fine "while..do"

```

```

C      Si controlla se il numero di elementi
C      uguali corrisponde alle specifiche del
C      problema
      IF (V.GE.2) THEN
      DO 30 J=1,M
      TEMP=MAT(K,J)
      MAT(K,J)=MAT(P,J)
      MAT(P,J)=TEMP
30    CONTINUE
      K=P+1
      ELSE
      K=K+1
      ENDIF
      GOTO 100
200   continue
c     Fine "while..do"
c     Stampa dei risultati
      DO 40 I=1,N
      WRITE(*,'(20I3)') (MAT(I,J),J=1,M)
40    CONTINUE
      STOP
      END

```

PROVA D'ESAME del 8 giugno 1999

Sia data in ingresso, da tastiera o da file, una matrice di caratteri aventi M righe ed N colonne ed un numero intero $B * 16$. Per ognuna delle righe, si verifichi se i caratteri ad essa appartenenti costituiscono una sequenza corretta di cifre in base B : in caso affermativo, si valuti il numero rappresentato e lo si fornisca in uscita; in caso contrario, si stampi l'indice di colonna del primo carattere che non rappresenta una cifra ammissibile nella base B .

Esempio:

	INGRESSO	USCITA
5	(M)	
4	(N)	
12	(B)	
	1744	2787
	2A57	4963
	935G	Cifra errata in colonna 4
	1B58	3380
	73FX	Cifra errata in colonna 3

DESCRIZIONE ALGORITMO

Fornita la base di numerazione $B * 16$ si fornisce in input anche l'insieme dei simboli rappresentativi del sistema di numerazione. Si fornisce poi il numero di righe e di colonne della matrice di caratteri. L'algoritmo consiste in uno scorrimento della matrice per righe mediante un costrutto iterativo "while..do". Per ogni riga si verifica se il carattere appartiene all'insieme di numerazione $[0,1,2,\dots]$ fissato inizialmente. Se si verifica tale appartenenza allora si calcola il numero nella base B , altrimenti si segnala riportando il numero di colonna, la non appartenenza del carattere all'insieme del sistema di numerazione.

DOCUMENTAZIONE INTERNA AL PROGRAMMA

1- Dichiarazione costanti:

Nome	Valore	Tipo	Attributo
card	16	Intero	Dimensione massima array monodimensionale
cardr	20	Intero	Una dimensione massima di array bidimensionale
cardc	20	Intero	Una dimensione massima di array bidimensionale
alfa	108	Intero	Indica il numero del carattere della tabella ASCII a partire dal quale si effettuano le elaborazioni in seno al programma

2- Tipi di dato definiti dall'utente:

Nome	Definizione	Descrizione
Matrice	array(1..cardr,1..cardc) di caratteri	Matrice contenente caratteri letterali e numerici

3- Struttura dei dati.

• Informazioni di ingresso:

Nome	Tipo	Attributo
B	integer	Indica la base di numerazione prescelta
M	integer	Indica il numero di colonne della matrice in ingresso
N	integer	Indica il numero di righe della matrice in ingresso
MAT	matrice	Contiene i caratteri su cui si deve eseguire l'elaborazione

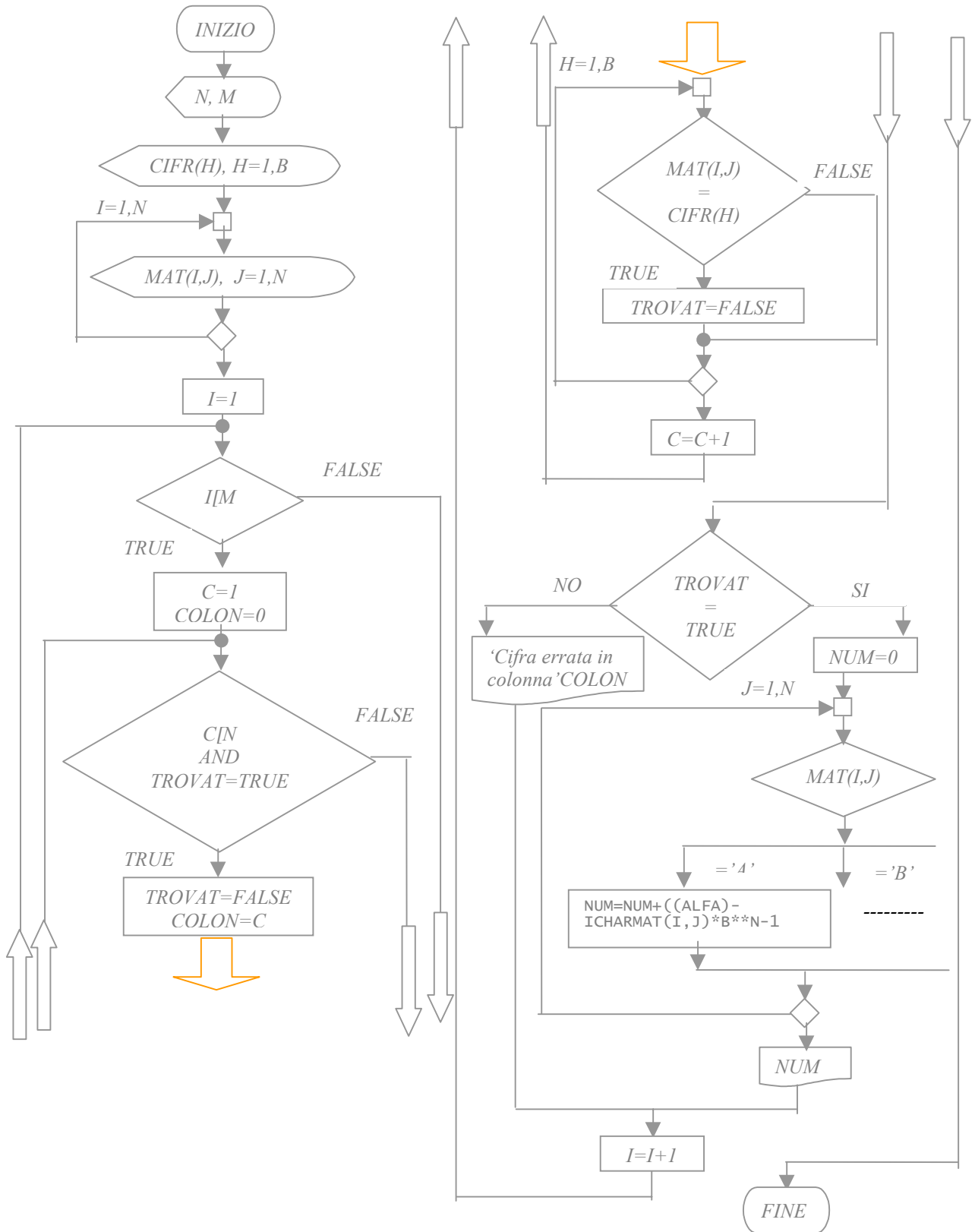
• Informazioni di uscita:

Nome	Tipo	Attributo
NUM	Integer	Numero calcolato in base decimale
COLON	Integer	Indice di colonna contenente cifra non ammissibile nella base B

• Informazione di algoritmo:

Nome	Tipo	Attributo
C	Integer	Contiene indice di colonna
I	Integer	Aggiorna l'indice di riga
J	Integer	Aggiorna l'indice di colonna
H	integer	Indice di scorrimento del vettore base numerica
TROVAT	boolean	Indica se il carattere corrente appartiene all'insieme di numerazione B

FLOW CHART
Programma "CALCIF"



```

program calcif

c   Data : 20/12/99
c   Versione : 1.0
c   Autore: ERMANNO NAPOLI
c   Motivazione globale:
c   data una matrice M,N di caratteri e un numero intero B<=16
c   per ognuna delle righe si verifichi che costituiscono una
c   sequenza corretta di cifre in base T. In caso affermativo
c   si valuti il numero ,in caso contrario si stampi l'indice
c   di colonna
c   Dichiarazione costanti
      INTEGER CARD,CARDR,CARDC,ALFA
      PARAMETER (CARD=16,CARDR=20,CARDC=20,ALFA=108)
c   Dichiarazione tipi
      CHARACTER MAT(CARDR,CARDC)
      CHARACTER CIFR(CARD)
c   Dichiarazione variabili
      INTEGER M,N,B,I,J, COLON,C,NUM,H
      LOGICAL TROVAT
c   Acquisizione dati in ingresso
      WRITE(*,*) 'dammi la base di numerazione minore di 16:'
      READ(*,1) B
      WRITE(*,*) 'dammi B caratteri per il sistema di numerazione:'
      READ(*,2) (CIFR(H),H=1,B)
      WRITE(*,*) 'DAMMI IL NUMERO DI RIGHE:'
      READ(*,3) M
      WRITE(*,*) 'DAMMI IL NUMERO DI COLONNE:'
      READ(*,4) N
      WRITE(*,*) 'DAMMI LA MATRICE DI CARATTERI PER RIGHE CON LETTERE MI
+NUSCOLE:'
      WRITE(*,*)
      DO 10 I=1,M
      READ(*,5) (MAT(I,J),J=1,N)
10  CONTINUE
c   Begin
      I=1
c   Inizio "while..do"
100  CONTINUE
      IF(.NOT.(I.LE.M)) GOTO 200
      C=1
      COLON=0
      TROVAT=.TRUE.
c   Inizio "while..do"
300  CONTINUE
      IF(.NOT.((C.LE.N).AND.(TROVAT))) GOTO 400
      TROVAT=.FALSE.
      COLON=C
      DO 20 H=1,B
          IF(MAT(I,C).EQ.CIFR(H)) THEN
              TROVAT=.TRUE.
          ENDIF
20  CONTINUE
      C=C+1
      GOTO 300
400  CONTINUE
c   Fine "while..do"

```

