

I VETTORI (O ARRAY)

Il linguaggio C

1



DEFINIZIONE DI VETTORE

- Un vettore può essere definito come una “collezione organizzata di oggetti”:
 - **“collezione”**: implica che gli oggetti contenuti nel vettore siano dello stesso tipo;
 - **“organizzata”**: implica che sia possibile identificare univocamente tutti gli oggetti del vettore.



I VETTORI IN C

- Un vettore in C è:
 - una **collezione** di variabili dello **stesso tipo**;
 - in cui **ogni elemento è univocamente identificabile tramite l'uso di indici numerici** che, in un vettore di dimensione N, vanno da 0 ad N-1.

<i>Indici</i> →	0	1	2	3	4	5	6	7
<i>Valori</i> →	5	-2	21	-10	-2	35	40	120



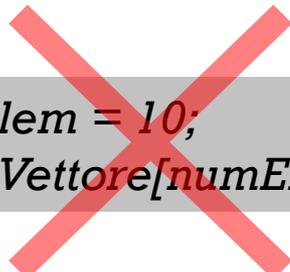
COME SI DICHIARA UN VETTORE?

- La dichiarazione di un vettore si compone di 3 elementi:
 1. il tipo di dati (int, float, char, ...);
 2. il nome del vettore (come per le variabili);
 3. il numero di elementi (dimensione) contenuti nel vettore, scritto tra le parentesi quadre.
- La dimensione del vettore deve essere un numero intero o una costante definita con l'istruzione “*#define*”, non può essere una variabile.

```
int nomeVettore[10];
```

```
#define NUM_ELEM 10  
...  
int nomeVettore[NUM_ELEM];
```

```
int numElem = 10;  
int nomeVettore[numElem];
```



USO PRATICO DEI VETTORI

- Ogni elemento di un vettore può essere utilizzato come una variabile.
- Esempi:
 - assegna al terzo elemento del vettore “*vett*” il valore 72:

```
vett[2] = 72;
```

- copia nella variabile “*val*” il quinto elemento del vettore “*vett*”:

```
val = vett[4];
```



INIZIALIZZAZIONE DI UN VETTORE

- Come per le variabili, gli elementi di un vettore, all'atto della dichiarazione, **non vengono inizializzati**, a meno che non lo si faccia in modo esplicito.
- L'inizializzazione di un vettore può avvenire in due modi:
 - contemporaneamente alla dichiarazione dello stesso;
 - in un secondo momento, assegnando esplicitamente ad ogni elemento del vettore il valore voluto.



INIZIALIZZAZIONE DI UN VETTORE

- Per dichiarare e contemporaneamente inizializzare un vettore si deve procedere come negli esempi seguenti:

```
int vett[8] = {10, 8, -1, 3, -5, 68, 24, -2};
```

```
int vett[] = {10, 8, -1, 3, -5, 68, 24, -2};
```

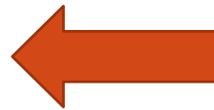


INIZIALIZZAZIONE DI UN VETTORE

- Assegnando esplicitamente ad ogni elemento del vettore il valore voluto. In questo caso si dichiara il vettore e poi si usa un ciclo pre-condizionale per inizializzarlo.

```
int vett[15], i = 0;

while(i < 15)
{
    vett[i] = 3;
    i++;
}
```



Inizializza tutti gli elementi del vettore di interi “vett” al valore 3



ESEMPIO

- Leggi 10 numeri interi e visualizzali nell'ordine inverso a quello di inserimento.

```
#include <stdio.h>
#include <stdlib.h>

int main()
{
    int vett[10], i;

    for(i = 0; i < 10; i++)
    {
        printf("Inserisci il valore: ");
        scanf("%d", &vett[i]);
    }
}
```

```
printf("\nValori in ordine inverso:\n");
for(i = 9; i >= 0; i--)
{
    printf("%d ", vett[i]);
}

system("PAUSE");
return 0;
}
```



ESERCIZI

- **Esercizio 1:** scrivere un programma che legga 10 numeri interi da tastiera, li memorizzi in un vettore e, dopo averne calcolato la media, visualizzi solo i numeri contenuti nel vettore che sono maggiori o uguali alla media calcolata.
- **Esercizio 2:** scrivere un programma che legga 20 numeri interi da tastiera e li visualizzi in ordine crescente.
- **Esercizio 2.1:** scrivere un programma che legga al massimo 100 numeri interi positivi da tastiera e li visualizzi in ordine crescente o decrescente in funzione della scelta dell'utente.
- **Esercizio 3:** scrivere un programma che riceve in ingresso una sequenza di N (inserito dall'utente e minore di 50) numeri interi. Terminato l'inserimento della sequenza di numeri, l'utente inserisce un ulteriore valore. Il programma deve indicare se tale valore è contenuto o meno nel vettore.

ESERCIZI

- **Esercizio 4:** Scrivere un programma che legge N (inserito dall'utente e minore di 50) numeri interi e li memorizza in un vettore. Il programma deve generare e visualizzare un secondo vettore che compatta i numeri contenuti nel primo vettore utilizzando i seguenti criteri:
 - ogni numero che compare ripetuto nel primo vettore, deve comparire una sola volta nel secondo vettore;
 - ogni numero uguale a zero presente nel primo vettore non deve comparire nel secondo vettore.
- Ad esempio, si supponga $N=8$ e si consideri la sequenza di numeri -1 18 3 0 24 3 6 0 inseriti da tastiera. Il programma deve visualizzare -1 18 3 24 6 .