

```

=====
PROGRAM Triang_sup
=====
IMPLICIT NONE
! Questo programma risolve un sistema triangolare superiore facendo
! uso delle subroutines
!   Leggimat (lettura matrici)
!   Scrivimat (scrittura matrici)
!   Trisup (risoluzione di un sistema triangolare superiore).
!
! Il programma legge da file la matrice aumentata del sistema (A|b) di
! dimensioni massime (dimMax, dimMax+1) e scrive
! sia su file sia a video la matrice A, il vettore dei termini noti b ed il
! vettore soluzione x del sistema triangolare Ax = b.
!
! Vengono richiesti all'utente il nome del file da cui leggere la matrice,
! il nome del file su cui scrivere i risultati e la dimensione del sistema.
!
! Viene effettuato un controllo sugli elementi diagonali della matrice,
! per verificare che il sistema triangolare sia determinato

=====
! Parte dichiarativa
=====
INTEGER, PARAMETER :: dimMax=100
REAL(8), DIMENSION(dimMax,dimMax+1) :: A
REAL(8), DIMENSION(dimMax) :: b, x
INTEGER :: n, i
REAL(8) :: diag
CHARACTER(50) :: nomefilein, nomefileout
INTEGER :: unitIn=21, unitOut=22, ioerror

=====
! Parte esecutiva
=====
WRITE(*,*) 'Risoluzione di un sistema triangolare superiore.'
WRITE(*,*) ' '

! Gestione files di input e di output
WRITE(*,*) 'Inserire il nome del file di input: '
READ(*,*) nomefilein

OPEN(UNIT=unitIn,FILE=nomefilein,STATUS='OLD',ACTION='READ',IOSTAT=ioerror)
IF (ioerror /= 0) THEN
    WRITE(*,*) 'Errore apertura file ', nomefilein
    WRITE(*,*) 'errore ', ioerror
    STOP
END IF
WRITE(*,*) ' '

WRITE(*,*) 'Inserire il nome del file di output: '
READ(*,*) nomefileout

OPEN(UNIT=unitOut,FILE=nomefileout,STATUS='REPLACE',ACTION='WRITE',&
    IOSTAT=ioerror)
IF (ioerror /= 0) THEN
    WRITE(*,*) 'Errore apertura file ', nomefileout
    WRITE(*,*) 'errore ', ioerror
    STOP
END IF
WRITE(*,*) ' '

! Inserimento dimensione del sistema
n = dimMax+1
DO WHILE (n>=dimMax .OR. n<=0)
    WRITE(*,*) 'Inserire la dimensione del sistema: '
    READ(*,*) n
END DO

! Lettura della matrice aumentata
CALL Leggimat(A, n, n+1, dimMax, unitIn)

! Creazione del vettore termini noti
DO i=1,n
    b(i)=A(i,n+1)
END DO

! Stampa a video di verifica della matrice letta

```

```

! e del vettore termine noto
! (8 cifre significative)
WRITE(*,*) 'Matrice del sistema:'
CALL Scrivimat(A, n, n, dimMax, 6, 8)
WRITE(*,*) ' '
WRITE(*,*) 'Vettore dei termini noti:'
CALL Scrivimat(b, n, 1, dimMax, 6, 8)
WRITE(*,*) ' '

! Stampa su file di verifica della matrice letta
! e del vettore termine noto
! (8 cifre significative)
WRITE(unitOut,*) 'Matrice del sistema:'
CALL scrivimat(A, n, n, dimMax, unitOut, 8)
WRITE(unitOut,*) ' '
WRITE(unitOut,*) 'Vettore dei termini noti:'
CALL scrivimat(b, n, 1, dimMax, unitOut, 8)
WRITE(unitOut,*) ' '

! Verifica della singolarita' della matrice triangolare
! (almeno un elemento nullo sulla diagonale principale)
diag = 1.0d0
DO i=1,n
    diag=diag*A(i,i)
END DO

IF (diag==0.0d0) THEN
    WRITE(*,*) 'Impossibile proseguire: elementi nulli sulla diagonale!'
    WRITE(unitOut,*) 'Impossibile proseguire: elementi nulli sulla diagonale!'
ELSE
    ! Risoluzione del sistema triangolare superiore
    CALL trisup (A, b, n, dimMax, x)

    ! Stampa a video e su file di verifica della soluzione
    ! (16 cifre significative)
    WRITE(*,*) 'Vettore soluzione:'
    CALL Scrivimat(x, n, 1, dimMax, 6, 16)
    WRITE(*,*) ' '
    WRITE(unitOut,*) 'Vettore soluzione:'
    CALL Scrivimat(x, n, 1, dimMax, unitOut, 16)
    WRITE(unitOut,*) ' '

END IF
! Chiusura files
CLOSE (unitIn)
CLOSE (unitOut)

