

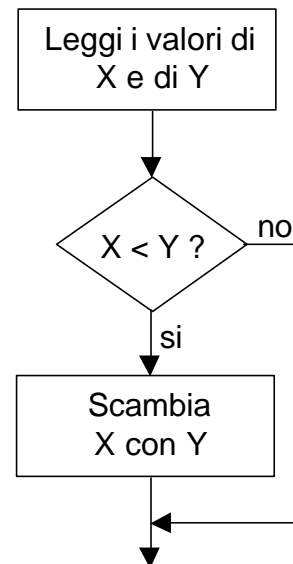
Istruzioni selettive: IF...THEN

Sintassi

```
IF (condizione) THEN
    istruzione_1
    istruzione_2
    ...
    istruzione_n
END IF
```

eseguite solo se *condizione* è TRUE

Esempio



```
INTEGER :: x,y,appo
READ(*,*) x,y
IF (x < y) THEN
    ! scambia x e y
    appo=x
    x=y
    y=appo
END IF
```

Istruzioni selettive: IF logico

Sintassi

IF (*condizione*) *istruzione*

Esempio

```
INTEGER :: x
```

```
READ( *, *) x
```

```
IF (x < 0) x=0
```

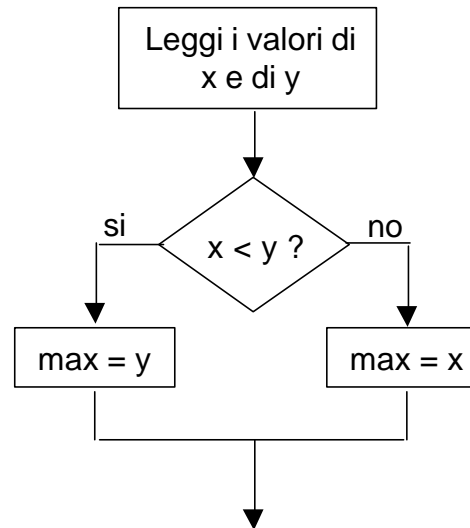
Istruzioni selettive: IF...THEN...ELSE

Sintassi

```
IF (condizione) THEN
    istruzione_1
    istruzione_2 } eseguite solo se condizione è TRUE
ELSE
    istruzione_3
    istruzione_4 } eseguite solo se condizione è FALSE
END IF
```

Esempio

Qual è il massimo tra x e y ?



```
REAL :: x,y,max
READ(*,*) x,y
IF (x < y) THEN
    max=y
ELSE
    max=x
END IF
```

Istruzioni selettive: IF...THEN...ELSE IF...ELSE

Sintassi

```
IF (condizione_1) THEN  
    blocco_1
```

← eseguito solo se *condizione_1* è TRUE

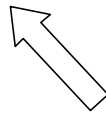
```
ELSE IF (condizione_2) THEN  
    blocco_2
```

← eseguito solo se *condizione_1* è FALSE
e *condizione_2* è TRUE

```
ELSE IF (condizione_3) THEN  
    blocco_3
```

← eseguito solo se *condizione_1* è FALSE,
condizione_2 è FALSE e
condizione_3 è TRUE

```
ELSE  
    blocco_4  
END IF
```



eseguito solo se *condizione_1* è FALSE,
condizione_2 è FALSE e
condizione_3 è FALSE

Esempio

```
INTEGER :: voto
READ(*,*) voto
IF (voto < 18) THEN
    WRITE(*,*) "Ritorna"
ELSE IF (voto < 24) THEN
    WRITE(*,*) "Si puo' dare di piu'"
ELSE IF (voto < 27) THEN
    WRITE(*,*) "Non c'e' male"
ELSE IF (voto < 30) THEN
    WRITE(*,*) "C'e' mancato poco"
ELSE IF (voto == 30) THEN
    WRITE(*,*) "Finalmente ci siamo"
ELSE
    WRITE(*,*) "WOW !"
END IF
```

