

Titolo unità didattica: Strutture dati: array

[07]

Titolo modulo : Proprietà di base degli array

[01-T]

Generalità sul concetto di tipo strutturato array

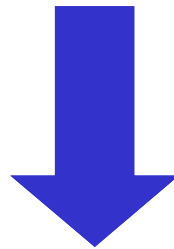
Argomenti trattati:

- ✓ nomi collettivi nei linguaggi naturali
- ✓ l'idea di "struttura di dati"
- ✓ tipi di dati strutturati nei linguaggi di programmazione
- ✓ caratteristiche peculiari degli array: array 1D e 2D, size, shape
- ✓ algoritmi per il calcolo della somma, media e varianza degli elementi di un array

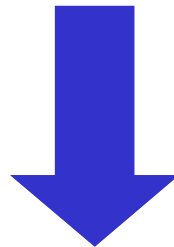
Prerequisiti richiesti: AP-02-*-T, AP-04-*-T, AP-06-02-T

organizzazione e relazioni tra dati

nei linguaggi naturali e nelle applicazioni reali è possibile e utile indicare con un **unico nome un insieme di dati**



nomi collettivi



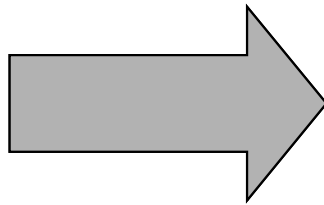
organizzazione tra i dati

organizzazione e relazioni tra dati

organizzazione tra i dati:

i dati aggregati hanno una **struttura**

struttura



esistenza di
relazioni tra i dati

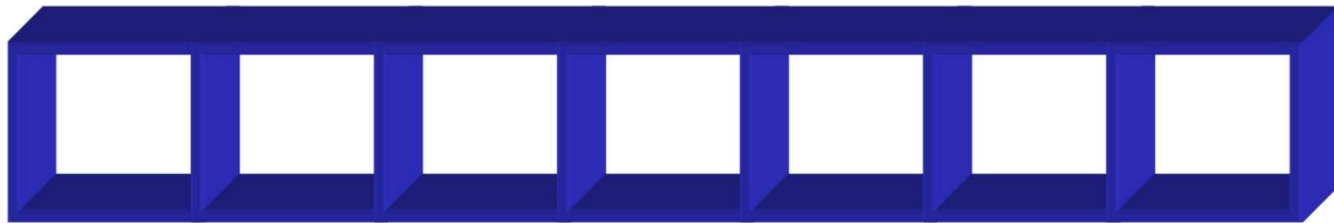
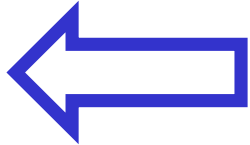
varie forme di **relazioni strutturali** tra i dati