```

IF (TROVAT) THEN
NUM=0
DO 30 J=1,N
IF (MAT(I,J).EQ.'A') THEN
  NUM=NUM+((ALFA)-ICCHAR(MAT(I,J)))*B**(J-1)
ELSEIF (MAT(I,J).EQ.'B') THEN
  NUM=NUM+((ALFA+2)-ICCHAR(MAT(I,J)))*B**(J-1)
ELSEIF (MAT(I,J).EQ.'C') THEN
  NUM=NUM+((ALFA+3)-ICCHAR(MAT(I,J)))*B**(J-1)
ELSEIF (MAT(I,J).EQ.'D') THEN
  NUM=NUM+((ALFA+4)-ICCHAR(MAT(I,J)))*B**(J-1)
ELSEIF (MAT(I,J).EQ.'E') THEN
  NUM=NUM+((ALFA+5)-ICCHAR(MAT(I,J)))*B**(J-1)
ELSEIF (MAT(I,J).EQ.'F') THEN
  NUM=NUM+((ALFA+6)-ICCHAR(MAT(I,J)))*B**(J-1)
ELSEIF (MAT(I,J).EQ.'0') THEN
  NUM=NUM+(48-ICCHAR(MAT(I,J)))*B**(J-1)
ELSEIF (MAT(I,J).EQ.'1') THEN
  NUM=NUM+(50-ICCHAR(MAT(I,J)))*B**(J-1)
ELSEIF (MAT(I,J).EQ.'2') THEN
  NUM=NUM+(52-ICCHAR(MAT(I,J)))*B**(J-1)
ELSEIF (MAT(I,J).EQ.'3') THEN
  NUM=NUM+(54-ICCHAR(MAT(I,J)))*B**(J-1)
ELSEIF (MAT(I,J).EQ.'4') THEN
  NUM=NUM+(56-ICCHAR(MAT(I,J)))*B**(J-1)
ELSEIF (MAT(I,J).EQ.'5') THEN
  NUM=NUM+(58-ICCHAR(MAT(I,J)))*B**(J-1)
ELSEIF (MAT(I,J).EQ.'6') THEN
  NUM=NUM+(60-ICCHAR(MAT(I,J)))*B**(J-1)
ELSEIF (MAT(I,J).EQ.'7') THEN
  NUM=NUM+(62-ICCHAR(MAT(I,J)))*B**(J-1)
ELSEIF (MAT(I,J).EQ.'8') THEN
  NUM=NUM+(64-ICCHAR(MAT(I,J)))*B**(J-1)
ELSEIF (MAT(I,J).EQ.'9') THEN
  NUM=NUM+(66-ICCHAR(MAT(I,J)))*B**(J-1)
ENDIF
30 CONTINUE
c   Stampa dei risultati
  WRITE(*,6) NUM
  ELSE
c   Stampa dei risultati
  WRITE(*,7) 'CIFRA ERRATA IN COLONNA:',COLON
ENDIF
  I=I+1
  GOTO 100
200 continue
c   fine "while..do"
1  FORMAT(I2)
2  FORMAT(16A2)
3  FORMAT(I3)
4  FORMAT(I3)
5  FORMAT(20A2)
6  FORMAT(1X,I20)
7  FORMAT(1X,A26,2X,I2)
  STOP
  END

```

PROVA D'ESAME del 22 giugno 1999

Sia assegnata in ingresso, da file o da tastiera, una sequenza di valori interi terminata dal valore "0". Si scriva un programma che modifichi la sequenza sostituendo, ad ogni terna di elementi consecutivi ABC tali che $A < B$ e $C < B$, la terna CBA senza che gli elementi di quest'ultima siano considerati nella successiva analisi. Si stampi inoltre la terna che, tra le terne fatte oggetto di scambio, sia formata dagli elementi la cui somma è massima.

Esempio

INPUT

1 5 7 13 5 2 8 22 10 12 4 0

OUTPUT

1 5 5 13 7 2 10 22 8 12 4 0

Terna con somma degli elementi massima : 8 22 10

ANALISI ALGORITMO

Per facilitare le operazioni di ricerca e di scambio si inseriscono gli elementi della sequenza in un vettore. Tramite un ciclo iterativo "while ...do" si inizia la ricerca della terna considerando ogni elemento i -esimo del vettore. La ricerca ha termine quando i è minore o uguale a $\text{riemp}-3$. Per ogni terna fatto oggetto di scambio si calcola la somma degli elementi costituenti e si confronta tale somma con un valore sommax precedentemente calcolato. Infine si stampano i risultati.

DOCUMENTAZIONE INTERNA AL PROGRAMMA

1-Costanti adoperate:

Nome	Valore	Tipo	Attributo
CARD	20	intero	Dimensione massima del array monodimensionale

2-Tipi di dato definiti dall'utente:

Nome	Definizione	Descrizione
VETTORE	Array(1..card) di interi	Deve contenere gli elementi su cui effettuare la ricerca

3-Struttura dei dati:

•Informazioni di ingresso:

Nome	Tipo	Attributo
SEQ	String[card]	Sequenza di valori interi terminata da zero

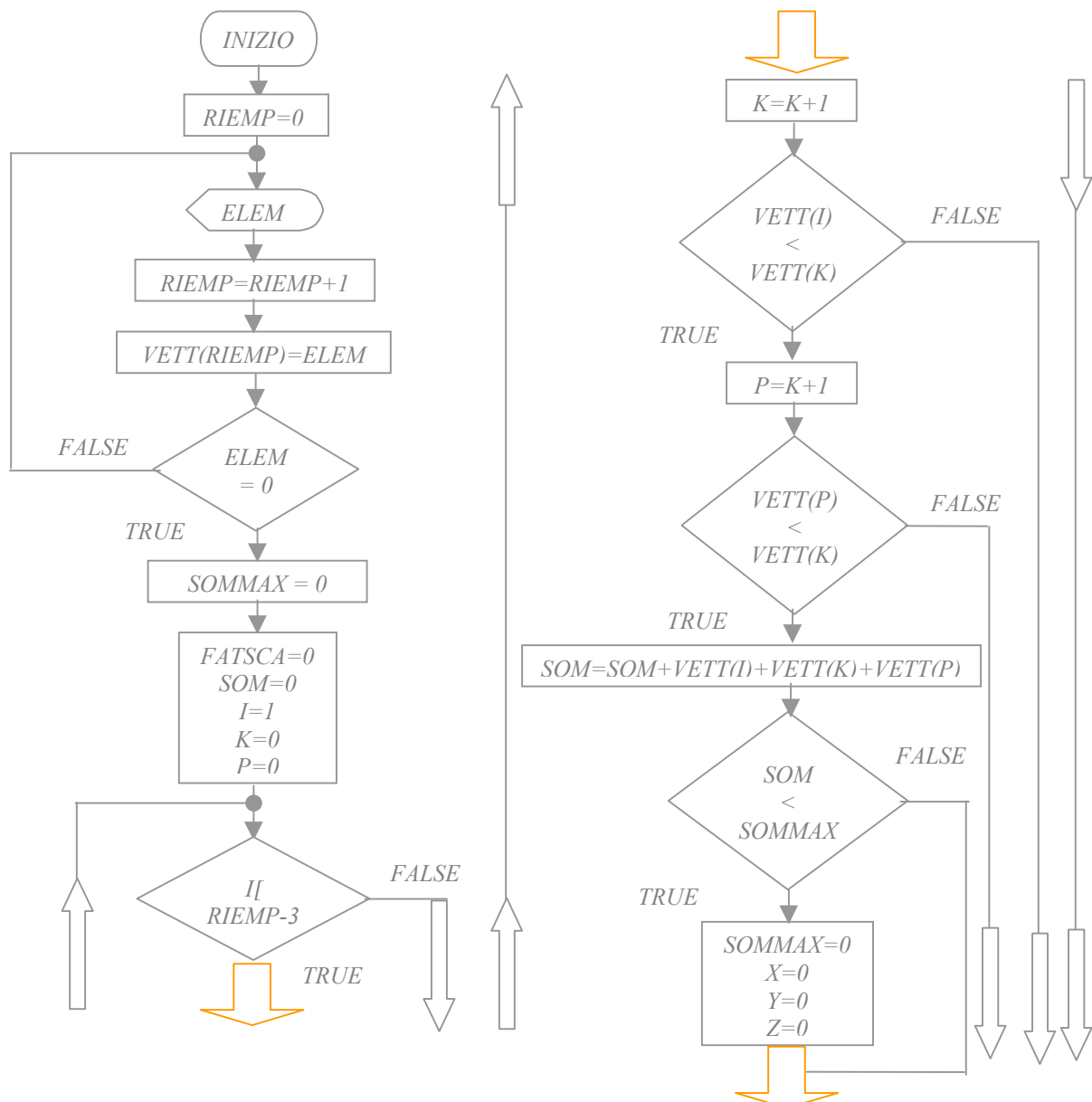
•Informazioni di uscita:

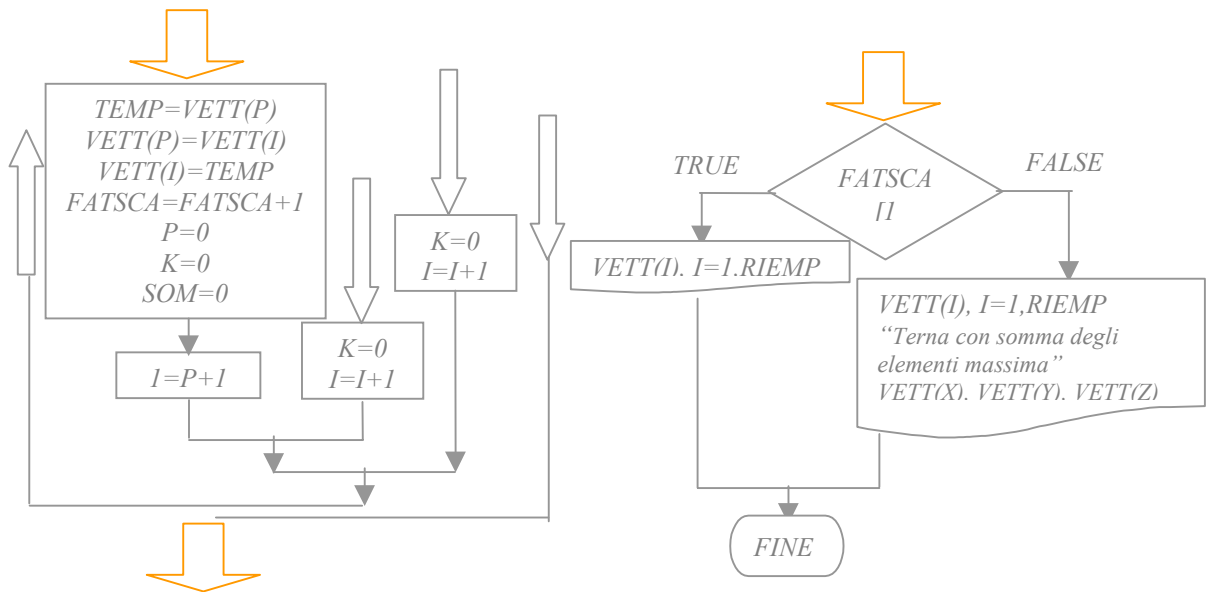
Nome	Tipo	Attributo
VETT	vettore	Deve contenere la sequenza di interi fornita in input

•Informazioni di algoritmo:

Nome	Tipo	Attributo
I	integer	Indica l'elemento corrente del vettore
K	integer	Indica l'elemento successivo dell'elemento corrente
P	integer	Indica il successivo di k
RIEMP	integer	Indica il riempimento del vettore
SOM	integer	Indica la somma degli elementi della terna fatta oggetto di scambio
SOMMAX	integer	Indica il valore massimo di som ottenuto
ELEM	integer	Indica l'elemento corrente della sequenza
Y	integer	Indica la posizione degli elementi del vettore per cui è som=max
X	integer	“ “ “
Z	integer	“ “ “
TEMP	integer	Variabile di appoggio
FATSCA	integer	Assume il valore del numero di scambi

FLOW CHART
Programma “TERNA”





CODIFICA FORTRAN

