

8. Verifica in fase elastica e calcolo a rottura 497

```

REM *****
REM *
REM *          L a v o r o I n t e r n o
REM *
REM *****
REM
REM *****
REM *
REM *          Questa subroutine calcola il lavoro compiuto dalle forze
REM *          interne per effetto del prescelto meccanismo
REM *
REM *****
REM
REM *****
REM *
REM *          P R I M A   C E R N I E R A
REM *
REM *****
M7 = TLIM * C(JCERN(1)) * B(JCERN(1))
M8 = CLIM * LARG * S(JCERN(1)) * B(JCERN(1))
IF WCERN$(1) = "INTR" THEN
  IF F1 = -1 THEN
    U1 = -F1 * M7
  ELSE
    U1 = F1 * M8
  END IF
ELSE
  IF F1 = -1 THEN
    U1 = -F1 * M8
  ELSE
    U1 = F1 * M7
  END IF
END IF
REM
REM *****
REM *
REM *          S E C O N D A   C E R N I E R A
REM *
REM *****
M7 = TLIM * C(JCERN(2)) * B(JCERN(2))
M8 = CLIM * LARG * S(JCERN(2)) * B(JCERN(2))
IF WCERN$(2) = "INTR" THEN

```

498 *Le strutture ad arco*

```

IF F2 - F1 < 0 THEN
  U2 = -(F2 - F1) * M7
ELSE
  U2 = (F2 - F1) * M8
END IF
ELSE
  IF F2 - F1 < 0 THEN
    U2 = -(F2 - F1) * M8
  ELSE
    U2 = (F2 - F1) * M7
  END IF
END IF
REM
REM *****
REM *
REM *          T E R Z A   C E R N I E R A
REM *
REM *          *****
M7 = TLIM * C(JCERN(3)) * B(JCERN(3))
M8 = CLIM * LARG * S(JCERN(3)) * B(JCERN(3))
IF WCERN$(3) = "INTR" THEN
  IF F3 - F2 < 0 THEN
    U3 = -(F3 - F2) * M7
  ELSE
    U3 = (F3 - F2) * M8
  END IF
ELSE
  IF F3 - F2 < 0 THEN
    U3 = -(F3 - F2) * M8
  ELSE
    U3 = (F3 - F2) * M7
  END IF
END IF
REM
REM *****
REM *
REM *          Q U A R T A   C E R N I E R A
REM *
REM *          *****
M7 = TLIM * C(JCERN(4)) * B(JCERN(4))
M8 = CLIM * LARG * S(JCERN(4)) * B(JCERN(4))
IF WCERN$(4) = "INTR" THEN

```

```

IF F3 < 0 THEN
  U4 = -F3 * M8
ELSE
  U4 = F3 * M7
END IF
ELSE
  IF F3 < 0 THEN
    U4 = -F3 * M7
  ELSE
    U4 = F3 * M8
  END IF
END IF
LINT = U1 + U2 + U3 + U4
END SUB

```

```

SUB MomentiEstremi

```

```

REM
REM *****
REM *
REM *           M o m e n t i E s t r e m i
REM *
REM *****
REM
REM *****
REM *
REM *           Q u e s t a   s u b r o u t i n e   c a l c o l a   i   m o m e n t i   i n t r a d o s s a l i   e d
REM *           e s t r a d o s s a l i
REM *
REM *****
REM
REM DIM A1(2, 2), B1(2)
REM
REM *****
REM *
REM *           P R I M A   C E R N I E R A
REM *
REM *****
M7 = TLIM * C(JCERN(1)) * B(JCERN(1))
M8 = CLIM * LARG * S(JCERN(1)) * B(JCERN(1))
IF WCERN$(1) = "INTR" THEN
  IF F1 = 1 THEN
    M1 = M8

```

500 *Le strutture ad arco*

```
ELSE
  M1 = -M7
END IF
ELSE
  IF F1 = 1 THEN
    M1 = M7
  ELSE
    M1 = -M8
  END IF
END IF
REM
REM
REM *****
REM *
REM *          S E C O N D A   C E R N I E R A
REM *
REM *****
M7 = TLIM * C(JCERN(2)) * B(JCERN(2))
M8 = CLIM * LARG * S(JCERN(2)) * B(JCERN(2))
IF WCERN$(2) = "INTR" THEN
  IF F2 - F1 > 0 THEN
    M2 = M8
  ELSE
    M2 = -M7
  END IF
ELSE
  IF F2 - F1 > 0 THEN
    M2 = M7
  ELSE
    M2 = -M8
  END IF
END IF
REM
REM *****
REM *
REM *          T E R Z A   C E R N I E R A
REM *
REM *****
M7 = TLIM * C(JCERN(4)) * B(JCERN(4))
M8 = CLIM * LARG * S(JCERN(4)) * B(JCERN(4))
IF WCERN$(4) = "INTR" THEN
  IF F3 < 0 THEN
```

```

      M4 = M8
    ELSE
      M4 = -M7
    END IF
  ELSE
    IF F3 < 0 THEN
      M4 = M7
    ELSE
      M4 = -M8
    END IF
  END IF
  F7 = 0: F8 = 0: P7 = 0: P8 = 0: T7 = 0: T8 = 0
  FOR I = JCERN(1) TO JCERN(4) - 1
    F7 = F7 + FVERT(I) * (LTRAT * (I - 1) + LTRAT / 2 - ZCERN(1))
    P7 = P7 + FACCV(I) * (LTRAT * (I - 1) + LTRAT / 2 - ZCERN(1))
    T7 = T7 + FACCO(I) * ((Q(I + 1) + Q(I)) / 2 + YCERN(1))
  NEXT I
  FOR I = JCERN(2) TO JCERN(4) - 1
    F8 = F8 + FVERT(I) * (LTRAT * (I - 1) + LTRAT / 2 - ZCERN(2))
    P8 = P8 + FACCV(I) * (LTRAT * (I - 1) + LTRAT / 2 - ZCERN(2))
    T8 = T8 + FACCO(I) * ((Q(I + 1) + Q(I)) / 2 + YCERN(2))
  NEXT I
  IF COLLASSO$ = "PROPORZIONALE" THEN
    B1(1) = GAMMA * (F7 + P7 + T7) + M1 - M4
    B1(2) = GAMMA * (F8 + P8 + T8) + M2 - M4
  ELSE
    B1(1) = F7 + GAMMA * (P7 + T7) + M1 - M4
    B1(2) = F8 + GAMMA * (P8 + T8) + M2 - M4
  END IF
  A1(1, 1) = YCERN(4) - YCERN(1)
  A1(1, 2) = ZCERN(1) - ZCERN(4)
  A1(2, 1) = YCERN(4) - YCERN(2)
  A1(2, 2) = ZCERN(2) - ZCERN(4)
  DET = A1(1, 1) * A1(2, 2) - A1(1, 2) * A1(2, 1)
  H2 = (B1(1) * A1(2, 2) - B1(2) * A1(1, 2)) / DET
  V2 = -(B1(1) * A1(2, 1) - B1(2) * A1(1, 1)) / DET
  A2 = M4
  FOR I = JCERN(4) TO T
    IF COLLASSO$ = "PROPORZIONALE" THEN
      H2 = H2 - GAMMA * FACCO(I)
      V2 = V2 - GAMMA * (FVERT(I) + FACCV(I))
    ELSE

