Programmazione in Fortran:

Lezione 4

A.A. 2009/2010

Ing. A . Siviglia

nunzio.siviglia@ing.unitn.it

stanza:

Laboratorio didattico di modellistica idrodinamica (2° piano)

Tel 2440

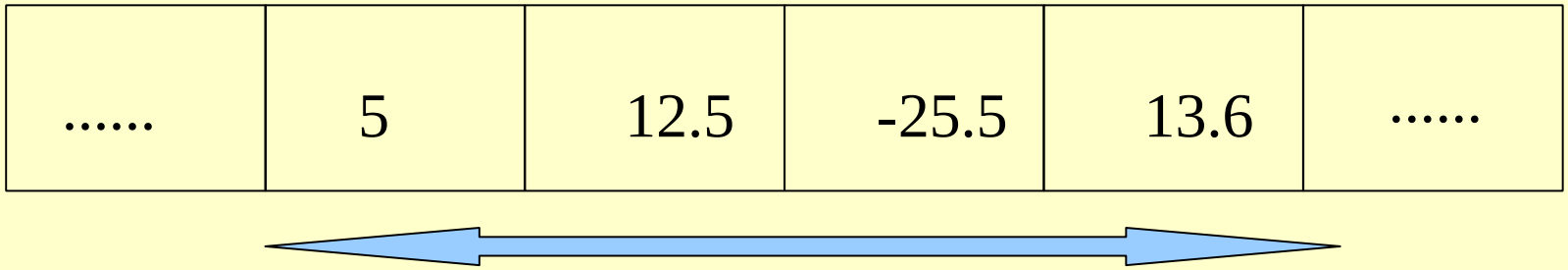
Vettori o array

Alcuni problemi di programmazione necessitano l'utilizzo di un'aggregazione di valori, piuttosto che di uno solo.

Questo significa che è conveniente indicare l'insieme di valori con una sola variabile piuttosto che tante variabili.

In FORTRAN questa problema è risolto utilizzando degliu array.

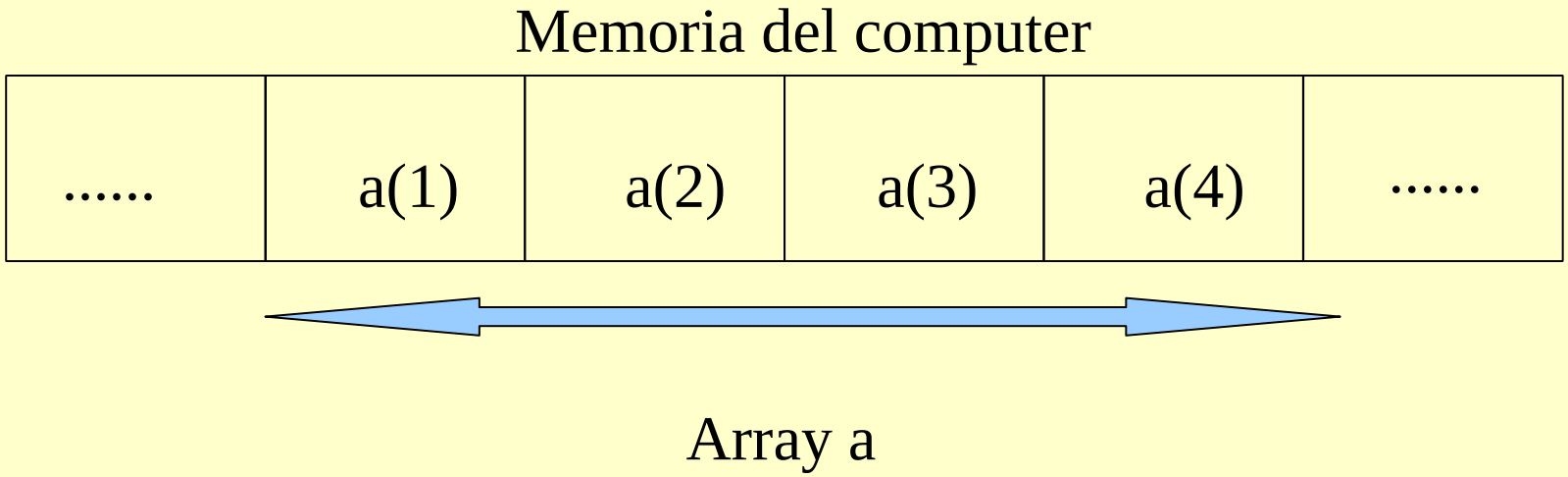
Memoria del computer



Array a

Vettori o array

Un array e' un gruppo di variabili o costanti , tutte dello stesso tipo, che si riferiscono ad un singolo nome. Ogni singolo valore dell'array e' detto elemento dell'array, ed e' identificato dal nome dell'array e da un indice che punta ad una particolare posizione all'interno dell'array. Es:



Vettori e Matrici (Terminologia)

Un array (vettore o matrici a più dimensioni) permette di specificare con un nome ed un insieme di indici una serie di elementi in modo molto semplice. Ogni array ha un tipo ed ogni elemento è di tale tipo. La terminologia relativa agli array è la seguente:

- RANGO (RANK) - Numero di dimensioni
- LIMITI (BOUNDS) - Limite superiore ed inferiore di ogni dimensione
- ESTENSIONE (EXTENT) - Numero di elementi in ogni dimensione
- GRANDEZZA (SIZE) - Numero totale di elementi
- FORMA (SHAPE) - Rango ed estensione
- CONFORMI (CONFORMABLE) - Stessa forma

Dichiarazione

```
INTEGER, DIMENSION(100)    ::  vettore_i
INTEGER                    ::  vettore_i(100)
```

```
REAL, DIMENSION(100)       ::  vettore_r
REAL                       ::  vettore_r(100)
```

```
CHARACTER(len=5), DIMENSION(100)  :: cognome
CHARACTER(len=5)           ::  cognome(100)
```

Visualizzazione degli array

REAL, DIMENSION(15) :: A

Gli elementi degli array sono semplici variabili

```
INTEGER, DIMENSION(10) :: index
REAL, DIMENSION(3)     :: temp
index(1) = 5
temp(3) = REAL(index(1))/5.
WRITE(*,*) “index(1)”,index(1)
```

Inizializzazione degli elementi degli array

Problema:

```
INTEGER, DIMENSION(10) :: j

WRITE(*,*) “j(1)”, j(1)
```

L'array e' stato dichiarato ma nessun valore

e' stato introdotto ancora

Inizializzazione degli elementi degli array

Inizializzazione con delle istruzioni di assegnazione

[A]

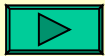
```
REAL, DIMENSION(10) :: array

DO i=1,10

   array(i) = REAL(i)

END DO
```

Inizializzazione degli elementi degli array

Inizializzazione con delle istruzioni di assegnazione

[B]

```
REAL, DIMENSION(10) :: array

array=(/1.,2.,3.,4.,5.,6.,7.,8.,9.,10./)
```



Inizializzazione degli elementi degli array

Inizializzazione con delle istruzioni di assegnazione

[C]

```
REAL, DIMENSION(10) :: array

array=0.
```

Inizializzazione degli elementi degli array

Inizializzazione all'interno delle dichiarazioni

[A]

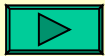
```
INTEGER, DIMENSION(5) :: array2 = (/1,2,3,4,5/)
```

[B]

DO implicito

/(arg1,arg2,....,index = istart,iend,incr)/

```
INTEGER, DIMENSION(5) :: array2 = (/(i,i=1,5)/)
```



Cambiare il limite inferiore del range di un array

```
REAL, DIMENSION(5) :: a1

REAL, DIMENSION(-2:2) :: b1

REAL, DIMENSION(5:9) :: c1
```

a1, b1, c1

sono array di 5 elementi!!!!

Superamento del range di un array (out of bounds)

```
INTEGER :: i

REAL, DIMENSION(5) :: a =(/1.,2.,3.,4.,5./)

REAL, DIMENSION(5) :: b =(/10.,20.,30.,40.,50./)

DO i = 1,6

   write(*,*) a(i)

END DO
```

Superamento del range di un array (out of bounds)

Bounds check on	Bounds check off
a(1) =1.0	a(1) =1.0
a(2) =2.0	a(2) =2.0
a(3) =3.0	a(3) =3.0
a(4) =4.0	a(4) =4.0
a(5) =5.0	a(5) =5.0
a(6	a(6) =10.0
Array subscript exceeds	
allocated area	

Uso delle costanti nella dichiarazione di array

```
INTEGER, PARAMETER :: max_size = 1000
REAL :: array1(max_size)
REAL :: array2(max_size)
REAL :: array3(2*max_size)
```

Operazioni sull'array globale

Condizione necessaria: gli array devono avere la stessa forma

```
INTEGER :: i

REAL, DIMENSION(4) :: a = (/1.,2.,3.,4.)

REAL, DIMENSION(4) :: b = (/5.,6.,7.,8.)

REAL, DIMENSION(4) :: c,d

! somma elemento per elemento

DO i=1,4

    c(i) = a(i) + b(i)

END DO

write(*,*) c
```

Operazioni sull'array globale

Condizione necessaria:

gli array devono avere la stessa forma

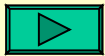
```
INTEGER :: i

REAL, DIMENSION(4) :: a = (/1.,2.,3.,4.)

REAL, DIMENSION(4) :: b = (/5.,6.,7.,8.)

REAL, DIMENSION(4) :: c,d

! somma sull'array globale

d = a + b

write(*,*) d
```

QUANDO SI DEVONO USARE GLI ARRAY?

1) Non risolvere tutti i problemi con gli array!!!!
(vedi esercizio lezione precedente!!!!!)

2) Si usano quando tutti, o la maggior parte dei dati, devono essere tenuti in memoria per risolvere in modo efficiente un problema.

3)Utilizzare array richiede memoria!!!



ESERCIZI

1) Assegnato un numero di dati N, calcolare la media e la deviazione standard

$$\hat{x} = \frac{1}{N}\sum_{i=1}^{N} x_i = \frac{x_1+x_2+x_3+\ldots+x_N}{N}$$

$$\sigma = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \hat{x})^2}$$

2) Trovare il max e il min di una serie di dati e la loro posizione