- **fila** di persone
- **albero** genealogico
- **tabella** di nomi

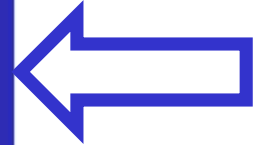
dati organizzati in **fila**



esce dalla
fila

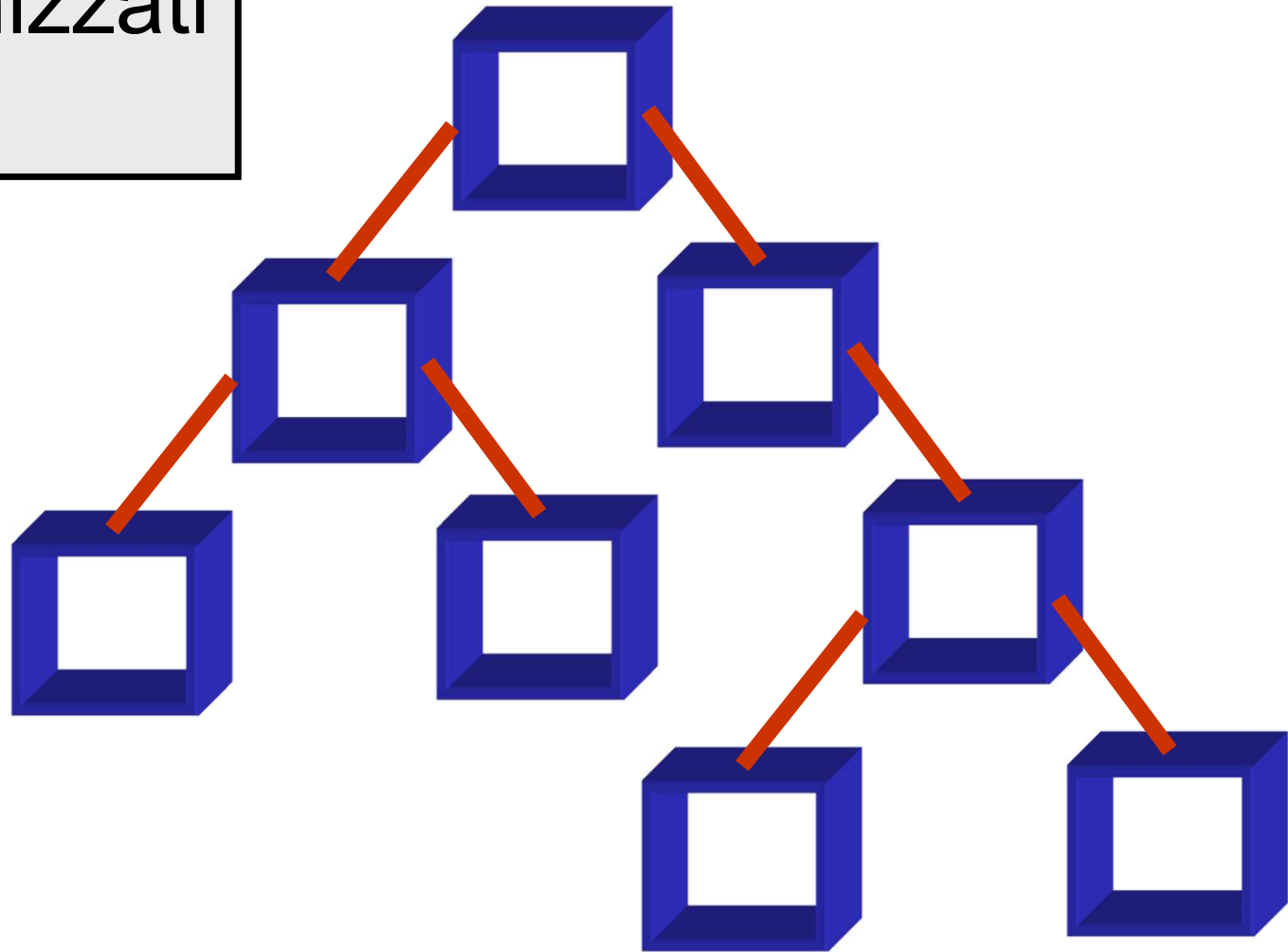


entra nella
fila



la posizione relativa tra i dati dipende dal tempo in cui il dato è entrato nella fila

dati organizzati
ad **albero**



la posizione relativa tra i dati dipende dalla loro
relazione gerarchica

dati organizzati in **tabelle**

esempio di tabella di numeri interi

34	51	87	19	26	14	74
----	----	----	----	----	----	----

una unica **riga** di caselle

--	--	--	--	--	--	--

dati organizzati in **tabelle**

una unica **riga** di caselle

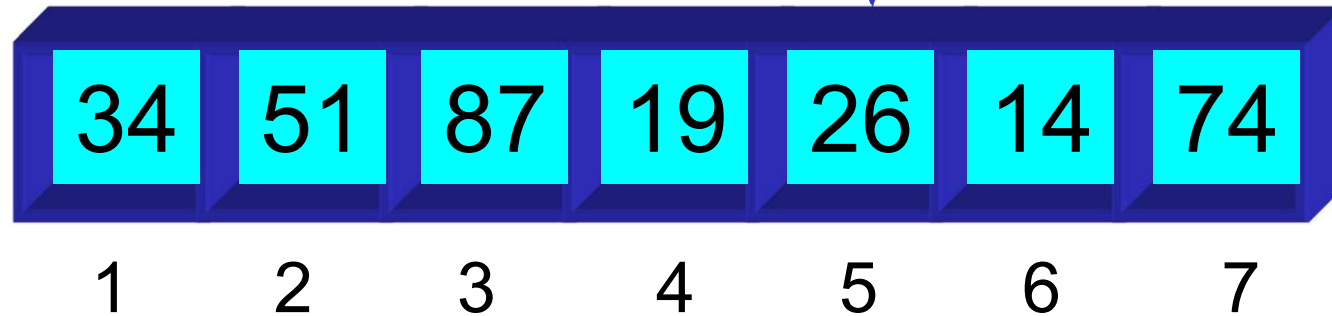
34	51	87	19	26	14	74
----	----	----	----	----	----	----