```

502 *Le strutture ad arco*

```
H2 = H2 - GAMMA * FACCO(I)
V2 = V2 - FVERT(I) - GAMMA * FACCV(I)
END IF
NEXT I
A2 = A2 + V2 * (LUCE - ZCERN(4)) + H2 * (YCERN(4) + RIAL)
FOR I = JCERN(4) TO T
  IF COLLASSO$ = "PROPORZIONALE" THEN
    A2 = A2 + GAMMA * (FVERT(I) + FACCV(I)) * (LTRAT * (I - JCERN(4))
      + LTRAT / 2)
    A2 = A2 - GAMMA * FACCO(I) * ((Q(I + 1) + Q(I)) / 2 + YCERN(4))
  ELSE
    A2 = A2 + (FVERT(I) + GAMMA * FACCV(I)) * (LTRAT * (I - JCERN(4))
      + LTRAT / 2)
    A2 = A2 - GAMMA * FACCO(I) * ((Q(I + 1) + Q(I)) / 2 + YCERN(4))
  END IF
NEXT I
FOR I = 1 TO T + 1
  F7 = 0: P7 = 0: T7 = 0: M7 = 0: M8 = 0: M9 = 0
  FOR J = I TO T
    F7 = F7 + FVERT(J)
    P7 = P7 + FACCV(J)
    T7 = T7 + FACCO(J)
    M7 = M7 + FVERT(J) * (LTRAT * (J - I) + LTRAT / 2)
    M8 = M8 + FACCV(J) * (LTRAT * (J - I) + LTRAT / 2)
    M9 = M9 + FACCO(J) * ((Q(J) + Q(J + 1)) / 2 - Q(I))
  NEXT J
  IF COLLASSO$ = "PROPORZIONALE" THEN
    N = -(V2 + GAMMA * (F7 + P7)) * SIN(INCL(I))
    N = N + (H2 + GAMMA * T7) * COS(INCL(I))
    M = A2 - V2 * (LUCE - LTRAT * (I - 1))
    M = M + H2 * (Q(I) - RIAL) - GAMMA * (M7 + M8 + M9)
  ELSE
    N = -(V2 + F7 + GAMMA * P7) * SIN(INCL(I))
    N = N + (H2 + GAMMA * T7) * COS(INCL(I))
    M = A2 - V2 * (LUCE - LTRAT * (I - 1))
    M = M + H2 * (Q(I) - RIAL) - M7 - GAMMA * (M8 + M9)
  END IF
  MINTR(I) = M - N * B(I) / 2
  MEXTR(I) = M + N * B(I) / 2
NEXT I
END SUB
```

```

SUB UscitaDati
REM
REM *****
REM *
REM *           U s c i t a D a t i
REM *
REM *****
REM
REM *****
REM *
REM *           Questa subroutine si occupa dell'uscita, su video, su stampa
REM *           e su file, dei dati di ingresso
REM *
REM *****
FOR ITER = 1 TO ITER1
  PRINT #ITER, " ====="
  PRINT #ITER, "PROGRAMMA AR1 - ANALISI A ROTTURA STATICA
  PRINT #ITER, TITLE$
  PRINT #ITER, AUMENTO DEI CARICHI "; COLLASSO$
  PRINT #ITER, " ====="
  PRINT #ITER,
  PRINT #ITER,
  PRINT #ITER,
  PRINT #ITER, "FILE DEI DATI ="; FILEINPUT$
  PRINT #ITER, "LUCE DELL'ARCO = "; LUCE
  PRINT #ITER, "FRECCIA DELL'ARCO = "; FREC
  PRINT #ITER, "RIALZO A DESTRA ="; RIAL
  PRINT #ITER, "LARGHEZZA DELLA SEZIONE ="; LARG
  PRINT #ITER, "NUMERO DEI TRATTI DI DIVISIONE = "; T
  PRINT #ITER, "TENSIONE LIMITE NELL'ACCIAIO = "; TLIM
  PRINT #ITER, "COMPRESSIONE LIMITE NEL CONGLOMERATO = "; CLIM
  PRINT #ITER, "ALTEZZA DELLA SEZIONE A SINISTRA = "; HS
  PRINT #ITER, "ALTEZZA DELLA SEZIONE AL CENTRO = "; HC
  PRINT #ITER, "ALTEZZA DELLA SEZIONE A DESTRA = "; HD
  PRINT #ITER, "SPESSORE DELLA SOLETTA A SINISTRA = "; SPES
  PRINT #ITER, "SPESSORE DELLA SOLETTA AL CENTRO = "; SPEC
  PRINT #ITER, "SPESSORE DELLA SOLETTA A DESTRA = "; SPED
  PRINT #ITER, "AREA DEL FERRO A SINISTRA = "; FERS
  PRINT #ITER, "AREA DEL FERRO AL CENTRO = "; FERC
  PRINT #ITER, "AREA DEL FERRO A DESTRA = "; FERD
  PRINT #ITER, "RAPPORTO MASSIMO SUL TEST DI USCITA "; MAXR

```

```

PRINT #ITER, "DIVIDENTI DELLE CERNIERE DI TENTATIVO = ";
      JCERN(1), JCERN(2), JCERN(3), JCERN(4)
FOR I = 1 TO 4
  IF WCERN$(I) = "INTR" THEN
    PRINT #ITER, "LA CERNIERA "; I; " E' ALL'INTRADOSSO"
  ELSE
    PRINT #ITER, "LA CERNIERA "; I; " E' ALL'ESTRADOSSO"
  END IF
NEXT I
PRINT #ITER,
IF CARICHI$ = "FUNC" THEN
  PRINT #ITER, "CARICO VERTICALE DA PESO PROPRIO A SINISTRA = ";
    QPROPS
  PRINT #ITER, "CARICO VERTICALE DA PESO PROPRIO AL CENTRO = ";
    QPROPC
  PRINT #ITER, "CARICO VERTICALE DA PESO PROPRIO A DESTRA = ";
    QPROPD
  PRINT #ITER, "CARICO VERTICALE ACCIDENTALE A SINISTRA = ";
    QACCVS
  PRINT #ITER, "CARICO VERTICALE ACCIDENTALE A SINISTRA = ";
    QACCVS
  PRINT #ITER, "CARICO VERTICALE ACCIDENTALE A SINISTRA = ";
    QACCVS
  PRINT #ITER, "CARICO ORIZZONTALE ACCIDENTALE A SINISTRA = ";
    QACCOS
  PRINT #ITER, "CARICO ORIZZONTALE ACCIDENTALE AL CENTRO = ";
    QACCOC
  PRINT #ITER, "CARICO ORIZZONTALE ACCIDENTALE A DESTRA = ";
    QACCOD
END IF
REM -----
REM STAMPA LE QUOTE DELL'ARCO PER CIASCUNA DIVIDENTE
REM -----
IF ASSE$ = "DATI" THEN
  PRINT #ITER,
  PRINT #ITER, " ===== "
  PRINT #ITER, " = "
  PRINT #ITER, " =          TABELLA DELLE QUOTE DELL'ARCO          = "
  PRINT #ITER, " = "
  PRINT #ITER, " ===== "
  PRINT #ITER,
  PRINT #ITER, " ----- "