Problema

Dati tre numeri reali x , y e z , stabilire se sono le lunghezze dei lati di un triangolo e, in caso affermativo, stabilire che tipo di triangolo sia (equilatero, isoscele o scaleno)

```

PROGRAM TRIANGOLO
  ! Variabili
  REAL:: x,y,z

  ! Istruzioni
  WRITE(*,*) "Dammi i tre valori:"
  READ(*,*) x,y,z

  IF((x+y>z).AND.(x+z>y).AND.(y+z>x)) THEN
    ! E' un triangolo. Stampo messaggio
    WRITE(*,*) "E' un triangolo"
    ! Quale tipo di triangolo?
    ! Verifica del tipo di triangolo
  ELSE
    ! Non e' un triangolo. Stampo messaggio
    WRITE(*,*) "Non e' un triangolo"
  END IF

  WRITE(*,*) "Premi un tasto per continuare..."
  READ(*,*)

END PROGRAM

```

```

PROGRAM TRIANGOLO
  ! Variabili
  REAL:: x,y,z

  ! Istruzioni
  WRITE(*,*) "Dammi i tre valori:"
  READ(*,*) x,y,z

  IF((x+y>z).AND.(x+z>y).AND.(y+z>x)) THEN
    ! E' un triangolo. Stampo messaggio
    WRITE(*,*) "E' un triangolo"
    ! Quale tipo di triangolo?

    IF((x==y).AND.(y==z)) THEN
      ! E' un triangolo equilatero. Stampo messaggio
      WRITE(*,*) "E' un triangolo equilatero"
    ELSE IF((x==y).OR.(y==z).OR.(x==z)) THEN
      ! E' un triangolo isoscele. Stampo messaggio
      WRITE(*,*) "E' un triangolo isoscele"
    ELSE
      ! E' un triangolo scaleno. Stampo messaggio
      WRITE(*,*) "E' un triangolo scaleno"
    END IF

  ELSE
    ! Non e' un triangolo. Stampo messaggio
    WRITE(*,*) "Non e' un triangolo"
  END IF

  WRITE(*,*) "Premi un tasto per continuare..."
  READ(*,*)

END PROGRAM

```



```
F:\fortran>fexec F:\fortran\triang
Dammi i tre valori:
3.0 2.0 9.5
Non e' un triangolo
Premi un tasto per continuare...
```

```
F:\fortran>fexec F:\fortran\triang
Dammi i tre valori:
3.0 3.0 3.0
E' un triangolo
E' un triangolo equilatero
Premi un tasto per continuare...
```

```
F:\fortran>fexec F:\fortran\triang
Dammi i tre valori:
2.0 2.0 3.0
E' un triangolo
E' un triangolo isoscele
Premi un tasto per continuare...
```

```
F:\fortran>fexec F:\fortran\triang
Dammi i tre valori:
3.000000 3.000001 3.000002
E' un triangolo
E' un triangolo scaleno
Premi un tasto per continuare...
```

Istruzioni selettive: SELECT...CASE

Sintassi

```
CASE SELECT (espressione)  
  CASE (selettore_1)  
    blocco_1  
  CASE (selettore_2)  
    blocco_2  
  CASE (selettore_3)  
    blocco_3  
  CASE DEFAULT  
    blocco_4  
END SELECT
```

espressione è di tipo intero

il *selettore* può indicare diverse condizioni:

min: TRUE se *espressione* \geq min

min:max TRUE se min \leq *espressione* \leq max

:max TRUE se *espressione* \leq max

val TRUE se *espressione* = val

Esempio

```
INTEGER:: voto
```

```
WRITE(*,*) "Voto ottenuto:"
```

```
READ(*,*) voto
```

```
SELECT CASE (voto)
```

```
  CASE(0:17)
```

```
    WRITE(*,*) "Ritorna"
```

```
  CASE(18:24)
```

```
    WRITE(*,*) "Si puo' dare di piu' "
```

```
  CASE(25:27)
```

```
    WRITE(*,*) "Non c'e' male"
```

```
  CASE(28:29)
```

```
    WRITE(*,*) "C'e' mancato poco"
```

```
  CASE(30)
```

```
    WRITE(*,*) "Finalmente ci siamo"
```

```
  CASE(31:)
```

```
    WRITE(*,*) "WOW !"
```

```
  CASE DEFAULT
```

```
    WRITE(*,*) "Voto non valido"
```

```
END SELECT
```