```

PROGRAM TERNA
c Data una sequenza di valori interi terminata da zero
c si scriva un programma che modifichi la sequenza so_
cstituendo, da ogni terna di elementi consecutivi ABC
c tale che A<B e C<B, la terna CBA, senza che gli ele_
cmenti di quest'ultima siano considerati nella succes_
c siva analisi. Si stampi, inoltre, la terna che, tra le
cterne fatte oggetto di scambio, sia formata dagli
c elementi la cui somma massima.
c Autore: ERMANN0 NAPOLI
c Dichiarazioni costanti
INTEGER CARD
PARAMETER (CARD=20)
c Dichiarazione tipi
INTEGER VETT(CARD)
c Dichiarazione variabili
INTEGER I,K,P,RIEMP,SOM,SOMMAX,X,Y,Z,ELEM,TEMP,FATSCA
c Begin
c Acquisizione dati in ingresso
WRITE(*,*) 'Dammi sequenza di numeri interi terminata da''0''':
WRITE(*,*)
RIEMP=0
c Inizio "repeat..until": inserimento degli elementi
c della sequenza nel vettore
100 CONTINUE
READ(*,'(I3)') ELEM
RIEMP=RIEMP+1
VETT(RIEMP)=ELEM
IF(.NOT.(ELEM.EQ.0)) GOTO 100
c Fine "repeat..until"
c Inizializzazione variabili di algoritmo
I=1
K=0
P=0
SOM=0
SOMMAX=0
FATSCA=0

```

```

c      Inizio "while..do"
200   CONTINUE
      IF(.NOT.(I.LE.RIEMP-3)) GOTO 300
      K=I+1
      IF(VETT(I).LT.VETT(K)) THEN
      P=K+1
      IF(VETT(P).LT.VETT(K)) THEN
      SOM=SOM+VETT(I)+VETT(K)+VETT(P)
      IF(SOM.GT.SOMMAX) THEN
      SOMMAX=SOM
      X=I
      Y=K
      Z=P
      ENDIF
      TEMP=VETT(P)
      VETT(P)=VETT(I)
      VETT(I)=TEMP
      FATSCA=FATSCA+1
      I=P+1
      K=0
      P=0
      SOM=0
      ELSE
      K=0
      I=I+1
      ENDIF
      ELSE
      K=0
      I=I+1
      ENDIF
      GOTO 200
300   continue
c      Fine "while..do"
      IF(FATSCA.LE.1) THEN
c      Stampa dei risultati
      WRITE(*,'(1X,20I3)') (VETT(I), I=1,RIEMP)
      ELSE
c      Stampa dei risultati
      WRITE(*,'(1X,20I3)') (VETT(I), I=1,RIEMP)
      WRITE(*,'(1X,''TERNA CON SOMMA DEGLI ELEMENTI MASSIMA:''',2X,
+ (I3,2X,I3,2X,I3)') VETT(X),VETT(Y),VETT(Z)
      ENDIF
      STOP
      END

```

PROVA D'ESAME del 13 luglio 1999

Siano date in ingresso, da file o da tastiera, le dimensioni di una matrice di interi, i cui elementi siano uguali, ad uno stesso valore, detto valore dominante. Gli elementi della matrice siano assegnati fornendo in ingresso, per ogni elemento, l'indice di riga, l'indice di colonna ed il valore dell'elemento. La successione di elementi in ingresso è terminata da una terna i cui indici di riga e di colonna sono nulli, mentre il valore corrispondente indica il valore dominante. Si scriva un programma che stampi la matrice fornita in ingresso e l'indice della prima colonna formata interamente da elementi con valore dominante.

ESEMPIO

INPUT

```
nrig=4
ncol=5
posizione e valore elementi matrice
1 3 12
2 5 3
2 1 14
4 1 9
3 5 12
2 3 9
4 3 11
0 0 7
```

OUTPUT

