

# ARRAY



CORSO DI GAME PROGRAMMING  
1° ANNO

Docente **Davide Caio**





# PROBLEMA

Voglio richiedere 10 numeri in input all'utente per poterli visualizzare in ordine inverso.

Con quello imparato fino ad ora, l'unico modo è creare 10 variabili dove memorizzare gli input dell'utente e poi stamparle in ordine inverso. Se i numeri fossero 100? Molto scomodo.

Se il numero dei numeri fosse deciso dall'utente prima di iniziare a inserirli? Impossibile da scrivere non sapendo a compile time quanti numeri saranno scelti dall'utente.



# ARRAY

È possibile archiviare più variabili dello stesso tipo in una struttura di dati chiamata **array**. Un array deve essere dichiarato specificando il tipo.

**SINTASSI:** per specificare un array durante la dichiarazione delle variabili basta utilizzare le [] dopo il tipo dell'array.

**ESEMPIO:**

```
int[] arrayInteri;
```

Un array DEVE avere una **dimensione fissata**. Non è possibile modificare la lunghezza di un array se non creando un nuovo array da 0 con il nuovo numero specificato di elementi.



# ARRAY INIZIALIZZAZIONE

Abbiamo detto che un array deve essere inizializzato con una specifica grandezza.

## SINTASSI

```
float[] arrayFloat = new float[10];
```

Così facendo ho dichiarato e inizializzato un array di float.



# ARRAY ACCESSO AD UN ELEMENTO

Si può accedere a un elemento di un array specificandone l'indice (cioè la sua posizione).

ATTENZIONE: gli indici partono da 0, quindi l'elemento con indice 0 è il primo elemento del mio array.

```
int[] arrayInteri = new int[10];  
arrayInteri[2] = 32; //Con questa riga assegno il valore 32 all'elemento in  
posizione 2 del mio array (cioè il terzo del mio insieme).
```

DOMANDA: Che valore hanno gli elementi dell'array subito dopo la sua inizializzazione?



# ARRAY ESEMPIO

```
int[] primiDieciNumeriPositivi = new int[10];  
for (int i = 0; i < 10; i++) {  
    primiDieciNumeriPositivi[i] = i;  
}  
  
for (int i = 9; i >= 0; i--) {  
    Console.WriteLine (primiDieciNumeriPositivi[i]);  
}
```

Cosa stampa il codice qui sopra?

Cosa c'è di molto scomodo nel codice qui sopra?



# ARRAY - PROPRIETA'

La proprietà **Length** di un array ne ritorna la sua dimensione.

```
int[] primiDieciNumeriPositivi = new int[10];  
for (int i = 0; i < primiDieciNumeriPositivi.Length; i++) {  
    primiDieciNumeriPositivi[i] = i;  
}  
  
for (int i = primiDieciNumeriPositivi.Length - 1; i >= 0; i--) {  
    Console.WriteLine (primiDieciNumeriPositivi[i]);  
}
```



# ARRAY - TIPO

Abbiamo detto che un array può contenere un insieme di elementi definito di un certo tipo.

Se è un array di interi conterrà elementi di tipo di valore.

Ma l'array in sé è un tipo di valore o un tipo di riferimento?

Proviamo a dedurlo dall'output a console del seguente esempio.





# ARRAY - TIPO

```
int[] a = new int[3];  
a[0] = 120;  
a[1] = 50;  
a[2] = 3;
```

```
int[] b = a;  
b[1] = 97;  
Console.WriteLine (a[1]);
```

Il codice sopra stampa a console 97. Con questa informazione, sapete dirmi quindi se un array è di tipo di valore oppure di tipo di riferimento?



# ESERCIZIO 1

Richiedere all'utente 10 numeri interi e visualizzarli in ordine inverso.



## ESERCIZIO 2

Chiedere all'utente 10 numeri interi. Contare quante volte viene inserito il numero 0. Stampare a video la stessa sequenza di numeri interi inserita dall'utente tranne che per gli elementi di valore 0 che saranno sostituiti dal numero di volte che l'elemento 0 compare nella lista stessa.



## ESERCIZIO 3

Memorizzare in un array i primi 20 numeri primi.



## ESERCIZIO 4

Leggere un array di interi e verificare se è palindromo (ovvero se non cambia ad essere letto dalla prima cella all'ultima o viceversa).



## ESERCIZIO 5

Leggere un array di interi di 6 posizioni, leggere un ulteriore numero intero e dire quanti numeri memorizzati nell'array sono inferiori e quanti superiori dell'ultimo numero letto.



## ESERCIZIO 6

Leggere un array di interi di 10 posizioni, e dire se i valori memorizzati nell'array sono memorizzati in modo crescente.



## ESERCIZIO 7

Leggere un array di interi di 10 posizioni e stampare il numero che compare più volte all'interno dell'array, qualora ci siano più numeri che compaiono lo stesso numero di volte stampare quello che compare per primo.