

# Programmazione in Fortran:

## Lezione 3

A.A. 2009/2010

Ing. A . Siviglia

nunzio.siviglia@ing.unitn.it

stanza:

Laboratorio didattico di modellistica idrodinamica (2° piano)

Tel 2440



# CICLI A CONTEGGIO

Permette di ripetere l'istruzione per un numero definito di volte:

Sintassi:

```
DO iter_ciclo=expr_1,expr_2,exp_3
```

```
  Istruzione_1
```

```
  Istruzione_2
```

```
  .
```

```
  .
```

```
  .
```

```
  Istruzione_n
```

```
END DO
```

**iter\_ciclo**                      è una variabile numerica

**expr\_1, expr\_2, expr\_3**    sono variabili dello stesso tipo di iter\_ciclo



# CICLI A CONTEGGIO

Il numero di iterazioni è calcolato prima dell'esecuzione del ciclo ed è pari a:

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

Il ciclo parte da  $\text{expr}_1$  e prosegue fino a  $\text{expr}_2$ , incrementandosi di  $\text{expr}_3$

Se il numero di iterazioni è  $< 0$ , le istruzioni all'interno del ciclo non vengono mai eseguite.

Il ciclo può effettuare conteggi decrescenti se  $\text{expr}_2 < \text{expr}_1$

Esempi:

! 5 iterazioni

```
DO i=4,24,5  
  write(*,*) i  
END DO
```

! 0 iterazioni

```
DO i=7,5  
  write(*,*) i  
END DO
```

! 3 iterazioni

```
DO i=12,6,-3  
  write(*,*) i  
END DO
```



# CICLI A CONTEGGIO

```
DO i=1,10, 2
```

```
    WRITE(*,*) i
```

```
END DO
```

```
WRITE(*,*) 'Fine del ciclo'
```

Risultato:

1

3

5

7

9

Fine del ciclo

```
DO i=1,5
```

```
    WRITE(*,*) i
```

```
END DO
```

```
WRITE(*,*) 'Fine del ciclo'
```

Risultato:

1

2

3

4

5

Fine del ciclo



# CICLI A CONTEGGIO

## ESERCIZIO

Stampare a video le tabelline dei numeri da 1 a 10.



# Cicli condizionati

## DO ... ENDDO

[ < nome > : ] **DO**

< blocco\_istr\_1 >

**IF** ( < espr\_logica\_1 > ) **EXIT** [ < nome > ]

[ < blocco\_istr\_2 > ]

[ **IF** ( < espr\_logica\_2 > ) **CYCLE** [ < nome > ]

[ < blocco\_istr\_3 > ]

**END DO** [ < nome > ]



## Esempio

### EXIT

```
DO i=1,5  
    IF( i == 3) EXIT  
    WRITE(*,*) i  
END DO  
WRITE(*,*) 'Fine del ciclo'
```

Risultato:

1

2

Fine del ciclo

### CYCLE

```
DO i=1,5  
    IF( i == 3) CYCLE  
    WRITE(*,*) i  
END DO  
WRITE(*,*) 'Fine del ciclo'
```

Risultato:

1

2

4

5

Fine del ciclo



## Esercizio

Scrivere un codice che calcola la media di  $n$  numeri reali positivi (al massimo 50) inseriti da tastiera.

Caso (1) Se si trova il numero -999, vuol dire che tutti i numeri sono stati inseriti e si deve interrompere il calcolo  
(usare EXIT);

Caso (2) Se si trova un numero negativo lo si scarta  
(usare CYCLE)





## Procedure intrinseche

Il Fortran 90 ha 113 procedure intrinseche (generalmente funzioni) appartenenti a diverse classi:

1. Di Conversione (INT, DBLE ...)
2. Matematiche (SIN , LOG ...)
3. Numeriche (SUM , CEILING .. )
4. Di tipo carattere ( INDEX , TRIM ... )
5. Per la manipolazione di bit (IAND , IOR ... )
6. Di indagine (ALLOCATED, SIZE ...)
7. Di trasformazione (REAL, TRANSPOSE ..)
8. Diverse Funzioni e Subroutine non elementari come  
SYSTE\_CLOCK o DATE\_AND\_TIME



# UTILIZZO DELLE FUNZIONI INTRINSECHE

- Le funzioni intrinseche possono essere usate ovunque sia richiesta all'interno di una espressione

- Utilizzo

`function_name(input1,input2)`

- Gli input delle funzioni sono chiamti argomenti

- Una funzione puo' restituire reali, interi, o altri valori dipendenti dalle sue caratteristiche



# Some Really Useful Intrinsic Functions

Function	Argument Type	Result Type	Comment
<code>sqrt(x)</code>	real	real	Square root
<code>abs(x)</code> or <code>abs(i)</code>	real integer	real integer	Absolute value
<code>sin(x)</code>	real	real	Sine
<code>cos(x)</code>	real	real	Cosine
<code>tan(x)</code>	real	real	Tangent
<code>exp(x)</code>	real	real	Exponential
<code>log(x)</code>	real	real	Natural logarithm
<code>log10(x)</code>	real	real	Base-10 logarithm
<code>asin(x)</code>	real	real	Inverse sine
<code>acos(x)</code>	real	real	Inverse cosine
<code>atan(x)</code>	real	real	Inverse tangent
<code>max(a,b)</code>	real/integer	real/integer	Maximum of A & B
<code>min(a,b)</code>	real/integer	real/integer	Minimum of A & B
<code>nint(x)</code>	real	integer	Nearest integer