```
matrice
7 7 12 7 7
14 7 9 7 3
7 7 7 7 12
9 7 11 7 7

colonna con soli elementi
dominanti: 2
```

DESCRIZIONE ALGORITMO

L'algoritmo risolvete è semplice: si crea una matrice avente numero di righe pari a nrig e un numero di colonne pari a ncol e costituita da tutti elementi di valore nullo. Si fornisce poi in input la terna di valori indicante rispettivamente il numero di riga, di colonna e l'elemento da inserire nella matrice suddetta. Tale acquisizione si effettua con un ciclo "while..do" che pone come condizione di terminazione il fatto che siano uguali a zero i primi due valori della terna in input. A questo punto si completa la costruzione della matrice facendo assumere agli elementi ancora uguale a zero il valore dell'elemento (detto "dominante") per il quale gli indici di riga e di colonna dati in input sono nulli. La ricerca della colonna con soli elementi dominanti si esegue con uno scorrimento della matrice per colonne. Infine si stampano i risultati.

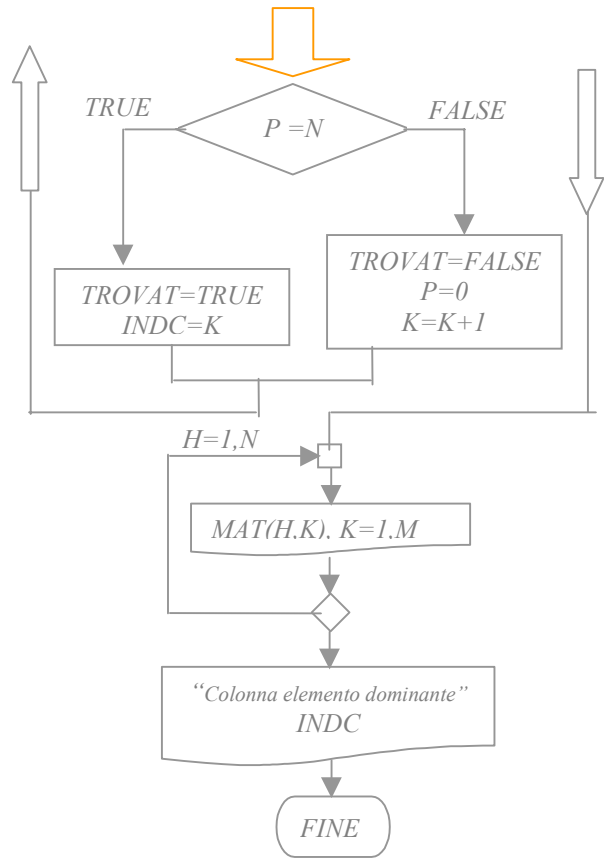
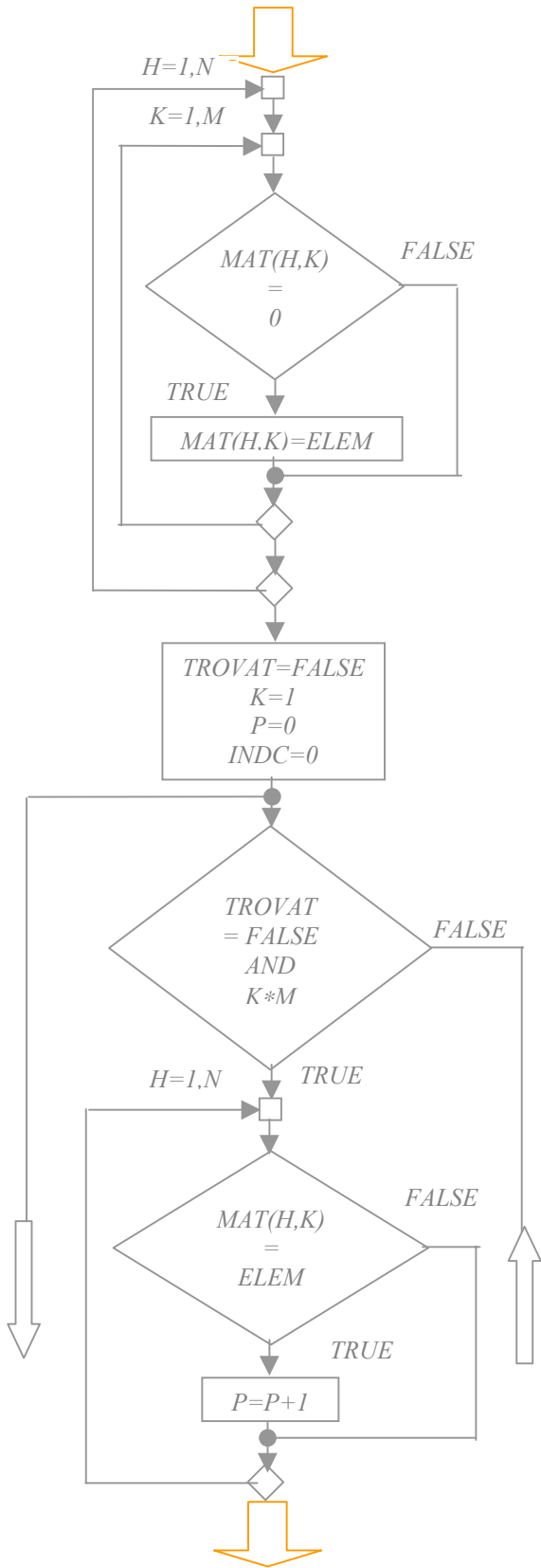
DOCUMENTAZIONE INTERNA AL PROGRAMMA

1-Costanti adoperate:

Nome	Valore	Tipo	Attributo
CARDR	50	intero	Una delle dimensioni massime di array
CARDC	50	intero	Idem

2-Tipi di dato definiti dall'utente:

Nome	Definizione	Descrizione
matrice	Array(1..cardr,1..cardc)of integer	Contiene i valori numerici su cui si basa l'elaborazione



```

PROGRAM MATDOM
c   Inserire in una matrice di n righe ed m colonne
c   di interi un certo numero di elementi terminati
c   da un elemento dominante
c   Autore:ERMANN0 NAPOLI
c   Dichiarazione costanti
      INTEGER CARDR,CARDC
      PARAMETER(CARDR=50,CARDC=50)
c   Dichiarazione tipi
      INTEGER MAT(CARDR,CARDC)
c   Dichiarazione variabili
      INTEGER N,M,H,K,INDC,S,T,ELEM,P
      LOGICAL TROVAT
c   begin
c   Acquisizione dati in ingresso
      WRITE(*,*) 'DAMMI IL NUMERO DI RIGHE:'
      WRITE(*,*)
      READ(*,1) N
      WRITE(*,*) 'DAMMI IL NUMERO DI COLONNE:'
      WRITE(*,*)
      READ(*,1) M
C   Inizializzazione matrice di uscita
      DO 10 H=1,N
      DO 20 K=1,M
      MAT(H,K)=0
20  CONTINUE
10  CONTINUE
      WRITE(*,*) 'Dammi l''indice di riga e colonna e elem da inserire:'
      WRITE(*,*)
      READ(*,2) S,T,ELEM
c   Inizio "while..do":fino a quando non si verifica
c   che gli indici di riga e di colonna sono nulli
c   ripeti le seguenti operazioni
100  continue
      if((s.eq.0).and.(t.eq.0)) goto 200
      mat(s,t)=elem
      write(*,*) 'dammi l''indice di riga e colonna e elem da inserire:'
      write(*,*)
      read(*,2) s,t,elem
      goto 100
200  continue
c   Fine "while..do"
c   Inserimento degli elementi forniti in input
c   nella matrice di uscita
      do 30 h=1,n
      do 40 k=1,m
      if(mat(h,k).eq.0) then
      mat(h,k)=elem
      endif
40  continue
30  continue
c   Inizio "while..do": si ricerca colonna costituita
c   da tutti elementi dominanti
      trovat=.false.
      k=1
      indc=0
      p=0
300  continue
      if(.not.((.not.trovat).and.(k.le.m))) goto 400
      do 50 h=1,n
      if(mat(h,k).eq.elem) then
      p=p+1
      endif
50  continue

```

```

    if(p.eq.n) then
        trovat=.true.
        indc=k
    else
        trovat=.false.
        p=0
        k=k+1
    endif
    goto 300
400 continue
c   Fine "while..do"
c   Stampa dei risultati
    do 60 h=1,n
        write(*,3) (mat(h,k), k=1,m)
60  continue
    write(*,'(1x,''colonna elemento dominante:'',i3)') indc
1   format(i2)
2   format(i2,1x,i2,1x,i3)
3   format(1x,20i3)
    stop
end

```

PROVA D'ESAME del 4 ottobre 1999

Siano dati in ingresso, da file o da tastiera, i riempimenti di riga e di colonna di una matrice di interi. Scrivere un programma che, per ogni elemento della matrice, verifichi se esiste almeno un altro elemento uguale al primo e posizionato sulla stessa riga o sulla stessa colonna e, in caso positivo, fornisca in uscita le coordinate del primo elemento, il suo valore e la distanza dall'elemento uguale più vicino fra quelli individuati. Si strutturi il programma in opportuni sottoprogrammi.

ESEMPIO

Ingresso