1 2 3 4 5 6 7

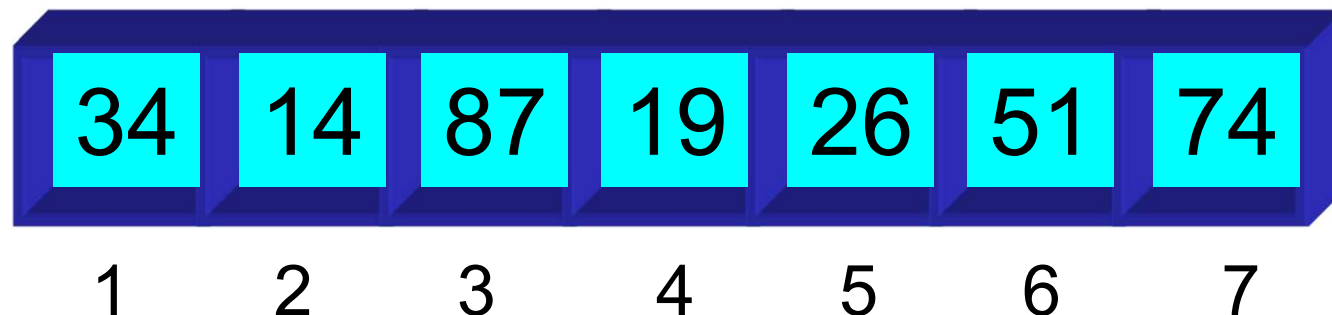
- ✓ ogni casella contiene un dato
- ✓ tutti i dati sono dello stesso tipo
- ✓ ogni dato è individuabile attraverso un numero intero (**indice**)

non si possono scambiare due caselle, ma si può scambiare il loro contenuto

dati organizzati in **tabelle**



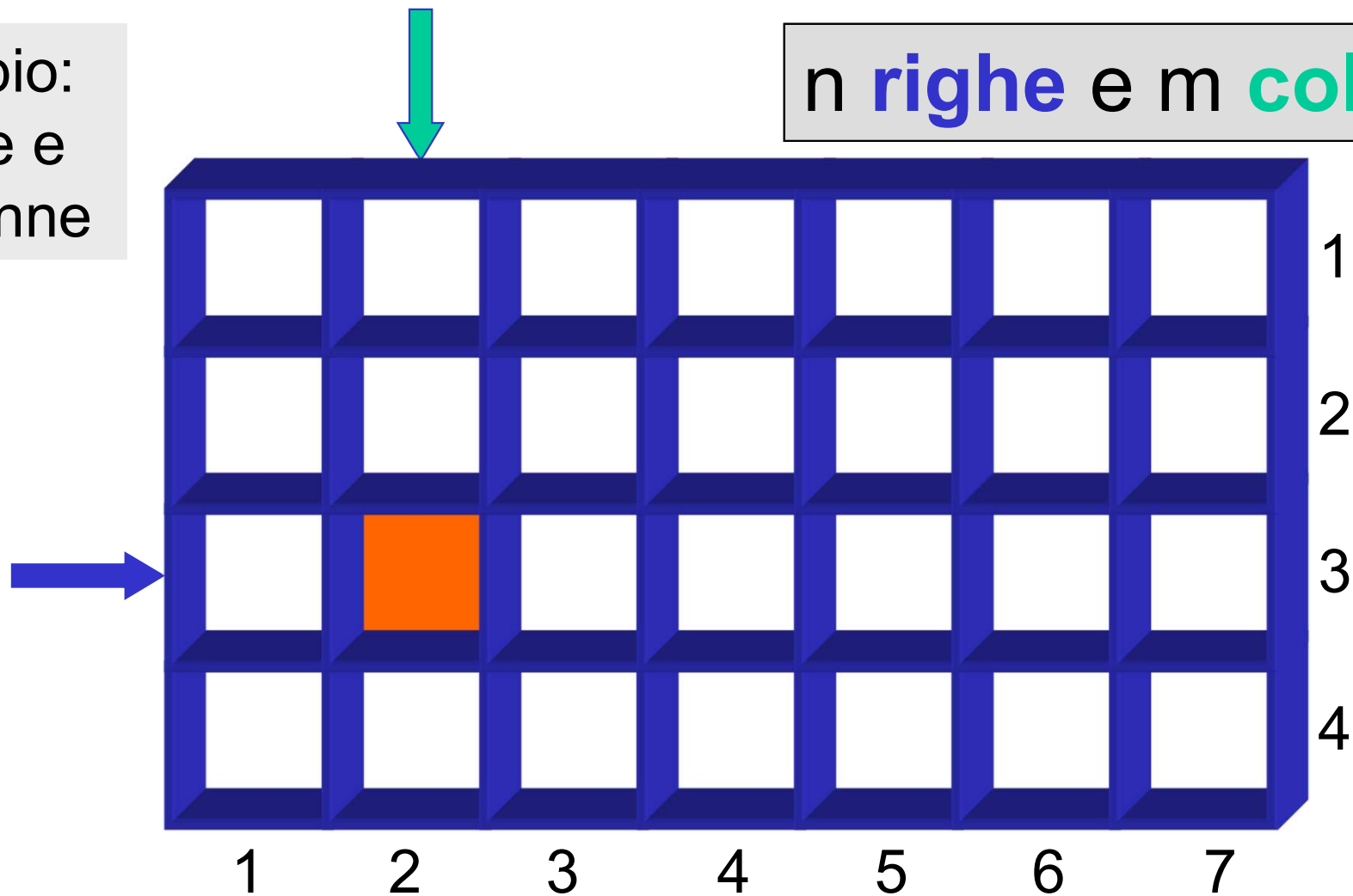
26 è l'elemento di indice 5 della tabella



