Alcuni problemi possono essere risolti con un numero limitato di variabili *elementari*, ognuna definita da un *nome* e da un *tipo* (int, float, char, …) che la caratterizza, ognuna utilizzata in modo diverso dalle altre, referenziandone semplicemente il nome.

Esempio: dovendo comprare un certo numero di quaderni, per calcolare l'importo da pagare:
- definiremo 3 variabili: int num; float prezzo, importo;
- leggeremo numero e prezzo: cin >> num >> prezzo;
- calcoleremo e restituiremo il risultato: importo = num*prezzo; cout<<"spesa = "<< importo;

Altri problemi, invece, hanno necessità di conservare ed elaborare **molti dati, tutti dello stesso tipo**.

Esempio: calcolare la temperatura media in una settimana ed indicare il numero di giorni in cui la temperatura è stata inferiore alla media settimanale.

Per questo tipo di problemi si utilizza una *struttura di dati omogenea*, l'**array monodimensionale** (detto anche **vettore**): il vettore è un insieme di dati, **tutti dello stesso *tipo***, definiti da un **nome collettivo** e da **una dimensione** (numero massimo di elementi che la struttura può contenere).

La *dichiarazione* di un vettore in C/C++ è la seguente: ***tipo*** **nome [dimensione];**
un elemento è individuato da **nome** e **posizione** indicata tra parentesi quadrate: **nome [posizione]**

La posizione, detta **indice dell'array**, deve essere un **valore intero** compreso tra **0** e **dimensione - 1** :
il **primo** elemento è in posizione **0**, l'**ultimo** elemento è in posizione **dimensione - 1**

Nell'esempio precedente *dichiareremo* un **array di 7 elementi** di tipo intero: ***int*** **temp [7];**
la temperatura del 1^ giorno sarà memorizzata nella **prima posizione dell'array,** posiz. **0** temp **[0]**
l'ultima temperatura sarà posta nella posizione **6** , al 7^ posto compresa la posizione 0 temp **[6]**

Se le temperature registrate nella settimana sono:
12° per il 1^ giorno, poi 10°, 6°, 0°, -2°, 4° e 13° per il 7^ e ultimo giorno,
la temperatura media è 6,14° e i giorni con temperatura inferiore a 6,14° sono 4:
il 3^ giorno con 6°, il 4^ con 0°, il 5^ con -2°, il 6^ giorno con 4°,
le temperature sono salvate rispettivamente nelle posizioni 2, 3, 4 e 5.

posizione nel vettore (=valore dell'indice):	0	1	2	3	4	5	6
valore degli elementi del vettore **temp**:	12	10	6	0	-2	4	13

media temperature: 6,14
giorni con temperatura inferiore: 4

L' **indice** che individua un particolare elemento di un array può essere:

- un valore costante intero; esempio 0 per il primo elemento: `temp[0] = 12;`
- un valore contenuto in una variabile intera; esempio k: `int k=1;` `temp[k] = 10;`
- una formula che restituisce un valore intero; se k vale 1, k+3 vale 4: `temp[k+3] = -2;`

Il programma che risolve il problema delle temperature è il seguente:

PSEUDOCODIFICA

```
INIZIO
  s ← 0
  k ← 0
  ESEGUI MENTRE k < 7
         leggi (temp[k])
         s ← s + temp[k]
         k ← k + 1
  RIPETI
  m ← s / 7
  num ← 0
  k ← 0
  ESEGUI MENTRE k < 7
         SE temp[k] > m  ALLORA
              num ← num + 1
         FINESE
         k ← k + 1
  RIPETI
  scrivi ("temp. inferiori media : ", num)
FINE
```

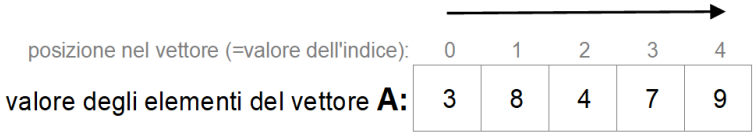
CODIFICA in C/C++

```
#include <iostream>
using namespace std;

int main() {
 int temp [7], k, num ;
 float  s, m;
  s = 0;
  for(k = 0; k < 7; k = k+1) {
         cin  >> temp[k];
         s = s + temp[k]; //-- somma temperature
  }
  m = (float) s / 7;       //-- calcolo della media
  num = 0;
  for(k = 0; k < 7; k++) {
         if (temp[k] < m)
              num = num + 1;    //-- calcolo giorni con
  }                    //-- temperature inferiori ad m (media)
  cout << "temp. inferiori media : " << num;
}
```

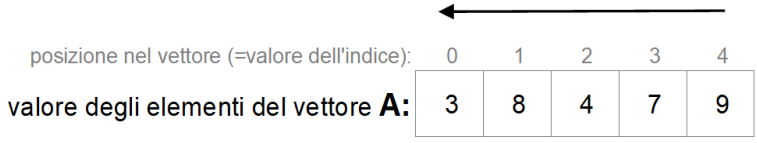
I vettori (array monodimensionali) possono essere elaborati in molti modi mediante l'istruzione ***for***;
vediamo qualche esempio:

- in ordine **crescente** di indice:




```
#define DIM 5
. . .
  int A[DIM];
  for (int k = 0;    k < DIM;   k++){
    cout<< " elem. posto " << k <<"? ";
    cin >> A[k];
  }
```

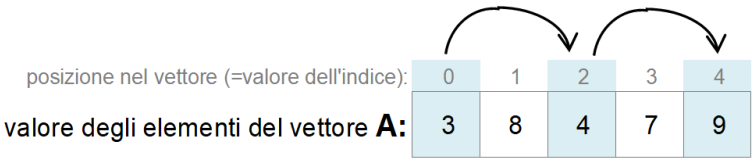
- in ordine **decrescente** di indice:



letti i valori dell'array **A**,
li esponiamo a video in ordine inverso:

```
  for (int h = DIM – 1;  h >= 0;   h--)
     cout << A[h] << endl;
```

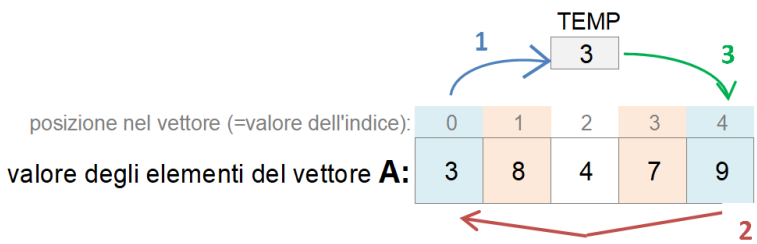
- a salti in ordine crescente (o decrescente):



scriviamo solo gli elementi si posto pari:

```
  for (int i = 0;    i < DIM;    i += 2)
     cout << A[ i ] << endl;
```

- dagli estremi fino al centro:



poi li memorizziamo in ordine inverso nel vettore A, scambiandoli di posto a 2 a 2, dagli estremi fino al centro (=DIM/2 = 5/2 = **2** in questo caso):

```
  for (int k = 0;  k < DIM/2;  k ++) {
     int TEMP = A [ k ] ;          //-- 1
     A[i] = A [ DIM – 1 – k ];     //-- 2
     A [ DIM – 1 – k ] = TEMP;     //-- 3
}
```