```

4 (righe)
5 (colonne)
2  12  13  7  6
8  8  4  11 2
1  15  21 3  9
18 8  4  3  23

```

Uscita

Riga	Colonna	Valore	Distanza
2	1	8	1
2	2	8	1
2	3	4	2
3	4	3	1
4	2	8	2
4	3	4	2
4	4	3	1

DESCRIZIONE ALGORITMO

Si effettua uno scorrimento della matrice per righe in modo da considerare ogni elemento consecutivamente. In questa maniera se ci sono elementi uguali ma situati su righe e colonne differenti vengono conteggiati una sola volta. Per questo elemento si esegue poi una ricerca sulla riga e sulla colonna di appartenenza; nel senso che viene messo a raffronto con tutti gli elementi rimanenti della sua riga e delle sua colonna. La ricerca ha termine quando si trova un elemento uguale o su una riga o su una colonna. I risultati vengono raccolti in appositi vettori e stampati secondo il formato di output richiesto.

NOTA.

L'algoritmo risolvete non è organizzato in sottoprogrammi. Per come è stato elaborato si presta a tale organizzazione in diverse maniere. Per esercizio si provi a strutturarlo in sottoprogrammi come richiesto dal problema, fornendo anche le relative tabelle.

DOCUMENTAZIONE INTERNA AL PROGRAMMA

1-Costanti adoperate:

Nome	Valore	Tipo	Attributo
CARDR	50	intero	Una delle dimensioni massime di un array
CARDC	50	intero	Idem

2- Tipi di dato definiti dall'utente:

Nome	Definizione	Descrizione
Vettore	array(1..cardr) of integer	Deve contenere numeri interi
matrice	array(1..cardr,1..cardc)of integer	Idem

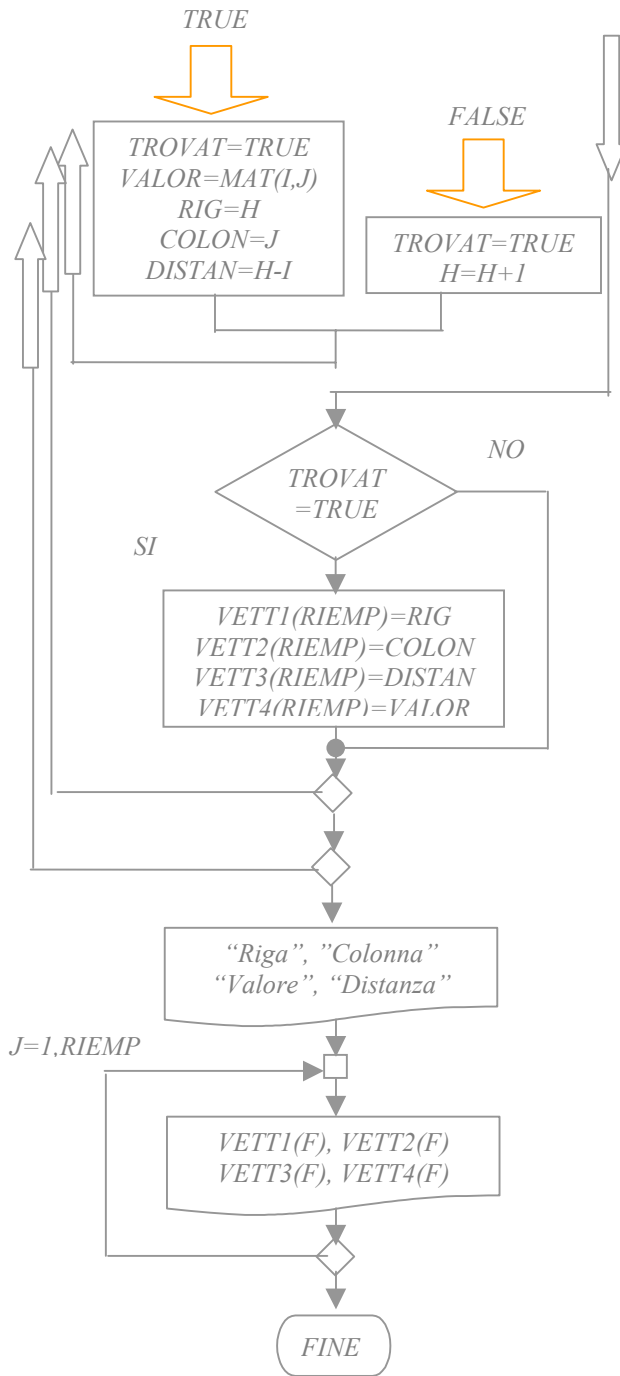
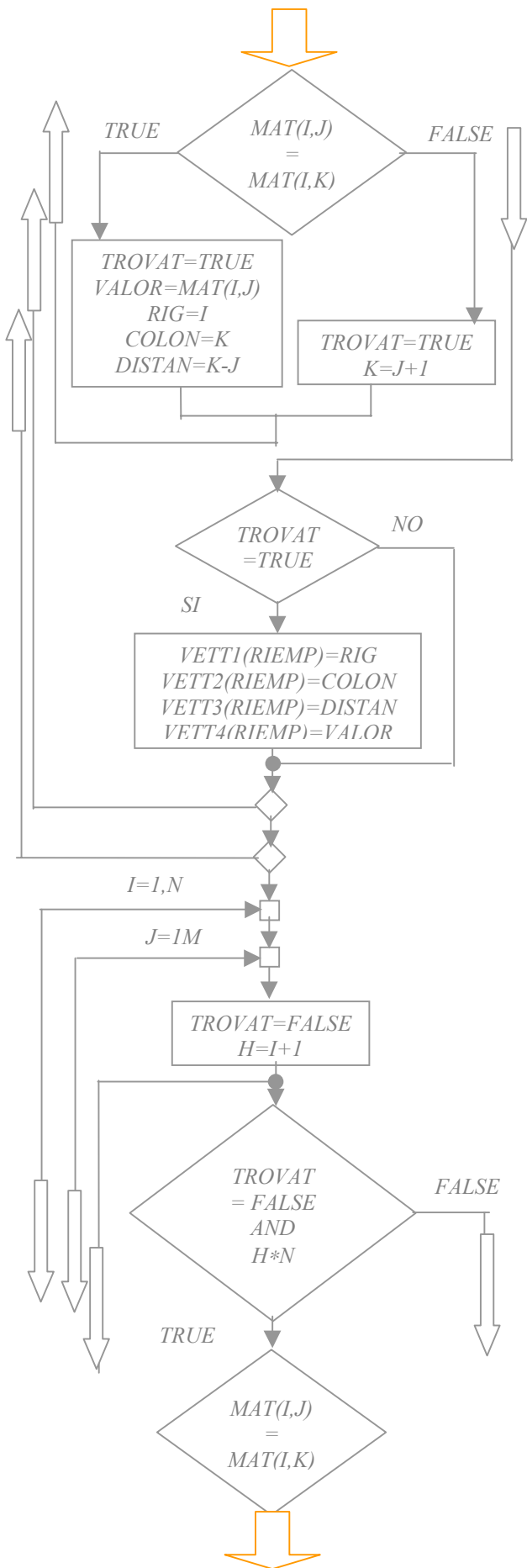
3-Struttura dei dati:

• Informazioni in ingresso:

Nome	Tipo	Attributo
N	integer	Numero di righe della matrice in input
M	integer	Numero di colonne della matrice in input
MAT	matrice	Elemento generico della matrice in input

• Informazioni di uscita:

Nome	Tipo	Attributo
VETT1	vettore	Contiene gli indici di riga
VETT2	vettore	Contiene gli indici di colonna
VETT3	vettore	Contiene il valore distanza valutato come da problema
VETT4	vettore	Contiene l'elemento della matrice in input che soddisfa le specifiche del problema



CODIFICA FORTRAN

```

PROGRAM MATDIS
C Data una matrice NxM si verifichi se per ogni
C elemento della matrice esiste uno uguale sulla
C stessa riga e sulla stessa colonna; in caso
C positivo si fornisca l'indice di colonna di
C riga la distanza e il valore
C Autore:ERMANN0 NAPOLI
C Dichiarazione costanti
INTEGER CARDR,CARDC
PARAMETER (CARDR=50,CARDC=50)
C Dichiarazione tipo
INTEGER VETT1(CARDR),VETT2(CARDR)
INTEGER VETT3(CARDR),VETT4(CARDR)
INTEGER MAT(CARDR,CARDC)
C Dichiarazione variabili
INTEGER I,J,K,n,m,riemp,valor
integer colon,distan,h,f,rig
logical trovat
c Dichiarazione sottoprogramma
c NESSUNO
c Begin
c Acquisizione dati in ingresso
write(*,*)'Dammi il numero di righe'
write(*,*)
read(*,'(i2)') n
write(*,*)'Dammi il numero di colonne'
write(*,*)
read(*,'(i2)') m
write(*,*)'Dammi la matrice'
write(*,*)
do 10 i=1,n
read(*,'(30i3)') (mat(i,j), j=1,m)
10 continue
c Inizializzazione riempimento
c vettori VETT1,VETT2,VETT3,VETT4
riemp=0
c Per ogni elemento della MAT si eseguono
c le operazioni seguenti
do 20 i=1,n
do 30 j=1,m
c Inizio "while..do": si controllano gli elementi
situati sulla stessa riga
trovat=.false.
k=j+1
100 continue
if(.not.((k.le.m).and.(.not.trovat))) goto 200
if (mat(i,j).eq.mat(i,k)) then
rig=i
colon=k
distan=k-j
valor=mat(i,j)
trovat=.true.
else
trovat=.false.
k=k+1
endif
goto 100
200 continue
c Fine "while..do"

```