```



```

PRINT #ITER, " DIVIDENTE                QUOTA                "
PRINT #ITER, " _____"
PRINT #ITER,
FOR I = 1 TO T + 1
  PRINT #ITER, TAB(1); I; TAB(34); Q(I)
NEXT I
END IF
IF CARICHI$ = "DATI" THEN
  REM _____
  REM STAMPA I CARICHI PER CIASCUNA DIVIDENTE
  REM _____
  PRINT #ITER,
  PRINT #ITER, " ====="
  PRINT #ITER, " =                                     ="
  PRINT #ITER, " =          CARICHI VERTICALI SULL'ARCO          ="
  PRINT #ITER, " =                                     ="
  PRINT #ITER, " ====="
  PRINT #ITER,
  PRINT #ITER, " _____"
  PRINT #ITER, " DIVIDENTE VERT. PROPRIO VERT. ACC. ORIZ. ACC. "
  PRINT #ITER, " _____"
  PRINT #ITER,
  FOR I = 1 TO T
    PRINT #ITER, TAB(2); I; TAB(9); FVERT(I); TAB(25); FACCV(I); TAB(45);
      FACCO(I)
  NEXT I
NEXT ITER
END SUB

SUB UscitaRisultati
REM
REM *****
REM *
REM *          U s c i t a R i s u l t a t i
REM *
REM *****
REM
REM *****
REM *
REM *          Questa subroutine si occupa dell'uscita (su video,
REM *          su stampa e su file) dei risultati
REM *

```

```

REM *****
REM
REM _____
REM CALCOLA SE LA CRISI AVVIENE PER SNERVAMENTO DEL FERRO
REM O PER SCHIACCIAMENTO DEL CONGLOMERATO
REM _____
IF WCERN$(1) = "INTR" THEN
  IF F1 = -1 THEN CRISI$(1) = "FERRO" ELSE CRISI$(1) = "CONGLOMER-
    ATO"
ELSE
  IF F1 = -1 THEN CRISI$(1) = "CONGLOMERATO" ELSE
    CRISI$(1) = "FERRO"
END IF
IF WCERN$(2) = "INTR" THEN
  IF F2 - F1 < 0 THEN CRISI$(2) = "FERRO" ELSE CRISI$(2) = "CONGLOM-
    ERATO"
ELSE
  IF F2 - F1 < 0 THEN CRISI$(2) = "CONGLOMERATO" ELSE CRISI$(2) =
    "FERRO"
END IF
IF WCERN$(3) = "INTR" THEN
  IF F3 - F2 < 0 THEN CRISI$(3) = "FERRO" ELSE CRISI$(3) = "CONGLOM-
    ERATO"
ELSE
  IF F3 - F2 < 0 THEN CRISI$(3) = "CONGLOMERATO" ELSE CRISI$(3) =
    "FERRO"
END IF
IF WCERN$(4) = "INTR" THEN
  IF F3 > 0 THEN CRISI$(4) = "FERRO" ELSE CRISI$(4) = "CONGLOMER-
    ATO"
ELSE
  IF F3 > 0 THEN CRISI$(4) = "CONGLOMERATO" ELSE
    CRISI$(4) = "FERRO"
END IF
FOR ITER = ITER1 + 1 TO ITER2
  PRINT #ITER, "MULTIPLICATORE STATICAMENTE AMMISSIBILE = ";
  PSI
  PRINT #ITER, "MULTIPLICATORE CINEMATICAMENTE SUFFICIENTE =
    "; GAMMA
  PRINT #ITER,
  PRINT #ITER, "CERNIERA ASCISSA LOCAZIONE CRISI"
  PRINT #ITER,

```

```

FOR I = 1 TO 4
  AA = LTRAT * (JCERN(I) - 1)
  PRINT #ITER, TAB(2); JCERN(I); TAB(12); AA; TAB(29); WCERN$(I);
  TAB(45); CRISI$(I)
NEXT I
PRINT #ITER,
PRINT #ITER, "MOMENTI INTRADOSSALI MOMENTI ESTRADOSSALI"
PRINT #ITER,
FOR K = 1 TO T + 1
  PRINT #ITER, TAB(2); K; TAB(8); MINTR(K); TAB(35); MEXTR(K)
NEXT K
PRINT #ITER,
IF COLLASSO$ = "NON PROPORZIONALE" THEN
  PRINT #ITER,
  PRINT #ITER, "MOMENTI INTRADOSSALI MOMENTI ESTRADOSSALI"
  PRINT #ITER, "DA PESO PROPRIO DA PESO PROPRIO"
  PRINT #ITER,
  FOR K = 1 TO T + 1
    PRINT #ITER, TAB(2); K; TAB(8); MINTRP(K); TAB(35); MEXTRP(K)
  NEXT K
  PRINT #ITER,
END IF
PRINT #ITER,
PRINT #ITER, "MOM. LIMITI FERRO MOM. LIMITI CONGLOMERATO"
PRINT #ITER,
FOR K = 1 TO T + 1
  PRINT #ITER, TAB(2); K; TAB(8); TLIM * C(K) * B(K); TAB(35); CLIM *
  LARG * S(K) * B(K)
NEXT K
PRINT #ITER,
PRINT #ITER, "TEMPO DI ESECUZIONE = "; TIME2; " SEC."
PRINT #ITER, "RUN DEL "; DATE$; " ORE "; TIME$
NEXT ITER
END SUB

```


Appendice 8.2. Il programma AR2

```
DEFDBL A-Z
DECLARE SUB MomentiResidui ()
DECLARE SUB CernieraVariata ()
DECLARE SUB Fuoruscite ()
DECLARE SUB EstremoSuperiore ()
DECLARE SUB EstremoInferiore ()
DECLARE SUB CernierePlastiche ()
DECLARE SUB LavoroMomenti ()
DECLARE SUB Geometria ()
DECLARE SUB UscitaDati ()
DECLARE SUB UscitaRisultati ()
DECLARE SUB IngressoDati ()
DECLARE SUB IngressoDatiArray ()
DECLARE SUB OpenFiles ()
REM *****
REM *
REM *          P R O G R A M M A      A R 2          *
REM *
REM *                      d i                      *
REM *
REM *          V I N C E N Z O      F R A N C I O S I      *
REM *
REM *          Release 1.0.0. 25 Novembre 1987 (in BASIC HP) *
REM *          Release 3.0.0. Ottobre 1994 (in Microsoft QBasic) *
REM *
REM *****
REM
REM
REM *****
REM *
REM *          Questo programma esegue l'analisi limite per collasso incrementale *
REM *                      di un arco in conglomerato armato *
REM *
```

```

REM *
REM *****
OPTION BASE 1
REM
REM *****
REM *
REM *          C O M M O N
REM *
REM *****
REM
COMMON SHARED LUCE, LARG, TLIM, CLIM, HS, HC, HD, SPES, SPEC, SPED
COMMON SHARED FERS, FERC, FERD, MAXR, FREC, RIAL, T
COMMON SHARED QPROPS, QPROC, QPROPD, QACCVS, QACCVC
COMMON SHARED QACCVD, QACCOS, QACCOC, QACCOD
COMMON SHARED LTRAT, F1, F2, F3, E1, E2
COMMON SHARED GAMMA, PSI, I9, G9, W9$
COMMON SHARED ITER1, ITER2, TIME2
COMMON SHARED JCERN() AS DOUBLE, Q() AS DOUBLE
COMMON SHARED B() AS DOUBLE, S() AS DOUBLE, C() AS DOUBLE
COMMON SHARED YCERN() AS DOUBLE, ZCERN() AS DOUBLE
COMMON SHARED MIMAX() AS DOUBLE, MEMAX() AS DOUBLE
COMMON SHARED MIMIN() AS DOUBLE, MEMIN() AS DOUBLE
COMMON SHARED MIG() AS DOUBLE, MEG() AS DOUBLE
COMMON SHARED R1() AS DOUBLE, R2() AS DOUBLE
COMMON SHARED R() AS DOUBLE, G() AS DOUBLE, INCL() AS DOUBLE
COMMON SHARED TITLE$, STAMPA$, STAMPARIS$, FILEINGR$, FILEUSC$
COMMON SHARED ASSE$, CARICHI$, FILEINPUT$, COLLASSO$
COMMON SHARED WCERN$(), CRISI$(), WCERN1$()
REM
REM *****
REM *
REM *          DIMENSIONA GLI ARRAY STATICI
REM *
REM *****
REM
DIM TITLE$(100), TITLE1$(100), STAMPA$(10), STAMPARIS$(10)
DIM FILEINGR$(10), FILEUSC$(10), JCERN(4) AS DOUBLE
DIM WCERN$(4), CRISI$(4), YCERN(5) AS DOUBLE, ZCERN(5) AS DOUBLE
REM
REM *****
REM *
REM *          DEFINISCE ALCUNE LEGGI DI VARIAZIONE
REM *