scambio dell'elemento di indice 2 e
l'elemento di indice 6 della tabella

esempio:
4 righe e
7 colonne

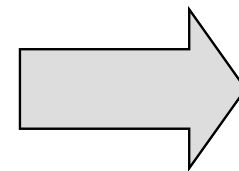
n righe e m colonne



ogni elemento è individuato da **due** indici

l'indice di **riga**

l'indice di **colonna**



elemento
di posto: **3,2**

tipi strutturati primitivi nei linguaggi di programmazione

il tipo strutturato **array**

l'**array** denota un insieme di dati organizzati a **tabella**

- ✓ array a **una** dimensione (array **1D**)
- ✓ array a **due** dimensioni (array **2D**)

- ✓ array a ***n*** dimensioni (array ***nD***)

tipo strutturato array nei linguaggi di programmazione

proprietà del tipo array

- ✓ tutti gli elementi di una variabile di tipo array sono variabili dello **stesso tipo**
- ✓ il numero complessivo di elementi (**size**) di una variabile di tipo array è **fissato** al momento della **dichiarazione** e **non è modificabile**
- ✓ l'insieme dei valori assumibili dagli indici di una variabile di tipo array è **fissato**:
da **0** a **size-1**

la **dichiarazione di variabili di tipo array** deve specificare

- ✓ l'**identificatore** della variabile array
- ✓ il **size** (numero di elementi) dell'array
- ✓ il **tipo** (degli elementi)

(array 1D)

<tipo elementi> <nome array>[<size>]

(array 2D)

**<tipo elementi>
<nome array>[<size1>] [<size2>]**

(size primo indice)

(size secondo indice)

size totale array 2D = **size1 x size2**

Esempio

float inflaz_mese [12] ; indice da 0 a 11

float temperatura_oraria [24]; indice da 0 a 23

char rigo [80]; indice da 0 a 79

int psi [5][7]; indice1 da 0 a 4 indice2 da 0 a 6

char pagina [20][80]; indice1 da 0 a 19 indice2 da 0 a 79

float teta [4][3]; indice1 da 0 a 3 indice2 da 0 a 2

la specificazione di un elemento di un array (**accesso** a un **elemento** di una **variabile** di **tipo array**) deve indicare

- ✓ l'**identificatore** della variabile array
- ✓ l'indice (o gli indici) dell'elemento dell'array

<variabile>[<indice>]