```

        if(trovat) then
c      E' stato trovato un elemento che ha
c      le caratteristiche espresse dal problema
c      allora si eseguono le seguenti operazioni
        riemp=riemp+1
        vett1(riemp)=rig
        vett2(riemp)=colon
        vett3(riemp)=distan
        vett4(riemp)=valor
        endif
30    continue
20    continue
c      Per ogni elemento della MAT si eseguono
c      le operazioni seguenti
        do 40 i=1,n
        do 50 j=1,m
c      Inizio "while..do": si controllano gli elementi
        situati sulla stessa colonna
        trovat=.false.
        h=i+1
300   continue
        if(.not.(h.le.n).and.(.not.trovat)) goto 400
        if (mat(i,j).eq.mat(h,j)) then
            rig=h
            colon=j
            distan=h-i
            valor=mat(i,j)
            trovat=.true.
        else
            trovat=.false.
            h=h+1
        endif
        goto 300
400   continue
c      Fine "while..do"
        if(trovat) then
c      E' stato trovato un elemento che ha
c      le caratteristiche espresse dal problema
c      allora si eseguono le seguenti operazioni
        riemp=riemp+1
        vett1(riemp)=rig
        vett2(riemp)=colon
        vett3(riemp)=distan
        vett4(riemp)=valor
        endif
50    continue
40    continue
c      Stampa dei risultati
        write(*,1)
1      format(1x,'riga',5x,'colonna',5x,'valore',5x,'distanza')
        do 60 f=1,riemp
            write(*,2) vett1(f),vett2(f),vett4(f),vett3(f)
2      format(1x,I2,9x,I2,10x,I2,10x,I2)
60    continue
        stop
        end

```

PROVA D'ESAME del 7 novembre 1999

Siano dati in ingresso (da tastiera o da file) due matrici MAT1 e MAT2 di interi, aventi lo stesso numero M di colonne e rispettivamente N1 ed N2 righe, con $N2 > N1$. Si scriva un programma che:

- elimini da ogni colonna di MAT2 il sottovettore di lunghezza N1 tale che la differenza fra la somma degli elementi del sottovettore e la somma degli elementi della corrispondente colonna di MAT1 risulti massima in valore assoluto. In caso di differenze uguali, si elimini solo il primo dei sottovettori individuati.
- Produca in uscita la matrice MAT2 modificata.

ESEMPIO

Ingresso		Uscita
MAT1	MAT2	MAT2
2 4 6	6 2 1	6 2 4
5 2 7	3 3 2	3 3 6
4 5 3	1 1 3	4 2 2
3 8 2	3 5 2	1 7 1
4 2 4	5 6 2	
2 3 6		
4 2 2		
1 7 1		
5 6 2		

DESCRIZIONE ALGORITMO

Per facilitare le operazioni di ricerca del sottovettore si memorizzano in un apposito vettore (SOM1) la somma degli elementi delle colonne della MAT1. Poi si effettua uno scorrimento per colonne della MAT2. Per ognuna delle sue colonne si calcola la somma di N1 elementi, iniziando per ogni colonna dall'elemento in prima posizione fino all'elemento con indice di riga pari a $N2 - N1$. Ogni somma viene memorizzata in un secondo vettore (SOM2). Si esegue la differenza in valore assoluto tra le componenti del secondo vettore (espressione della somma di tutti i sottovettori di lunghezza N1 individuati nelle colonne di MAT2) e la componente del primo vettore con indice di posizione pari all'indice di colonna corrente di MAT2. Queste differenze si memorizzano in un terzo vettore (SOM3). Si ricerca il massimo del terzo vettore secondo le specifiche del problema. Si osserva, inoltre, che per come sono stati costruiti i vettori, l'indice di posizione del generico elemento del terzo vettore coincide con l'indice di riga della colonna corrente a partire dalla quale si considera il sottovettore di lunghezza N1. L'eliminazione poi del sottovettore si effettua con un algoritmo noto.

DOCUMENTAZIONE INTERNA AL PROGRAMMA

1- Costanti adoperate:

Nome	Valore	Tipo	Attributo
CARD	50	integer	Dimensione massima di un array
CARDR	50	integer	Dimensione massima di un array
CARDC	50	integer	Dimensione massima di un array

2-Tipi di dato definiti dall'utente:

Nome	Definizione	Descrizione
vettore	array(1..card) of integer	Contiene numeri interi
matrice	array(1..cardr,1..cardc)of integer	Deve contenere numeri interi

3-Struttura dei dati:

• Informazioni di ingresso:

Nome	Tipo	Attributo
M	integer	Numero di colonne di entrambe le matrici di input
N1	integer	Numero di righe della prima matrice
MAT1	matrice	Prima matrice
N2	integer	Numero di colonne della seconda matrice
MAT2	matrice	Seconda matrice

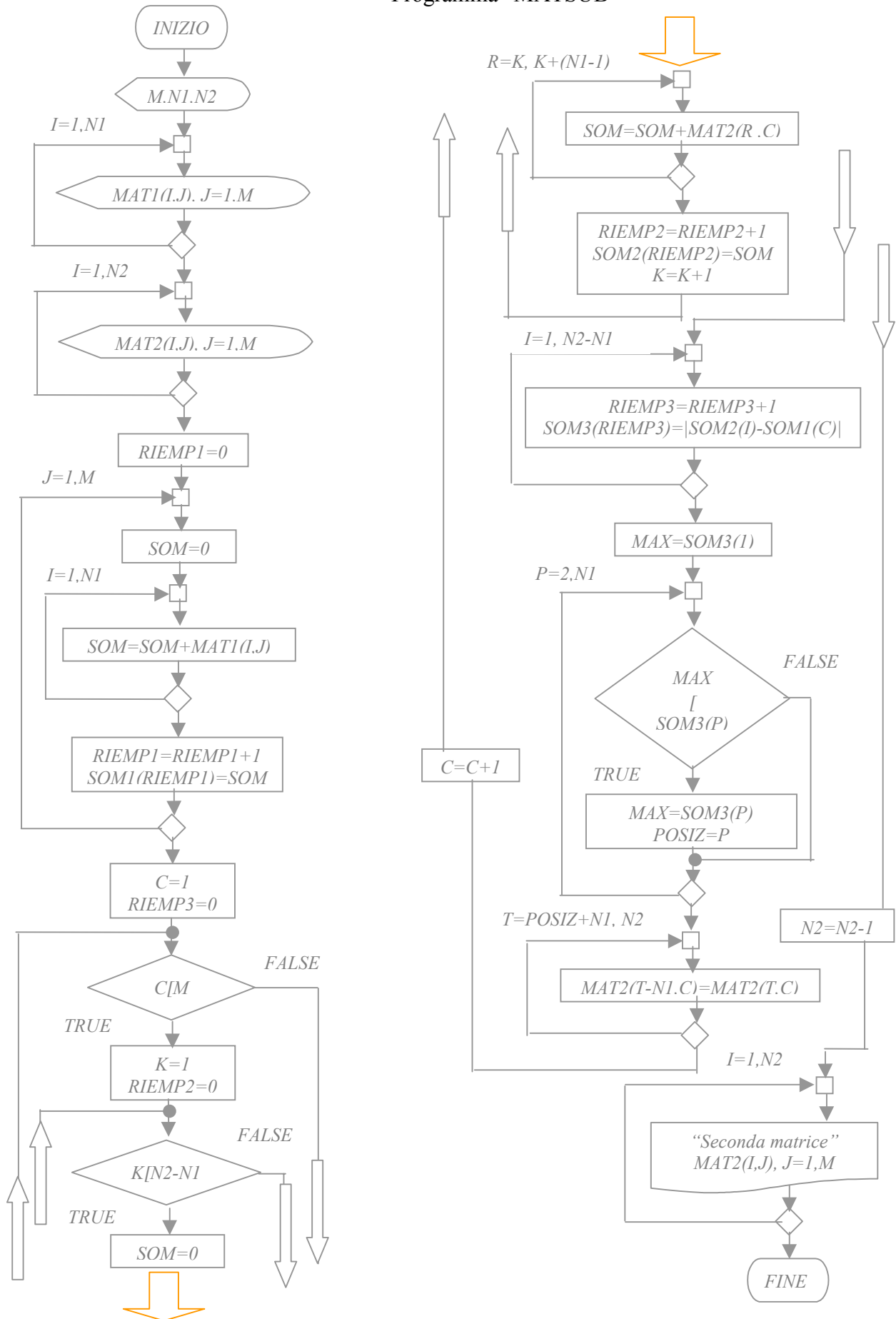
• Informazioni di uscita:

Nome	Tipo	Attributo
MAT2	matrice	MAT2 modificata

• Informazioni di algoritmo:

Nome	Tipo	Attributo
C	integer	Aggiorna il valore dell'indice di colonna della MAT2 nel ciclo "while..do"
I	integer	Aggiorna il valore dell'indice di riga MAT1,MAT2 nei cicli "do"
J	integer	Aggiorna il valore dell'indice di colonna di MAT1,MAT2 nei cicli "do"
K	integer	Aggiorna il valore dell'indice di riga della colonna corrente di MAT2 a partire dal quale si considera il sottovettore di lunghezza N1
SOM	integer	Acquista temporaneamente il valore somma degli elementi appartenenti ad una colonna di MAT1,MAT2
RIEMP1	integer	Riempimento vettore SOM1
SOM1	vettore	Contiene la somma degli elementi di una colonna di MAT1
RIEMP2	integer	Riempimento vettore SOM2
SOM2	vettore	Contiene la somma degli elementi in numero di N1 di una colonna di MAT2
RIEMP3	integer	Riempimento vettore SOM3
SOM3	vettore	Contiene la differenza in valore assoluto tra SOM1 e SOM2
MAX	integer	Contiene la differenza in valore assoluto più alta
POSIZ	integer	Contiene temporaneamente l'indice dell'elemento di SOM3 massimo
R	integer	Aggiorna il valore dell'indice di riga di MAT2 nel ciclo "do"
T	integer	Aggiorna il valore dell'indice di riga di MAT2 nel ciclo "do"
P	integer	Aggiorna l'indice dell'elemento di SOM3 nei cicli "do"

FLOW CHART
Programma "MATSUB"



CODIFICA FORTRAN

