

# Istruzioni per il controllo di ciclo - ciclo a conteggio

Permette di ripetere l'esecuzione di un blocco di istruzioni per un numero predefinito di volte.

## Sintassi

```
DO var_ciclo=espr_1,espr_2,espr_3
  istruzione_1
  istruzione_2
  ...
  istruzione_n
END DO
```

*var\_ciclo* è una variabile numerica  
*espr\_1*, *espr\_2* ed *espr\_3* sono espressioni  
numeriche dello stesso tipo di *var\_ciclo*

## Esempio

stampare i numeri interi  
pari da 0 a 10

```
INTEGER :: i
DO i=0,10,2
  WRITE(*,*) i
END DO
```

## Note

Il numero di iterazioni è valutato prima dell'esecuzione del ciclo ed è uguale a

$\text{MAX}(\text{INT}((\text{expr}_2 - \text{expr}_1 + \text{expr}_3) / \text{expr}_3), 0)$

La variabile di ciclo assume i valori compresi tra  $\text{expr}_1$  ed  $\text{expr}_2$  incrementandosi di  $\text{expr}_3$  ad ogni iterazione. Se  $\text{expr}_3$  è assente, viene assunto uguale a 1.

Se il numero di iterazioni è  $\leq 0$ , le istruzioni in ciclo non vengono mai eseguite.

Il ciclo può realizzare anche conteggi decrescenti. In tal caso,  $\text{expr}_2 < \text{expr}_1$  ed  $\text{expr}_3 < 0$ .

## Esempio

```
! 6 iterazioni
DO i=1,36,7
...
END DO
```

```
! 0 iterazioni
DO i=6,5
...
END DO
```

```
! 4 iterazioni
DO i=10,-3,-4
...
END DO
```

```

PROGRAM Prova_DO
  IMPLICIT NONE
  ! Dichiarazioni
  INTEGER :: indice,c
  INTEGER :: inizio,fine,incremento

  ! Prova ciclo DO
  ! Input
  WRITE(*,*) "Valore iniziale: "
  READ(*,*) inizio
  WRITE(*,*) inizio
  WRITE(*,*) "Valore finale: "
  READ(*,*) fine
  WRITE(*,*) fine
  WRITE(*,*) "Incremento: "
  READ(*,*) incremento
  WRITE(*,*) incremento

  c=0
  DO indice=inizio,fine,incremento
    c=c+1
    WRITE(*,*)
    WRITE(*,*) "Ciclo: ",c
    WRITE(*,*) "Indice: ",indice
    WRITE(*,*)
  END DO

  WRITE(*,*) "Numero cicli effettuati: ",c
  WRITE(*,*) "Indice all'uscita dal ciclo: ",indice
  WRITE(*,*)

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

END PROGRAM

```

## **Problema**

Stampare la “tabellina” di un valore dato in input.

Stampare le “tabelline” dei valori compresi tra 1 e 10.

```
PROGRAM Tabellina
  IMPLICIT NONE
  ! Dichiarazioni
  INTEGER :: i,val

  ! Input
  WRITE(*,*) "Quale tabellina vuoi stampare ?"
  READ(*,*) val
  WRITE(*,*) val

  WRITE(*,*)
  WRITE(*,*) "Tabellina del ",val, ":"
  DO i=1,10
    WRITE(*,*)
    WRITE(*,*) val," x",i," = ",val*i
    WRITE(*,*)
  END DO

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

END PROGRAM
```

Quale tabellina vuoi stampare ?  
7

Tabellina del 7:

$$7 \times 1 = 7$$

$$7 \times 2 = 14$$

$$7 \times 3 = 21$$

$$7 \times 4 = 28$$

$$7 \times 5 = 35$$

$$7 \times 6 = 42$$

$$7 \times 7 = 49$$

$$7 \times 8 = 56$$

$$7 \times 9 = 63$$

$$7 \times 10 = 70$$

Premi un tasto per continuare...

```
PROGRAM Tabellina_2
  IMPLICIT NONE
  ! Dichiarazioni
  INTEGER :: i,val

  ! Input

  DO val=1,10
    ! Stampa la "tabellina" del val
  END DO

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

END PROGRAM
```

```
PROGRAM Tabellina_2
  IMPLICIT NONE
  ! Dichiarazioni
  INTEGER :: i, val

  ! Input

  DO val=1,10

    WRITE(*,*)
    WRITE(*,*) "Tabellina del ",val, ":"
    DO i=1,10
      WRITE(*,*)
      WRITE(*,*) val, " x", i, " = ", val*i
    END DO

  END DO

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

END PROGRAM
```



Tabellina del 1:

$$1 \times 1 = 1$$

$$1 \times 2 = 2$$

$$1 \times 3 = 3$$

$$1 \times 4 = 4$$

$$1 \times 5 = 5$$

$$1 \times 6 = 6$$

$$1 \times 7 = 7$$

$$1 \times 8 = 8$$

$$1 \times 9 = 9$$

$$1 \times 10 = 10$$

Tabellina del 2:

$$2 \times 1 = 2$$

$$2 \times 2 = 4$$

$$2 \times 3 = 6$$

$$2 \times 4 = 8$$

$$2 \times 5 = 10$$

$$2 \times 6 = 12$$

$$2 \times 7 = 14$$

$$2 \times 8 = 16$$

$$2 \times 9 = 18$$

$$2 \times 10 = 20$$

. . .