```

```

REM *
REM *****
REM
DEF FNB (Z) = HS + (HD - HS) / LUCE * Z - 4 * (HS + (HD - HS) / 2 - HC) /
  LUCE ^ 2 * Z * (LUCE - Z)
DEF FNS (Z) = SPES + (SPED - SPES) / LUCE * Z - 4 * (SPES + (SPED - SPES)
  / 2 - SPEC) / LUCE ^ 2 * Z * (LUCE - Z)
DEF FNC (Z) = FERS + (FERD - FERS) / LUCE * Z - 4 * (FERS + (FERD -
  FERS) / 2 - FERC) / LUCE ^ 2 * Z * (LUCE - Z)
DEF FNQ (Z) = RIAL * Z / LUCE + 4 * FREC / LUCE ^ 2 * Z * (LUCE - Z)
CLS
REM
REM *****
REM *
REM *          LEGGE IL NOME DEL FILE DI INPUT DATI          *
REM *
REM *****
REM
PRINT "NOME DEL FILE DI INPUT"
INPUT FILEINPUT$
TIME1 = TIMER
OPEN FILEINPUT$ FOR INPUT AS #7
REM -----
REM CHIAMA LA SUBROUTINE PER L'INGRESSO DEI DATI
REM -----
CALL IngressoDati
LTRAT = LUCE / T
F1 = -1
REM
REM *****
REM *
REM *          REDIMENSIONA GLI ARRAY DINAMICI          *
REM *
REM *****
REM
REDIM Q(T + 1) AS DOUBLE, MIMAX(T + 1) AS DOUBLE
REDIM MEMAX(T + 1) AS DOUBLE, INCL(T + 1) AS DOUBLE
REDIM MIMIN(T + 1) AS DOUBLE, MEMIN(T + 1) AS DOUBLE
REDIM B(T + 1) AS DOUBLE, S(T + 1) AS DOUBLE
REDIM C(T + 1) AS DOUBLE, MIG(T + 1) AS DOUBLE
REDIM R1(T + 1) AS DOUBLE, R2(T + 1) AS DOUBLE
REDIM R(T + 1) AS DOUBLE, G(T + 1) AS DOUBLE

```

```
REDIM MEG(T + 1) AS DOUBLE, WCERN1$(T + 1)
REM _____
REM CHIAMA LA SUBROUTINE PER L'INGRESSO DEI DATI IN ARRAY
REM _____
CALL IngressoDatiArray
REM _____
REM CHIAMA LA SUBROUTINE PER RIEMPIRE ALCUNI ARRAY
REM _____
CALL Geometria
REM _____
REM CHIAMA LA SUBROUTINE PER STAMPARE I DATI DI INGRESSO
REM _____
TIME2 = TIMER - TIME1
CALL UscitaDati
TIME1 = TIMER
REM _____
REM COMINCIA IL CALCOLO RIPETITIVO DEL MOLTIPLICATORE
REM CINEMATICAMENTE SUFFICIENTE
REM _____
COUNT = 0
DO
  REM _____
  REM CHIAMA LA SUBROUTINE PER IL CALCOLO DELLE COORDINATE
  REM DELLE CERNIERE PLASTICHE E DEL CENTRO DI ROTAZIONE
  REM DEL TRATTO MEDIO
  REM _____
  CALL CernierePlastiche
  REM _____
  REM CHIAMA LA SUBROUTINE PER IL CALCOLO DEGLI ANGOLI
  REM DI ROTAZIONE E DEL LAVORO DEI MOMENTI
  REM _____
  CALL LavoroMomenti
  REM _____
  REM ESEGUE IL TEST SUL VERSO DEL MECCANISMO
  REM _____
  IF E2 < 0 THEN
    F1 = -F1
    CALL LavoroMomenti
  END IF
  REM _____
  REM CHIAMA LA SUBROUTINE PER CALCOLARE IL COEFFICIENTE
  REM CINEMATICAMENTE SUFFICIENTE
```



```
REM _____  
CALL EstremoSuperiore  
REM _____  
REM CHIAMA LA SUBROUTINE PER CALCOLARE I MOMENTI  
REM INTRADOSSALI ED ESTRADOSSALI RESIDUI  
REM _____  
CALL MomentiResidui  
REM _____  
REM CHIAMA LA SUBROUTINE PER IL CALCOLO DELLE FUORUSCITE  
REM DALLA FRONTIERA LIMITE  
REM _____  
CALL Fuoruscite  
REM _____  
REM CHIAMA LA SUBROUTINE PER CALCOLARE IL COEFFICIENTE  
REM STATICAMENTE AMMISSIBILE  
REM _____  
CALL EstremoInferiore  
COUNT = COUNT + 1  
FOR ITER = ITER1 + 1 TO ITER2  
  PRINT #ITER, "INTERVALLO ALL'ITERAZIONE "; COUNT  
  PRINT #ITER, "GAMMA ="; GAMMA  
  PRINT #ITER, "PSI ="; PSI  
  PRINT #ITER,  
NEXT ITER  
REM _____  
REM TEST DI USCITA DAL CICLO  
REM _____  
IF (GAMMA - PSI) / PSI < MAXR THEN EXIT DO  
REM _____  
REM CHIAMA LA SUBROUTINE PER VARIARE LA POSIZIONE  
REM DELLE CERNIERE  
REM _____  
CALL CernieraVariata  
LOOP  
REM _____  
REM CHIAMA LA SUBROUTINE PER STAMPARE I RISULTATI  
REM _____  
TIME2 = TIME2 + TIMER - TIME1  
CALL UscitaRisultati  
REM _____  
REM CHIUDE TUTTI I BUFFER TEMPORANEI  
REM _____
```

CLOSE
END

SUB CernieraVariata

REM

REM *****

REM * * * * * *

REM * C e r n i e r a V a r i a t a * *

REM * * * * * *

REM *****

REM

REM *****

REM * * * * * *

REM * Questa subroutine varia sia le dividendi che le locazioni * *

REM * delle quattro cerniere. * *

REM * * * * * *

REM *****

REM

SELECT CASE I9

CASE IS <= JCERN(1)

JCERN(1) = I9

WCERN\$(1) = W9\$

G(JCERN(1)) = G9

CASE IS <= JCERN(2)

IF G9 * F1 > 0 THEN

JCERN(1) = I9

WCERN\$(1) = W9\$

G(JCERN(1)) = G9

ELSE

JCERN(2) = I9

WCERN\$(2) = W9\$

G(JCERN(2)) = G9

END IF

CASE IS <= JCERN(3)

