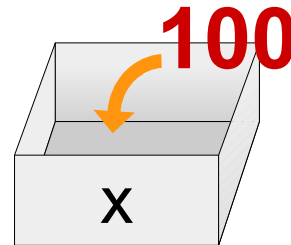


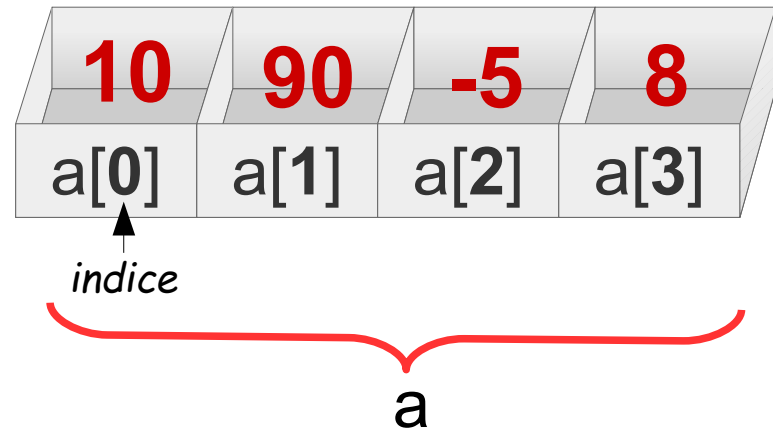
Cos'è un array

una variabile di tipo *semplice*
(int, float, double, char, ..)
può contenere un solo valore.



```
int x;  
x = 100;
```

Una variabile di tipo "array" è una
variabile in grado di contenere più
valori, tutti dello stesso tipo.



int a[4]; ← dichiara un "array di 4 elementi di tipo int"

a[0] = 10; ← assegna il valore 10 all'elemento di indice 0

a[1] = 90;

a[2] = -5;

a[3] = 8;

*Nota: 4 è la **dimensione** (o **lunghezza**) dell'array. E' il numero massimo di valori memorizzabili ed è fissato in fase di dichiarazione (non può essere modificato in seguito).*

Note importanti

consideriamo la dichiarazione `int arr[4];`

NOTA BENE:

- Il primo elemento dell'array è `arr[0]`, perchè in C gli indici degli array partono da zero (non da uno!)
- L'ultimo elemento dell'array è `arr[3]` (non `arr[4]` !!!). Nella dichiarazione si specifica la *dimensione* dell'array, quindi l'indice dell'ultimo elemento è pari a *dimensione - 1*.
- L'accesso a un indice non valido (esempio `arr[4]` , `arr[5]`, ecc..) non genera un errore di compilazione, ma porta a errori molto gravi nel programma.

Esempio: `arr[5]=1000;` scrive il valore 1000 in un'area di memoria non destinata all'array (sovrascrivendo potenzialmente il valore di altre variabili del programma o generando la terminazione immediata del programma da parte del S.O)

Esempio: utilizzo di un array in C

Esempio 1

```
#include <stdio.h>

int main() {
    int arr[4];
    arr[0] = 10; arr[1] = 90;
    arr[2] = -5; arr[3] = 8;

    scanf("%d", &arr[2]);
    printf("%d", arr[2]);

    for( int i=0; i<4; i++ ) {
        printf("%d \n", arr[i]);
    }
}
```

Esempio 2 (migliore!)

```
#define MAX 4
#include <stdio.h>

int main() {
    int arr[MAX];
    arr[0] = 10; arr[1] = 90;
    arr[2] = -5; arr[3] = 8;

    scanf("%d", &arr[2]);
    printf("%d", arr[2]);

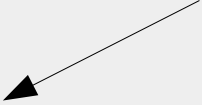
    for( int i=0; i<MAX; i++ ) {
        printf("%d \n", arr[i]);
    }
}
```

Inizializzazione di un array


```
#define MAX 5  
#include <stdio.h>
```

```
int main() {  
    int arrayUno[MAX];  
    double arrayDue[MAX] = { 1.5, 2.5 , 3.5 , 4.5 , 5.5 } ;  
    char arrayTre[MAX] = { 'a', 'b', 'c' } ;  
}
```

array non inizializzato.
Gli elementi contengono valori indefiniti (casuali).



array parzialmente inizializzato
(solo i primi tre valori sono inizializzati. Gli altri elementi contengono valori indefiniti)



Errori da evitare

```
#define MAX 1000
```

```
int main() {
```

```
    int vet1[MAX] = { 1,2,3 4};
```

```
    int vet2[MAX] = {5,6,7 8};
```

```
    vet2 = vet1; Non è possibile assegnare un array ad un altro array
```

```
    vet2[MAX] = vet1[0]; L'ultimo indice valido per vet2 è MAX-1, non MAX !
```

```
    scanf("%d", &vet1);
```

```
    printf("%d", vet2);
```

Non è possibile leggere un intero array con una singola scanf o stamparlo con una singola printf

```
}
```

Esercizio: "inversione" di un array

Scrivi un programma che richiede all'utente di inserire N numeri interi (con N stabilito dall'utente e compreso tra 1 e 1000) , e visualizza i numeri inseriti in ordine inverso rispetto all'ordine di inserimento.

<i>Esempio</i>	<i>INPUT:</i>	<i>N=4</i>
		<i>5, 8, 3, 2</i>
	<i>OUTPUT:</i>	<i>2, 3, 8, 5</i>

Soluzione

```
#define MAX 1000
int main() {
    int vettore[MAX],i,N;
    do {
        printf("\n Quanti numeri vuoi inserire? : ");
        scanf("%d", &N );
    } while( N <1 || N > MAX );
    for( i=0; i<N; i++ ) {
        printf("\n vettore[%d] = ",i);
        scanf("%d", &vettore[i] );
    }
    printf("\n Elementi in ordine inverso: ");
    for( i = N-1 ; i >= 0 ; i-- ) {
        printf( "%d ", vettore[i] );
    }
}
```

Esercizio: eliminazione dei duplicati

Scrivi un programma che richiede all'utente di inserire N numeri interi (con N stabilito dall'utente e compreso tra 1 e 1000) e li memorizza in un array privo di duplicati (cioè, se l'utente fornisce un numero già inserito in precedenza, questo non deve essere inserito nell'array). Terminato l'inserimento, il programma mostra il contenuto dell'array.

<i>Esempio</i>	<i>INPUT:</i>	<i>N=10</i>
		<i>1,8,1,1,3,5,3,3,3,1</i>
	<i>OUTPUT:</i>	<i>1,8,3,5</i>

Soluzione

```
#define MAX 1000
int main() {
    int N, i, j, k = 0, elem, arr[MAX], presente;
    ... leggo N ...
    for( i=0; i < N; i++ ) {
        scanf("%d", &elem );
        presente = 0;
        for( j=0; ( j < k ) && ( !presente ); j++ ) {
            if (arr[j] == elem )
                presente = 1;
        }
        if ( presente == 0 ) {
            arr[k] = elem;
            k++;
        }
    }
    for( i=0; i < k ; i++ ) { printf(" %d ", arr[i] ); }
}
```

- **N** = numero di elementi che l'utente vuole inserire
- **k** = numero di elementi attualmente memorizzati nell'array
- **presente** = flag che indica se l'elemento appena inserito è già presente (0) oppure no (1) nell'array