```

PROGRAM MATSUB
c   Date due matrici di interi Mat1,MAT2 aventi
c   lo stesso numero M di colonne e N1 e N2 righe
c   con N2>N1 scrivere un programma che: elimini
c   da MAT2 il sottovettore di lunghezza N1 tale
c   che la differenza fra la somma degli elementi
c   del sottovettore e la somma degli elementi
c   della corrispondente colonna di MAT1 risulti
c   massima in valore assoluto. In caso di differenze
c   uguali si elimini solo il primo dei sottovettori
c   individuati
c   Autore:Ermanno Napoli
c   Dichiarazione costanti
INTEGER CARDR,CARDC,CARD
PARAMETER(CARDR=50,CARDC=50,CARD=20)
c   Dichiarazione tipo
INTEGER MAT1(CARDR,CARDC),MAT2(CARDR,CARDC)
INTEGER SOM1(CARD),SOM2(CARD),SOM3(CARD)
c   Dichiarazione variabili
INTEGER C,I,J,K,M,N1,N2,P,POSIZ,T
INTEGER RIEMP1,RIEMP2,SOM,R,MAX,RIEMP3
c   Dichiarazione sottoprogramma
c   NESSUNA
c   Begin
c   Acquisizione dati in ingresso
WRITE(*,*)'Dammi il numero di colonne delle matrici'
WRITE(*,*)
READ(*,'(I2)') M
WRITE(*,*)'Dammi il numero di righe della prima matrice'
WRITE(*,*)
READ(*,'(I2)') N1
WRITE(*,*)'Dammi il numero di righe della seconda matrice'
WRITE(*,*)
READ(*,'(I2)') N2
WRITE(*,*)'Dammi la prima matrice'
WRITE(*,*)
DO 10 I=1,N1
READ(*,'(30I3)') (MAT1(I,J), J=1,M)
10 CONTINUE
WRITE(*,*)'Dammi la seconda matrice'
WRITE(*,*)
DO 20 I=1,N2
READ(*,'(30I3)') (MAT2(I,J), J=1,M)
20 CONTINUE
c   Per la matrice MAT1: inserisci la somma degli
c   elementi di ogni colonna nel vettore som1
RIEMP1=0
DO 30 J=1,M
SOM=0
DO 40 I=1,N1
SOM=SOM+MAT1(I,J)
40 CONTINUE
RIEMP1=RIEMP1+1
SOM1(RIEMP1)=SOM
30 continue

```

```

c      Per la matrice MAT2: per ogni colonna fai le
c      operazioni di seguito riportate
c      Inizio "while..do"
c      c=1
c      riemp3=0
100    continue
c      IF(.NOT.(C.LE.M)) GOTO 200
c      Per la colonna corrente individua il sottovettore
c      di lunghezza N1
c      Inizio "while..do"
c      riemp2=0
c      K=1
300    CONTINUE
c      IF(.NOT.(K.LE.N2-N1)) GOTO 400
c      Inserimento della somma degli elementi della corrente
c      colonna, in numero pari a N1, nel vettore SOM2
c      SOM=0
c      DO 50 R=K,K+(N1-1)
c      SOM=SOM+MAT2(R,C)
50     CONTINUE
c      RIEMP2=RIEMP2+1
c      SOM2(RIEMP2)=SOM
c      K=K+1
c      GOTO 300
400    CONTINUE
c      Inserisci le differenze, in valore assoluto, tra gli
c      gli elementi dei vettori SOM1,SOM2 nel vettore SOM3;
c      tale differenza deve essere eseguita tra l'elemento
c      di SOM1 di posizione pari a c, e tutti gli elementi
c      del vettore SOM2
c      DO 60 I=1,N2-N1
c      RIEMP3=RIEMP3+1
c      SOM3(RIEMP3)=ABS(SOM2(I)-SOM1(C))
60     CONTINUE
c      Cerca il valore massimo nel vettore SOM3 e segnala
c      la posizione di max; tale posizione coincide con
c      l'indice di riga nella colonna corrente a partire
c      dal quale si considera il sottovettore di lunghezza N1
c      MAX=SOM3(1)
c      DO 70 P=2,N1
c      IF(MAX.LE.SOM3(P)) THEN
c      MAX=SOM3(P)
c      POSIZ=P
c      ENDIF
70     CONTINUE
c      Individuato tramite posiz il sottovettore rispondente
c      alle specifiche del problema, si scambia il posto degli
c      elementi nella colonna corrente
c      DO 80 T=POSIZ+N1,N2
c      MAT2(T-N1,C)=MAT2(T,C)
80     CONTINUE
c      Aggiorna la colonna
c      C=C+1
c      GOTO 100
200    CONTINUE
c      Riduci il numero di righe della MAT2
c      N2=N2-N1
c      Stampa la matrice MAT2 modificata secondo
c      le specifiche del problema
c      WRITE(*,*)'Seconda matrice modificata'
c      WRITE(*,*)
c      DO 90 I=1,N2
c      WRITE(*,'(1x,30I3)') (MAT2(I,J), J=1,M)
90     CONTINUE
c      STOP
c      END

```

PROVA D'ESAME del 6 dicembre 1999

Siano dati in ingresso, da file o da tastiera, M vettori, ciascuno contenente N numeri interi. I valori di M ed N siano ugualmente forniti in ingresso insieme con un ulteriore valore K. Scrivere un programma che inserisca, volta per volta, i vettori in una matrice avente M righe ed N colonne secondo la seguente regola: se nel vettore letto esiste un valore maggiore di K, il vettore deve essere ordinato in senso crescente e inserito nella prima riga disponibile della matrice a partire dall'alto; altrimenti, si ordini il vettore in senso decrescente e lo si inserisca nella prima riga disponibile della matrice a partire dal basso. Si stampi, infine, la matrice risultante.

Si strutturi il programma in opportuni sottoprogrammi.

ESEMPIO

Ingresso					Uscita				
6	(M)								
5	(N)								
10	(K)								
vettori in ingresso					matrice in uscita				
2	3	8	7	7	2	4	8	11	13
13	8	4	11	2	3	4	8	18	23
1	6	2	3	9	1	3	9	15	21
18	8	4	23	3	7	6	5	4	3
7	5	6	3	4	9	6	3	2	1
1	15	21	9	3	8	7	6	3	2

DESCRIZIONE ALGORITMO

Si introducono i dati in input non come dei veri e propri vettori, ma, come vettori riga di una matrice avente M righe (cioè i vettori) ed N colonne (cioè gli elementi dei vettori). A questo punto verificare se esiste un elemento di uguale valore di K è semplice: basta confrontare ogni elemento del vettore riga con K e segnalare con una variabile booleana la sua esistenza. Per l'inserimento dei vettori nella matrice secondo le specifiche del problema si procede come segue: si fanno assumere a due variabili (P,S) i valori di indice di prima riga e ultima riga rispettivamente; poi si incrementa, nel corso dell'elaborazione, il valore della prima variabile e si decrementa il valore della seconda variabile. A inserimento avvenuto si ordinano gli elementi delle righe dall'alto in senso crescente, e dal basso in senso decrescente. Il numero di righe interessato nell'uno e nell'altro caso è fornito dalle suddette variabili dopo averle incrementate di una unità. Infine si stampano i risultati.

N.B.

L'algoritmo risolvete non è organizzato in sottoprogrammi. Per come è stato elaborato si presta a tale organizzazione in diverse maniere. Per esercizio si provi a strutturarli in sottoprogrammi come richiesto dal problema, fornendo anche le relative tabelle.

DOCUMENTAZIONE INTERNA AL PROGRAMMA

1- Costanti adoperate:

Nome	Valore	Tipo	Attributo
CARDR	50	intero	Una dimensione massima di array
CARDC	50	intero	Una dimensione massima di array

2- tipi di dato definiti dall'utente:

Nome	Definizione	Descrizione
vettore	array(1..cardr,1..cardc) of integer	Contiene numeri interi
Matrice	idem	idem

3- Struttura dei dati:

• Informazioni di ingresso:

Nome	Tipo	Attributo
M	integer	Numero di vettori in input (righe)
N	integer	Numero di elementi dei vettori (colonne)
VETT	vettore	Contiene il generico elemento del vettore

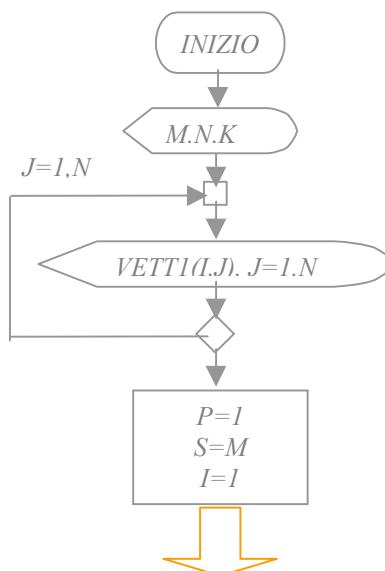
• Informazioni in uscita:

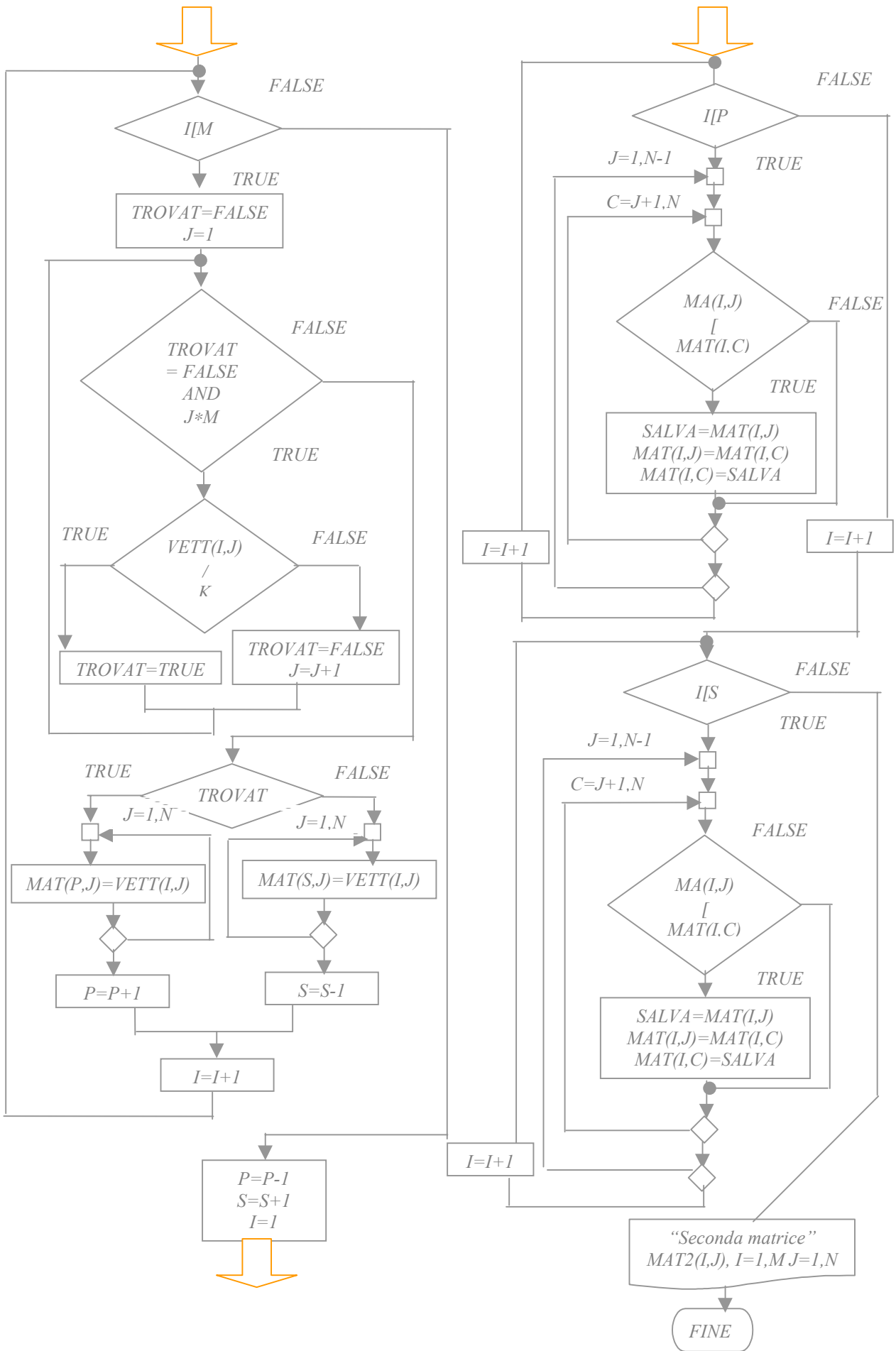
Nome	Tipo	Attributo
MAT	matrice	Contiene il generico elemento del vettore

• Informazioni di algoritmo:

Nome	Tipo	Attributo
C	integer	Aggiorna il valore di indice di colonna di MAT nei cicli "do"
I	integer	Aggiorna il valore di indice di riga di VETT,MAT nei cicli "do"
J	integer	Aggiorna il valore di indice di colonna di VETT,MAT nei cicli "do"
NUM	integer	Segnala il numero del vettore fornito in input
P	integer	Indica il valore della riga disponibile per inserire il vettore dall'alto
S	integer	Indica il valore della riga disponibile per inserire il vettore dal basso
TROVAT	boolean	Segnala se è stato trovato un elemento del vettore uguale a K
SALVA	integer	Variabile di appoggio nello scambio di valori

FLOW CHART
Programma "VETMAT"