IF G9 * (F2 - F1) > 0 THEN

JCERN(2) = I9

WCERN\$(2) = W9\$

G(JCERN(2)) = G9

ELSE

JCERN(3) = I9

WCERN\$(3) = W9\$

G(JCERN(3)) = G9

```

END IF
CASE IS <= JCERN(4)
  IF G9 * (F3 - F2) > 0 THEN
    JCERN(3) = I9
    WCERN$(3) = W9$
    G(JCERN(3)) = G9
  ELSE
    JCERN(4) = I9
    WCERN$(4) = W9$
    G(JCERN(4)) = G9
  END IF
CASE IS > JCERN(4)
  JCERN(4) = I9
  WCERN$(4) = W9$
  G(JCERN(4)) = G9
END SELECT
END SUB

```

```

SUB EstremoSuperiore

```

```

REM
REM *****
REM *
REM *           E s t r e m o S u p e r i o r e
REM *
REM *****
REM
REM *****
REM *
REM *           Questa subroutine calcola il coefficiente cinematicamente
REM *           sufficiente
REM *
REM *****
REM
GAMMA = E1 / E2
END SUB

```

```

SUB Fuoruscite

```

```

REM
REM *****
REM *
REM *           F u o r u s c i t e
REM *
REM *****

```

```

REM *****
REM
REM *****
REM *
REM *          Questa subroutine calcola le fuoruscite dalla frontiera
REM *
REM *          limite
REM *
REM *****
REM
FOR I = 1 TO T + 1
  G(I) = 0
  WCERN1$(I) = ""
  R(I) = 0
NEXT I
IF COLLASSO$ = "PROPORZIONALE" THEN
  FOR I = 1 TO T + 1
    C8 = 1
    K1 = (GAMMA * (MIG(I) + MIMAX(I)) + R1(I)) / (CLIM * LARG * S(I) *
      B(I))
    IF K1 > C8 THEN
      C8 = K1
      WCERN1$(I) = "INTR"
      G(I) = 1
    END IF
    K2 = (GAMMA * (MEG(I) + MEMAX(I)) + R2(I)) / (TLIM * C(I) * B(I))
    IF K2 > C8 THEN
      C8 = K2
      WCERN1$(I) = "EXTR"
      G(I) = 1
    END IF
    K3 = -(GAMMA * (MIG(I) + MIMIN(I)) + R1(I)) / (TLIM * C(I) * B(I))
    IF K3 > C8 THEN
      C8 = K3
      WCERN1$(I) = "INTR"
      G(I) = -1
    END IF
    K4 = -(GAMMA * (MEG(I) + MEMIN(I)) + R2(I)) / (CLIM * LARG * S(I) *
      B(I))
    IF K4 > C8 THEN
      C8 = K4
      WCERN1$(I) = "EXTR"
      G(I) = -1
    END IF
  NEXT I

```

```

    END IF
    IF C8 > 1 THEN R(I) = C8
NEXT I
ELSE
  FOR I = 1 TO T + 1
    C8 = 1
    K1 = (GAMMA * MIMAX(I) + R1(I)) / (CLIM * LARG * S(I) * B(I) - MIG(I))
    IF K1 > C8 THEN
      C8 = K1
      WCERN1$(I) = "INTR"
      G(I) = 1
    END IF
    K2 = (GAMMA * MEMAX(I) + R2(I)) / (TLIM * C(I) * B(I) - MEG(I))
    IF K2 > C8 THEN
      C8 = K2
      WCERN1$(I) = "EXTR"
      G(I) = 1
    END IF
    K3 = (GAMMA * MIMIN(I) + R1(I)) / (-TLIM * C(I) * B(I) - MIG(I))
    IF K3 > C8 THEN
      C8 = K3
      WCERN1$(I) = "INTR"
      G(I) = -1
    END IF
    K4 = (GAMMA * MEMIN(I) + R2(I)) / (-CLIM * LARG * S(I) * B(I) - MEG(I))
    IF K4 > C8 THEN
      C8 = K4
      WCERN1$(I) = "EXTR"
      G(I) = -1
    END IF
    IF C8 > 1 THEN R(I) = C8
  NEXT I
END IF
END SUB

SUB Geometria
REM
REM *****
REM *
REM *           G e o m e t r i a
REM *
REM *****

```

```

REM -----
REM Riempimento dell'array delle quote, se descritte da funzioni
REM -----
IF ASSE$ = "FUNC" THEN
  FOR I = 1 TO T + 1
    Z = LTRAT * (I - 1)
    Q(I) = FNQ(Z)
  NEXT I
END IF
REM -----
REM Riempimento degli array delle inclinazioni dei tratti
REM -----
INCL(1) = ATN(Q(2) / LUCE * T)
FOR I = 2 TO T
  INCL(I) = ATN((Q(I + 1) - Q(I - 1)) / LUCE / 2 * T)
NEXT I
INCL(T + 1) = ATN((Q(T + 1) - Q(T)) / LUCE * T)
REM -----
REM Riempimento degli array delle altezze, degli spessori
REM e delle aree di ferro
REM -----
FOR I = 1 TO T + 1
  Z = LTRAT * (I - 1)
  B(I) = FNB(Z)
  S(I) = FNS(Z)
  C(I) = FNC(Z)
NEXT I
END SUB

SUB IngressoDati
REM
REM *****
REM *
REM *           I n g r e s s o D a t i
REM *
REM *****
REM
REM *****
REM *
REM *           Questa subroutine si occupa dell'ingresso dei dati
REM *           non organizzati in array
REM *

```

```

REM *****
REM *****
REM *
REM * LEGGE IL TITOLO DELLA STRUTTURA *
REM * IL FLAG DI STAMPA PER LE VARIABILI DI INGRESSO *
REM * IL FLAG DI STAMPA PER LE VARIABILI DI USCITA *
REM * - CARTA → I DATI VENGONO STAMPATI *
REM * - VIDEO → I DATI VENGONO INVIATI A VIDEO *
REM * - FILES → I DATI VENGONO MEMORIZZATI SU *
REM * FILES DA SPECIFICARE *
REM * - VIDEOFILES → I DATI VENGONO INVIATI A *
REM * VIDEO E MEMORIZZATI SU FILES *
REM * - CARTAFILES → I DATI VENGONO STAMPATI *
REM * CARTA E MEMORIZZATI SU FILES *
REM * - TUTTO → I DATI VENGONO INVIATI A VIDEO, *
REM * STAMPATI SU CARTA E MEMORIZZATI SU FILES *
REM *
REM *****
REM
INPUT #7, TITLE$
INPUT #7, STAMPA$
INPUT #7, STAMPARIS$
IF STAMPA$ = "FILES" OR STAMPA$ = "VIDEOFILES" OR STAMPA$ = "CAR-
TAFILES" OR STAMPA$ = "TUTTO" THEN
INPUT #7, FILEINGR$
END IF
IF STAMPARIS$ = "FILES" OR STAMPARIS$ = "VIDEOFILES" OR STAM-
PARIS$ = "CARTAFILES" OR STAMPARIS$ = "TUTTO" THEN
INPUT #7, FILEUSC$
END IF
CALL OpenFiles
REM
REM *****
REM *
REM * LEGGE: *
REM * LUCE Luce dell'arco *
REM * T Numero di tratti di divisione *
REM * LARG Larghezza della sezione (costante) *
REM * TLIM Tensione limite dell'acciaio *
REM * CLIM Compressione limite del conglomerato *
REM * (in valore assoluto) *
REM * HS Altezza della sezione a sinistra *