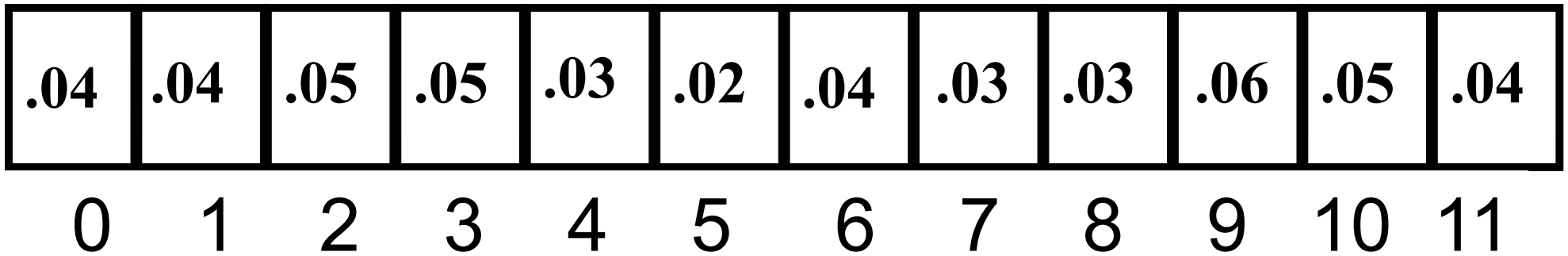
ogni elemento di una variabile di tipo **array 1D** è individuato dal nome della variabile seguito in parentesi quadre da **un** indice

la specificazione di un elemento di un array (**accesso** a un **elemento** di una **variabile** di **tipo array**) deve indicare

- ✓ l'**identificatore** della variabile array
- ✓ l'indice (o gli indici) dell'elemento dell'array

<variabile>[<indice_riga>][<indice_col>]

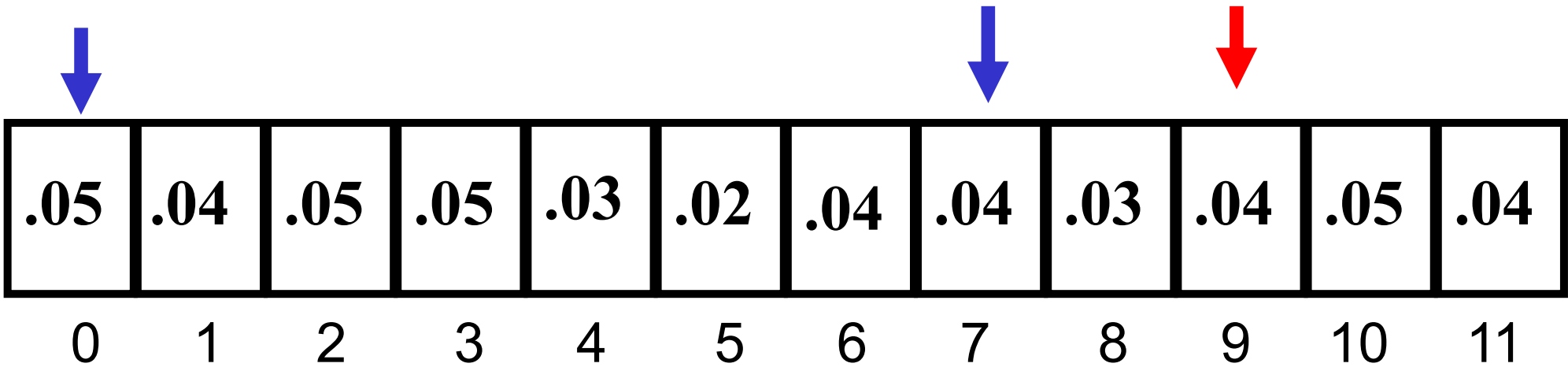
ogni elemento di una variabile di tipo **array 2D** è individuato dal nome della variabile seguito da **due** indici, ciascuno in parentesi quadre



```

inflaz_mese[0] = 0.05 ;
inflaz_mese[7] = inflaz_mese[7]+0.01 ;
i = 9 ;
inflaz_mese[i] = ( inflaz_mese[i+1]+...
                   inflaz_mese[i-1] )/2. ;

