

VETTORI IN C++

vettore-> Chiamato anche array, che è un sinonimo, è un collezione di dati di tipo omogeneo, ovvero devono essere dello stesso tipo.

- **Es:** un vettore di float (numeri decimali) per contenere i voti di informatica di uno studente.

Si utilizzano quando si ha la necessità di memorizzare diversi dati di uno stesso tipo e consentono di agevolare il programmatore in quanto si avrà un codice più pulito (organizzato) e in quanto con le sole variabili normali non è possibile risolvere problemi che richiedono un numero di input, da tastiera, che potrebbe variare di esecuzione in esecuzione.

- **Es:** Si vuole salvare il numero di giorni per ogni mese dell' anno. Non è necessario dichiarare 12 variabili intere, ma basterà dichiarare un array (vettore) di interi di 12 celle:
Si potrà dichiarare un array di nome giorniMesi[12] per risolvere questo problema.

DICHIARAZIONE DI UN ARRAY

Per dichiarare un array analogamente alle dichiarazioni delle variabili normali basterà specificare:

tipo nomeArray[dimensione]

Il **tipo** si riferisce al tipo dei dati che si vorranno salvare come può essere int, string, char, bool, float, ...

nomeArray si riferisce al nome dell' array e deve essere sensato in base a che cosa verrà salvato in esso.

La **dimensione** deve essere un intero positivo e deve essere specificata in fase di compilazione*.

IMPORTANTE Se la dimensione non dovesse venire specificata

in compilazione, ma in fase di esecuzione sulla base di un input da tastiera, potrebbe essere che accedendo e scrivendo nelle celle delle array si vadano a modificare dati in RAM che sono usati da altre variabili o addirittura altri programmi.

COME RISOLVERE IL PROBLEMA se vi viene richiesto di usare un array di dimensione n, dove n viene inserito da tastiera in esecuzione?

RISPOSTA In questo caso basterà andare a dichiarare un array abbastanza grande da contenere i vari input, per esempio un migliaio di celle, e si andranno ad utilizzare solo le prime n celle di esso. (Vedi avanti un esempio pratico)

ACCESSO E MODIFICA DATI ARRAY

Ogni elemento in un array occupa una **posizione** numerata, chiamata **indice**.

L'indice di un array inizia sempre da **0** e termina a **dimensione dell'array - 1**.

Ad esempio, in un array con 5 elementi:

- L'indice del primo elemento è 0.
- L'indice dell'ultimo elemento è 4.

Per accedere a un elemento, usa il nome dell'array seguito da parentesi quadre [] e l'indice dell'elemento desiderato:

```
array[i];
```

Con array[i] è possibile svolgere tutte le operazioni che è possibile svolgere con una qualsiasi variabile quindi per esempio:

- stamparne il valore svolgendo `cout<<array[i];`
- modificarne il valore svolgendo `array[i]=variabile`

NOTA: la variabile usata come indice all'interno delle graffe può avere qualsiasi nome, non per forza i, ma solitamente per chiarezza si usa i, l'idea è in casi specifici di rinominare una variabile con un nome adatto a far capire cosa andrà a contenere

DICHIARAZIONE E INIZIALIZZAZIONE

È possibile dichiarare e inizializzare i valori di un array in questo modo:

```
int array[5] = {10, 20, 30, 40, 50}; // Array di 5 interi
```

ESEMPIO DI CODICE 1:

```
int main() {  
    int array[5] = {10, 20, 30, 40, 50};  
  
    // Modifica di alcuni elementi  
    array[0] = 15; // Cambia il primo elemento  
    array[3] = 45; // Cambia il quarto elemento  
  
    // Stampa degli elementi modificati  
    cout << "Primo elemento modificato: " << array[0] << endl; // Output: 15  
    cout << "Quarto elemento modificato: " << array[3] << endl; // Output: 45  
  
    return 0;  
}
```

compilazione*-> è la fase di traduzione del codice in c++ in codice macchina, eseguibile dal processore del computer utilizzato, con il fine di generare un programma eseguibile. Su dev c++ avviene quando si preme il bottone con i quattro rettangoli colorati. Avviene anche quando si clicca il terzo bottone, in quanto prima svolge la compilazione e poi esegue il programma generato.

x x