```

520 *Le strutture ad arco*

```

REM *      HC      Altezza della sezione al centro      *
REM *      HD      Altezza della sezione a destra      *
REM *      SPES    Spessore della soletta inferiore      *
REM *              (e superiore) a sinistra              *
REM *      SPEC    Spessore della soletta inferiore      *
REM *              (e superiore) al centro              *
REM *      SPED    Spessore della soletta inferiore      *
REM *              (e superiore) a destra              *
REM *      FERS    Area del ferro a sinistra            *
REM *      FERC    Area del ferro al centro            *
REM *      FERD    Area del ferro a destra            *
REM *      MAXR    Valore limite sul test di uscita     *
REM *      JCERN(4) Indici delle dividenti dove sono    *
REM *              inizialmente ipotizzate le cerniere  *
REM *      WCERN$(4) INTR o EXTR a seconda che la      *
REM *              corrispondente cerniera sia all'intradosso *
REM *              o all'estradosso                    *
REM *      ASSE$   Se l'asse dell'arco e' descritto da una *
REM *              funzione, ASSE$ = FUNZ, se invece e' dato *
REM *      COLLASSO$ PROPORZIONALE/NON PROPORZIONALE *
REM *              se il collasso avviene per aumento proporzionale *
REM *              (o meno) dei carichi.                *
REM *      Se l'asse e' definito da funzione:          *
REM *      FREC    Freccia dell'arco                  *
REM *      RIAL    Rialzo a destra                    *
REM *
REM *****
INPUT #7, LUCE, T, LARG, TLIM, CLIM
INPUT #7, HS, HC, HD, SPES, SPEC, SPED, FERS, FERC, FERD, MAXR
FOR I = 1 TO 4
  INPUT #7, JCERN(I)
NEXT I
FOR I = 1 TO 4
  INPUT #7, WCERN$(I)
NEXT I
INPUT #7, ASSE$, COLLASSO$
IF ASSE$ = "FUNC" THEN
  INPUT #7, FREC, RIAL
END IF
END SUB

```

SUB IngressoDatiArray


```

REM
REM *****
REM *
REM *           I n g r e s s o D a t i A r r a y           *
REM *
REM *****
REM
REM *****
REM *
REM *           Questa subroutine si occupa dell'ingresso dei dati           *
REM *           organizzati in array                                         *
REM *
REM *****
REM
REM _____
REM Legge le quote dell'arco
REM _____
IF ASSE$ = "DATI" THEN
  FOR I = 1 TO T + 1
    INPUT #7, Q(I)
  NEXT I
  RIAL = Q(T + 1)
  FREC = Q(T / 2 + 1) - RIAL / 2
END IF
REM
REM _____
REM Legge i momenti intradossali massimi
REM _____
FOR I = 1 TO T + 1
  INPUT #7, MIMAX(I)
NEXT I
REM
REM _____
REM Legge i momenti intradossali minimi
REM _____
FOR I = 1 TO T + 1
  INPUT #7, MIMIN(I)
NEXT I
REM
REM _____
REM Legge i momenti estradossali massimi
REM _____

```

```

FOR I = 1 TO T + 1
  INPUT #7, MEMAX(I)
NEXT I
REM
REM -----
REM Legge i momenti estradossali minimi
REM -----
FOR I = 1 TO T + 1
  INPUT #7, MEMIN(I)
NEXT I
REM
REM -----
REM Legge i momenti intradossali da peso proprio
REM -----
FOR I = 1 TO T + 1
  INPUT #7, MIG(I)
NEXT I
REM
REM -----
REM Legge i momenti estradossali da peso proprio
REM -----
FOR I = 1 TO T + 1
  INPUT #7, MEG(I)
NEXT I
END SUB

SUB LavoroMomenti
REM
REM *****
REM *
REM *           L a v o r o M o m e n t i
REM *
REM *****
REM
REM *****
REM *
REM *           Questa subroutine calcola il lavoro compiuto dai momenti
REM *
REM *****
REM
DIM DELTAF(4), N(4), D(4)
REM -----

```

```

REM CALCOLO DEGLI ANGOLI DI ROTAZIONE
REM _____
F2 = -F1 * (ZCERN(2) - ZCERN(1)) / (ZCERN(5) - ZCERN(2))
FAUX = F1 * (ZCERN(2) - ZCERN(1)) * (ZCERN(3) - ZCERN(5))
F3 = FAUX / ((ZCERN(5) - ZCERN(2)) * (ZCERN(4) - ZCERN(3)))
REM _____
REM ROTAZIONI RELATIVE
REM _____
DELTA(1) = F1
DELTA(2) = F2 - F1
DELTA(3) = F3 - F2
DELTA(4) = -F3
SELECT CASE COLLASSO$
CASE "PROPORZIONALE"
  FOR I = 1 TO 4
    IF DELTA(I) > 0 THEN
      IF WCERN$(I) = "INTR" THEN
        N(I) = CLIM * LARG * S(JCERN(I)) * B(JCERN(I)) * DELTA(I)
        D(I) = (MIG(JCERN(I)) + MIMAX(JCERN(I))) * DELTA(I)
      ELSE
        N(I) = TLIM * C(JCERN(I)) * B(JCERN(I)) * DELTA(I)
        D(I) = (MEG(JCERN(I)) + MEMAX(JCERN(I))) * DELTA(I)
      END IF
    ELSE
      IF WCERN$(I) = "INTR" THEN
        N(I) = -TLIM * C(JCERN(I)) * B(JCERN(I)) * DELTA(I)
        D(I) = (MIG(JCERN(I)) + MIMIN(JCERN(I))) * DELTA(I)
      ELSE
        N(I) = -CLIM * LARG * S(JCERN(I)) * B(JCERN(I)) * DELTA(I)
        D(I) = (MEG(JCERN(I)) + MEMIN(JCERN(I))) * DELTA(I)
      END IF
    END IF
  NEXT I
CASE "NON PROPORZIONALE"
  FOR I = 1 TO 4
    IF DELTA(I) > 0 THEN
      IF WCERN$(I) = "INTR" THEN
        N(I) = (CLIM * LARG * S(JCERN(I)) * B(JCERN(I)) - MIG(JCERN(I)))
          * DELTA(I)
        D(I) = MIMAX(JCERN(I)) * DELTA(I)
      ELSE

```

```

      N(I) = (TLIM * C(JCERN(I)) * B(JCERN(I)) - MEG(JCERN(I)))
        * DELTAF(I)
      D(I) = MEMAX(JCERN(I)) * DELTAF(I)
    END IF
  ELSE
    IF WCERN$(I) = "INTR" THEN
      N(I) = (-TLIM * C(JCERN(I)) * B(JCERN(I)) - MIG(JCERN(I))) *
        DELTAF(I)
      D(I) = MIMIN(JCERN(I)) * DELTAF(I)
    ELSE
      N(I) = (-CLIM * LARG * S(JCERN(I)) * B(JCERN(I)) -
        MEG(JCERN(I))) * DELTAF(I)
      D(I) = MEMIN(JCERN(I)) * DELTAF(I)
    END IF
  END IF
NEXT I
END SELECT
E1 = N(1) + N(2) + N(3) + N(4)
E2 = D(1) + D(2) + D(3) + D(4)
END SUB

SUB MomentiResidui
REM
REM *****
REM *
REM *           M o m e n t i R e s i d u i
REM *
REM *****
REM
REM *****
REM *
REM *           Questa subroutine calcola i momenti intradossali
REM *           ed estradossali
REM *
REM *****
REM
DIM A1(2, 2), B1(2), DELTAF(4), MRES(4)
REM -----
REM ROTAZIONI RELATIVE
REM -----
DELTAF(1) = F1
DELTAF(2) = F2 - F1