```

```

END PROGRAM Triang_sup

```

```

!=====
! Subroutine per la LETTURA di una matrice a n righe ed m colonne
! dall'unita' identificata dall'intero io
!=====
SUBROUTINE Leggimat(A, n, m, maxrighe,io)
  IMPLICIT NONE
  !
  ! Variabili INGRESSO:
  !   n      numero effettivo di righe di A
  !   m      numero effettivo di colonne di A
  !   maxrighe numero massimo di righe di A (come da dimensionamento
  !           del programma principale)
  !   io     numero dell'unita' da cui deve leggere la matrice
  !
  ! Variabili USCITA:
  !   A      matrice di n righe ed m colonne che viene letta e
  !           memorizzata
  !
  !=====
  ! Parte dichiarativa
  !=====
  REAL(8), DIMENSION(maxrighe,m), INTENT(OUT) :: A
  INTEGER, INTENT(IN) :: n, m, maxrighe, io
  INTEGER :: i, j
  !=====
  ! Parte esecutiva
  !=====

```

```

DO i=1,n
  READ(io,*) (A(i,j), j=1,m)
END DO

```

```

END SUBROUTINE Leggimat

```

```

!=====
! Subroutine per la SCRITTURA di una matrice a n righe ed m colonne
! sull'unita` identificata dall'intero io
!=====
SUBROUTINE Scrivimat(A, n, m, maxrighe, io, cifreSign)
  IMPLICIT NONE
!
! Variabili INGRESSO:
!   A      matrice n righe ed m colonne
!   n      numero di righe di A
!   m      numero di colonne di A
!   maxrighe numero massimo di righe di A (come da dimensionamento
!           del programma principale)
!   io     numero dell'unita' da cui deve scrivere la matrice
!   cifreSign Parametro per scegliere il numero di cifre significative
!           che si desidera per la stampa.
!           Se cifreSign==16 si visualizzano 16 cifre decimali,
!           altrimenti se ne visualizzano solo 8
!

```

```

!=====
! Parte dichiarativa
!=====

```

```

REAL(8), DIMENSION(maxrighe,m), INTENT(IN) :: A
INTEGER, INTENT(IN) :: n, m, maxrighe, io, cifreSign
INTEGER :: i, j

```

```

!=====
! Parte esecutiva
!=====

```

```

IF (cifreSign==16) THEN
  DO i=1,n
    WRITE(io,100) (A(i,j), j=1,m)
    100 FORMAT (1000G25.16)
  END DO
ELSE
  DO i=1,n
    WRITE(io,200) (A(i,j), j=1,m)
    200 FORMAT (1000G17.8)
  END DO
END IF

```

```

END SUBROUTINE Scrivimat

```

```

!=====
! Subroutine per la RISOLUZIONE di un sistema triangolare superiore
! non singolare con matrice U di ordine n
!=====

```

```

SUBROUTINE Trisup (U, b, n, maxrighe, x)
  IMPLICIT NONE
!
! Variabili INGRESSO:
!   U      matrice quadrata triangolare superiore di ordine n
!   b      vettore di n componenti, termine noto del sistema
!   n      numero effettivo di righe e di colonne di U
!   maxrighe numero massimo di righe di U (come da dimensionamento
!           del programma principale)
!

```

```

! Variabili USCITA:
!   x      vettore di n componenti, soluzione del sistema
!

```

```

!=====
! Parte dichiarativa
!=====

```

```

REAL(8), DIMENSION(maxrighe,n), INTENT(IN) :: U
REAL(8), DIMENSION(n), INTENT(IN) :: b
REAL(8), DIMENSION(n), INTENT(OUT) :: x
INTEGER, INTENT(IN) :: n, maxrighe
INTEGER :: i, j

```

```

DO i=n,1,-1
  x(i)=b(i)

```

```
DO j=i+1,n
  x(i)=x(i)-u(i,j)*x(j)
END DO
x(i)=x(i)/u(i,i)
END DO
END SUBROUTINE Trisup
```