```



problema:

calcolo della somma dei valori di un array 1D

dati di input: l'array (variabile **a**), il size dell'array (variabile **n**)

dato di output: la somma (variabile **somma**)

costrutto ripetitivo: **for**

operazione ripetuta (al generico passo **i**):

sommare l' **i**-simo elemento dell'array alla somma degli (**i-1**) precedenti

array in una intestazione di function

```
float somma_array(float a[], int n) {  
    int i;  
    float somma ;  
    somma = 0.0 ;  
    for (i=0; i < n; i++) {  
        somma = somma + a[i] ;  
    }  
    return somma ;  
}
```

size
di
lavoro

n
somme

array come PARAMETRO in una intestazione di function

array come ARGOMENTO in una attivazione di function

```
main {  
  {  
    float miei_dati[100];  
    float somma_miei_dati;  
    read (miei_dati);  
    somma_miei_dati = somma_array(miei_dati, 100);  
    printf(somma_miei_dati);  
  }  
}
```

size
dichiarato

read da modificare in C

size
effettivo

array in una
attivazione di function

array in una
attivazione di function

array come ARGOMENTO in una attivazione di function

```
main {  
  {  
    float miei_dati[100];  
    float somma_miei_dati;  
    read (miei_dati);  
    somma_miei_dati = somma_array(miei_dati, 50);  
    printf(somma_miei_dati);  
  }  
}
```

size
dichiarato

read da modificare in C

size
effettivo

50

size effettivo deve
essere \leq del size dichiarato

array in una
attivazione di function

array come ARGOMENTO in una attivazione di function

read da modificare in C

```
main {  
  {  
  int s;  
  float miei_dati[100];  
  float somma_miei_dati;  
  read (s);  
  read (miei_dati);  
  somma_miei_dati = somma_array(miei_dati, s) ;  
  printf(somma_miei_dati);  
}
```

size
effettivo

size effettivo deve
essere \leq del size dichiarato

problema:

calcolo della media aritmetica dei valori di un array 1D

dati di input: l'array (variabile **a**), il size dell'array (variabile **n**)

dato di output: la media (variabile **media**)

costrutto ripetitivo: **for**

operazione ripetuta (al generico passo **i**):

sommare l'**i**-simo elemento dell'array alla somma degli (**i-1**) precedenti

array in una intestazione di function

```
float media_array(float a[], int n) {  
    int i;  
    float somma ;  
    somma = 0.0 ;  
    for (i=0; i < n; i++) {  
        somma = somma + a[i] ;  
    }  
    return somma/(float) n ;  
}
```

n
somme

problema:
calcolo della **varianza**
dei valori di un array 1D

$$\sigma^2 = \frac{1}{n-1} \sum_{i=1}^n (a_i - \mu)^2$$

media

dati di input: l'array (variabile **a**),
il size dell'array (variabile **n**)

dato di output: la varianza (variabile **varianza**)

struttura dell'algoritmo:

- calcolo della media
- calcolo della sommatoria
- divisione per $n-1$

problema:
calcolo della **varianza**
dei valori di un array 1D

$$\sigma^2 = \frac{1}{n-1} \sum_{i=1}^n (a_i - \mu)^2$$

dati di input: l'array (variabile **a**),
il size dell'array (variabile **n**)

dato di output: la varianza (variabile **varianza**)

costrutto ripetitivo: **for**

operazione ripetuta (al generico passo **i**):

sommare il quadrato della differenza tra
l'**i**-simo elemento dell'array e la media dei
valori dell'array alla somma dei quadrati delle
omologhe (**i-1**) differenze precedenti



media

```
float varianza_array(float a[], int n) {  
    int i;  
    float media, varianza ;  
    media = media_array(a, n) ;  
    varianza = 0.0 ;  
    for (i = 0; i < n; i++) {  
        varianza = varianza + (a[i]-media)^2 ;  
    }  
    varianza = varianza/(float)(n-1) ;  
    return varianza ;  
}
```

2n somme
n+1 prodotti
+ costo di
media_array

```

float varianza_array(float a[],int n) {
int i;
float media, varianza;
varianza = -1.0 ;
if (n > 1) {
    varianza = 0.0 ;
    media = media_array(a,n) ;
    for (i=0; i < n; i++) {
        varianza = varianza+(a[i]-media)^2 ;
    }
    varianza = varianza/(float)(n-1) ;
}
return varianza;
}

```

$3*n$
 somme
 $n+2$
 prodotti