```

```

DELTA(4) = -F3
REM -----
REM                               INIZIO CALCOLO
REM -----
FOR I = 1 TO 4
  IF I = 3 THEN I = I + 1
  IF COLASSO$ = "PROPORZIONALE" THEN
    IF WCERN$(I) = "INTR" THEN      IF DELTA(I) > 0 THEN
      MRES(I) = CLIM * LARG * S(JCERN(I)) * B(JCERN(I)) - GAMMA *
        (MIG(JCERN(I)) + MIMAX(JCERN(I)))
    ELSE
      MRES(I) = -TLIM * C(JCERN(I)) * B(JCERN(I)) - GAMMA *
        (MIG(JCERN(I)) + MIMIN(JCERN(I)))
    END IF      ELSE      IF DELTA(I) > 0 THEN
      MRES(I) = TLIM * C(JCERN(I)) * B(JCERN(I)) - GAMMA *
        (MEG(JCERN(I)) + MEMAX(JCERN(I)))
    ELSE
      MRES(I) = -CLIM * LARG * S(JCERN(I)) * B(JCERN(I)) - GAMMA *
        (MEG(JCERN(I)) + MEMIN(JCERN(I)))
    END IF
  END IF
ELSE
  IF WCERN$(I) = "INTR" THEN
    IF DELTA(I) > 0 THEN
      MRES(I) = CLIM * LARG * S(JCERN(I)) * B(JCERN(I)) -
        MIG(JCERN(I)) - GAMMA * MIMAX(JCERN(I))
    ELSE
      MRES(I) = -TLIM * C(JCERN(I)) * B(JCERN(I)) - MIG(JCERN(I)) -
        GAMMA * MIMIN(JCERN(I))
    END IF      ELSE      IF DELTA(I) > 0 THEN
      MRES(I) = TLIM * C(JCERN(I)) * B(JCERN(I)) - MEG(JCERN(I)) -
        GAMMA * MEMAX(JCERN(I))
    ELSE
      MRES(I) = -CLIM * LARG * S(JCERN(I)) * B(JCERN(I)) -
        MEG(JCERN(I)) - GAMMA * MEMIN(JCERN(I))
    END IF      END IF      END IF NEXT I
B1(1) = MRES(1) - MRES(4)
B1(2) = MRES(2) - MRES(4)
A1(1, 1) = YCERN(4) - YCERN(1)
A1(1, 2) = ZCERN(1) - ZCERN(4)
A1(2, 1) = YCERN(4) - YCERN(2)
A1(2, 2) = ZCERN(2) - ZCERN(4)

```

526 *Le strutture ad arco*

```

DET = A1(1, 1) * A1(2, 2) - A1(1, 2) * A1(2, 1)
H2 = (B1(1) * A1(2, 2) - B1(2) * A1(1, 2)) / DET
V2 = -(B1(1) * A1(2, 1) - B1(2) * A1(1, 1)) / DET
A2 = MRES(4)
A2 = A2 + V2 * (LUCE - ZCERN(4)) + H2 * (YCERN(4) + RIAL)
FOR I = 1 TO T + 1
  N = -V2 * SIN(INCL(I)) + H2 * COS(INCL(I))
  M = A2 - V2 * (LUCE - LTRAT * (I - 1)) + H2 * (Q(I) - RIAL)
  R1(I) = M - N * B(I) / 2
  R2(I) = M + N * B(I) / 2
NEXT I
END SUB
SUB UscitaDati
REM
REM *****
REM *
REM *
REM *           U s c i t a D a t i
REM *
REM *****
REM
REM *****
REM *
REM *           Questa subroutine si occupa dell'uscita, su video, su stampa
REM *           e su file, dei dati di ingresso
REM *
REM *****
FOR ITER = 1 TO ITER1
  PRINT #ITER, " ===== "
  PRINT #ITER, "PROGRAMMA AR2 - ANALISI LIMITE INCREMENTALE
  PRINT #ITER, TITLE$
  PRINT #ITER, AUMENTO DEI CARICHI "; COLLASSO$
  PRINT #ITER, " ===== "
  PRINT #ITER,
  PRINT #ITER,
  PRINT #ITER,
  PRINT #ITER, "FILE DEI DATI ="; FILEINPUT$
  PRINT #ITER, "LUCE DELL'ARCO = "; LUCE
  PRINT #ITER, "FRECCIA DELL'ARCO = "; FREC
  PRINT #ITER, "RIALZO A DESTRA ="; RIAL
  PRINT #ITER, "LARGHEZZA DELLA SEZIONE ="; LARG
  PRINT #ITER, "NUMERO DEI TRATTI DI DIVISIONE = "; T
  PRINT #ITER, "TENSIONE LIMITE NELL'ACCIAIO = "; TLIM

```

```

PRINT #ITER, "COMPRESSIONE LIMITE NEL CONGLOMERATO = "; CLIM
PRINT #ITER, "ALTEZZA DELLA SEZIONE A SINISTRA = "; HS
PRINT #ITER, "ALTEZZA DELLA SEZIONE AL CENTRO = "; HC
PRINT #ITER, "ALTEZZA DELLA SEZIONE A DESTRA = "; HD
PRINT #ITER, "SPESSORE DELLA SOLETTA A SINISTRA = "; SPES
PRINT #ITER, "SPESSORE DELLA SOLETTA AL CENTRO = "; SPEC
PRINT #ITER, "SPESSORE DELLA SOLETTA A DESTRA = "; SPED
PRINT #ITER, "AREA DEL FERRO A SINISTRA = "; FERS
PRINT #ITER, "AREA DEL FERRO AL CENTRO = "; FERC
PRINT #ITER, "AREA DEL FERRO A DESTRA = "; FERD
PRINT #ITER, "RAPPORTO MASSIMO SUL TEST DI USCITA "; MAXR
PRINT #ITER, "DIVIDENTI DELLE CERNIERE DI TENTATIVO = ";
    JCERN(1), JCERN(2), JCERN(3), JCERN(4)
FOR I = 1 TO 4
    IF WCERN$(I) = "INTR" THEN
        PRINT #ITER, "LA CERNIERA "; I; " E' ALL'INTRADOSSO"
    ELSE
        PRINT #ITER, "LA CERNIERA "; I; " E' ALL'ESTRADOSSO"
    END IF
NEXT I
PRINT #ITER,
IF CARICHI$ = "FUNC" THEN
    PRINT #ITER, "CARICO VERTICALE DA PESO PROPRIO A SINISTRA =
"; QPROPS
    PRINT #ITER, "CARICO VERTICALE DA PESO PROPRIO AL CENTRO =
"; QPROPC
    PRINT #ITER, "CARICO VERTICALE DA PESO PROPRIO A DESTRA =
"; QPROPD
    PRINT #ITER, "CARICO VERTICALE ACCIDENTALE A SINISTRA = ";
    QACCVS
    PRINT #ITER, "CARICO VERTICALE ACCIDENTALE A SINISTRA = ";
    QACCVS
    PRINT #ITER, "CARICO VERTICALE ACCIDENTALE A SINISTRA = ";
    QACCVS
    PRINT #ITER, "CARICO ORIZZONTALE ACCIDENTALE A SINISTRA =
"; QACCOS
    PRINT #ITER, "CARICO ORIZZONTALE ACCIDENTALE AL CENTRO =
"; QACCOC
    PRINT #ITER, "CARICO ORIZZONTALE ACCIDENTALE A DESTRA = ";
    QACCOD
END IF
REM _____