```

PROGRAM VETMAT
c   Dati in input M vettori di N elementi ed
c   un valore K.Si inseriscano tali vettori
c   in una matrice MxN con la seguente regola:
c   se esiste almeno un elemento in un vettore
c   uguale a K si introduca tale vettore nella
c   matrice dall'alto nella prima posizione
c   disponibile; analogamente dal basso nel
c   caso non esistesse tale elemento
c   Autore:ERMANN0 NAPOLI
c   Dichiarazione costanti
      INTEGER CRDR,CARDC
      PARAMETER (CARDR=50,CARDC=50)
c   Dichiarazione tipo
      integer vett(cardr,cardc),mat(cardr,cardc)
c   Dichirazione variabili
      integer i,j,m,n,p,s,num
      integer c,d,salva
      logical trovat
c   Dichiarazione sottoprogrammi
      NESSUNO
c   Begin
c   Acquisizione dati in ingresso
      write(*,*)'Dammi il numero di vettori:'
      read(*,'(i2)') m
      write(*,*)'Dammi il numero di elementi:'
      read(*,'(i2)') n
      write(*,*)'Dammi il valore k:'
      read(*,'(i3)') k
      do 10 i=1,m
        num=i
        write(*,'(1x,'Dammi il vettore n°',i2)')num
        write(*,*)
        read(*,'(30i3)') (vett(i,j), j=1,n)
10    continue
c   Inizializzazione contatori righe per
c   l'inserimento dei vettori nella matrice
c   secondo le specifiche del problema
      p=1
      s=m
c   Inizio "while..do":si individui per il
c   generico vettore se esiste tra i suoi
c   elementi il valore K
      i=1
300  continue
      if(.not.(i.le.m)) goto 200
c   Inizio "while..do:si considerino tutti
c   gli elementi del vettore per la ricerca
c   del valore K
      trovat=.false.
      j=1
300  continue
      if(.not.((j.le.n).and.(.not.trovat))) goto 400
      if(vett(i,j).ge.k) then
        trovat=.true.
      else
        trovat=.false.
        j=j+1
      endif
      goto 300
400  continue
      Fine "while..do"

```

```

c   Introduzione dei vettori nella matrice
c   secondo le specifiche del problema
   if(trovat) then
   do 40 j=1,n
   mat(p,j)=vett(i,j)
40  continue
   p=p+1
   else
   do 50 j=1,n
   mat(s,j)=vett(i,j)
50  continue
   s=s-1
   endif
   i=i+1
   goto 100
200 continue
   Fine "while...do"
c   Si riordinino i vettori righe
c   secondo le specifiche del problema
   p=p-1
   s=s+1
   i=1
500 continue
   if(.not.(i.le.p)) goto 600
   do 20 j=1,n-1
   do 30 c=j+1,n
   if(mat(i,j).gt.mat(i,c)) then
   salva=mat(i,j)
   mat(i,j)=mat(i,c)
   mat(i,c)=salva
   endif
30  continue
20  continue
   i=i+1
   goto 500
600 continue
   i=p+1
700 continue
   if(.not.(i.le.s)) goto 800
   do 70 j=1,n-1
   do 80 c=j+1,n
   if(mat(i,j).lt.mat(i,c)) then
   salva=mat(i,j)
   mat(i,j)=mat(i,c)
   mat(i,c)=salva
   endif
80  continue
70  continue
   i=i+1
   goto 700
800 continue
c   Fine dell'ordinamento e
c   stampa dei risultati
   write(*,*)
   do 60 i=1,m
   write(*,'(1x,30i3)') (mat(i,j), j=1,n)
60  continue
   stop
   end

```