```

```

REM STAMPA LE QUOTE DELL'ARCO PER CIASCUNA DIVIDENTE
REM -----
IF ASSE$ = "DATI" THEN
  PRINT #ITER,
  PRINT #ITER, " ===== "
  PRINT #ITER, " = "
  PRINT #ITER, " =          TABELLA DELLE QUOTE DELL'ARCO          = "
  PRINT #ITER, " = "
  PRINT #ITER, " ===== "
  PRINT #ITER,
  PRINT #ITER, " ----- "
  PRINT #ITER, " DIVIDENTE          QUOTA          "
  PRINT #ITER, " ----- "
  PRINT #ITER,
  FOR I = 1 TO T + 1
    PRINT #ITER, TAB(1); I; TAB(34); Q(I)
  NEXT I
END IF
REM -----
REM STAMPA I MOMENTI INTRADOSSALI MINIMI, MASSIMI
REM E DA PESO PROPRIO
REM -----
PRINT #ITER,
PRINT #ITER, " ===== "
PRINT #ITER, " = "
PRINT #ITER, " =          MOMENTI INTRADOSSALI          = "
PRINT #ITER, " = "
PRINT #ITER, " ===== "
PRINT #ITER,
PRINT #ITER, " ----- "
PRINT #ITER, " DIVIDENTE  MINIMI  MASSIMI  PESO PROPRIO  "
PRINT #ITER, " ----- "
PRINT #ITER,
FOR I = 1 TO T + 1
  PRINT #ITER, TAB(2); I; TAB(9); MIMIN(I); TAB(25); MIMAX(I); TAB(45);
  MIG(I)
NEXT I
REM -----
REM STAMPA I MOMENTI ESTRADOSSALI MINIMI, MASSIMI
REM E DA PESO PROPRIO
REM -----
PRINT #ITER,

```



```

PRINT #ITER, " ===== "
PRINT #ITER, " = "
PRINT #ITER, " =          MOMENTI ESTRADOSSALI          = "
PRINT #ITER, " = "
PRINT #ITER, " ===== "
PRINT #ITER,
PRINT #ITER, " _____ "
PRINT #ITER, " DIVIDENTE  MINIMI  MASSIMI  PESO PROPRIO  "
PRINT #ITER, " _____ "
PRINT #ITER,
FOR I = 1 TO T + 1
PRINT #ITER, TAB(2); I; TAB(9); MEMIN(I); TAB(25); MEMAX(I);
      TAB(45); MEG(I)
NEXT I
PRINT #ITER,
NEXT ITER
END SUB

SUB UscitaRisultati
REM
REM *****
REM *
REM *          U s c i t a R i s u l t a t i          *
REM *
REM *****
REM
REM *****
REM *
REM *          Questa subroutine si occupa dell'uscita (su video,
REM *          su stampa e su file) dei risultati
REM *
REM *****
REM
REM _____
REM CALCOLA SE LA CRISI AVVIENE PER SNERVAMENTO DEL FERRO
REM O PER SCHIACCIAMENTO DEL CONGLOMERATO
REM _____
IF WCERN$(1) = "INTR" THEN
  IF F1 = -1 THEN CRISIS$(1) = "FERRO" ELSE CRISIS$(1) = "CONGLOMER-
    ATO"
ELSE

```

```

    IF F1 = -1 THEN CRISI$(1) = "CONGLOMERATO" ELSE
      CRISI$(1) = "FERRO"
END IF
IF WCERN$(2) = "INTR" THEN
  IF F2 - F1 < 0 THEN CRISI$(2) = "FERRO" ELSE CRISI$(2) = "CONGLOMERATO"
ELSE
  IF F2 - F1 < 0 THEN CRISI$(2) = "CONGLOMERATO" ELSE CRISI$(2) = "FERRO"
END IF
IF WCERN$(3) = "INTR" THEN
  IF F3 - F2 < 0 THEN CRISI$(3) = "FERRO" ELSE CRISI$(3) = "CONGLOMERATO"
ELSE
  IF F3 - F2 < 0 THEN CRISI$(3) = "CONGLOMERATO" ELSE CRISI$(3) = "FERRO"
END IF
IF WCERN$(4) = "INTR" THEN
  IF F3 > 0 THEN CRISI$(4) = "FERRO" ELSE CRISI$(4) = "CONGLOMERATO"
ELSE
  IF F3 > 0 THEN CRISI$(4) = "CONGLOMERATO" ELSE
    CRISI$(4) = "FERRO"
END IF
FOR ITER = ITER1 + 1 TO ITER2
  PRINT #ITER, "MULTIPLICATORE STATICAMENTE AMMISSIBILE = ";
  PSI
  PRINT #ITER, "MULTIPLICATORE CINEMATICAMENTE SUFFICIENTE = ";
  GAMMA
  PRINT #ITER,
  PRINT #ITER, "CERNIERA ASCISSA LOCAZIONE CRISI"
  PRINT #ITER,
  FOR I = 1 TO 4
    AA = LTRAT * (JCERN(I) - 1)
    PRINT #ITER, TAB(2); JCERN(I); TAB(12); AA; TAB(29); WCERN$(I);
    TAB(45); CRISI$(I)
  NEXT I
  PRINT #ITER,
  PRINT #ITER, "MOMENTI INTRADOSSALI MAX MOMENTI INTRADOSSALI MIN"
  PRINT #ITER,
  FOR K = 1 TO T + 1

```

```

    PRINT #ITER, TAB(2); K; TAB(8); MIMAX(K); TAB(38); MIMIN(K)
NEXT K
PRINT #ITER,
PRINT #ITER, "MOMENTI ESTRADOSSALI MAX MOMENTI ESTRADOS-
    SALI MIN"
PRINT #ITER,
FOR K = 1 TO T + 1
    PRINT #ITER, TAB(2); K; TAB(8); MEMAX(K); TAB(38); MEMIN(K)
NEXT K
PRINT #ITER,
PRINT #ITER, "MOMENTI INTRADOSSALI MOMENTI INTRADOSSALI "
PRINT #ITER, "DA PESO PROPRIO DA PESO PROPRIO"
PRINT #ITER,
FOR K = 1 TO T + 1
    PRINT #ITER, TAB(2); K; TAB(8); MIG(K); TAB(38); MEG(K)
NEXT K
PRINT #ITER,
PRINT #ITER, "MOMENTI INTRADOSSALI RES. MOMENTI INTRADOS-
    SALI RES."
PRINT #ITER,
FOR K = 1 TO T + 1
    PRINT #ITER, TAB(2); K; TAB(8); R1(K); TAB(38); R2(K)
NEXT K
PRINT #ITER,
PRINT #ITER, "MOMENTI LIMITI FERRO MOMENTI LIMITI CONGLOM-
    ERATO"
PRINT #ITER,
FOR K = 1 TO T + 1
    PRINT #ITER, TAB(2); K; TAB(8); TLIM * C(K) * B(K); TAB(35); CLIM *
        LARG * S(K) * B(K)
NEXT K
PRINT #ITER,
PRINT #ITER, "TEMPO DI ESECUZIONE = "; TIME2; " SEC."
PRINT #ITER, "RUN DEL "; DATE$; " ORE "; TIME$
NEXT ITER
END SUB

```