

Array Monodimensionali

[Descrizione](#)

[Dichiarazione](#)

[Operazioni](#)

[Accesso elemento](#)

[Lettura elemento](#)

[Scrittura elemento](#)

[Algoritmi principali](#)

[Stampa Array](#)

[Leggi Array](#)

[Ordinamento Array](#)

[Ricerca Massimo/Minimo](#)

[Esercizi](#)

[ESERCIZIO 1](#)

[ESERCIZIO 2](#)

Descrizione

Un **array monodimensionale** (o vettore) è una sequenza di variabili dello stesso tipo che vengono situate consecutivamente nella memoria ed alle quali è possibile accedere usando uno stesso nome (identificatore) a cui viene aggiunto un indice. Pensiamo all'**array** come ad un insieme di cassette numerati.

Dichiarazione

```
#define N 10
const int M = 100

//dichiara ad esempio un array di 10 elementi
int A[N];

//dichiara un array di 10 elementi float
float V[10];

//dichiara un array di 100 elementi
int B[M];

//array di interi con la dimensione definita implicitamente
int C = {3,5,6,2,1}

//array di interi parzialmente inizializzato
int D[10] = {3,5,6,2,1}
```

```
//array di interi totalmente inizializzato
int E[10] = {3,5,6,2,1,8,4,7,9,0}

//si può anche indicare {} inizializza gli elementi dell'array a zero
int F[N] = {0};
```

Operazioni

Accesso elemento

```
int vett[5] = {2, 0, 1, 6, 3};
vett[4]; // -> mi riferisco al valore 6
```

Lettura elemento

```
int vett[5];
cin >> vett[2]; // inserisco un valore nella posizione 3
```

Scrittura elemento

```
int vett[5] = {2, 0, 1, 6, 3};
cout << vett[2]; //Output: 1
```

Algoritmi principali

Stampa Array

```
int vett[5] = {2, 0, 1, 6, 3};

for (int i=0; i<5; i++) {
    cout << vett[i] << " ";
}

// Output: 2 0 1 6 3
```

Leggi Array

```
int vett[5];

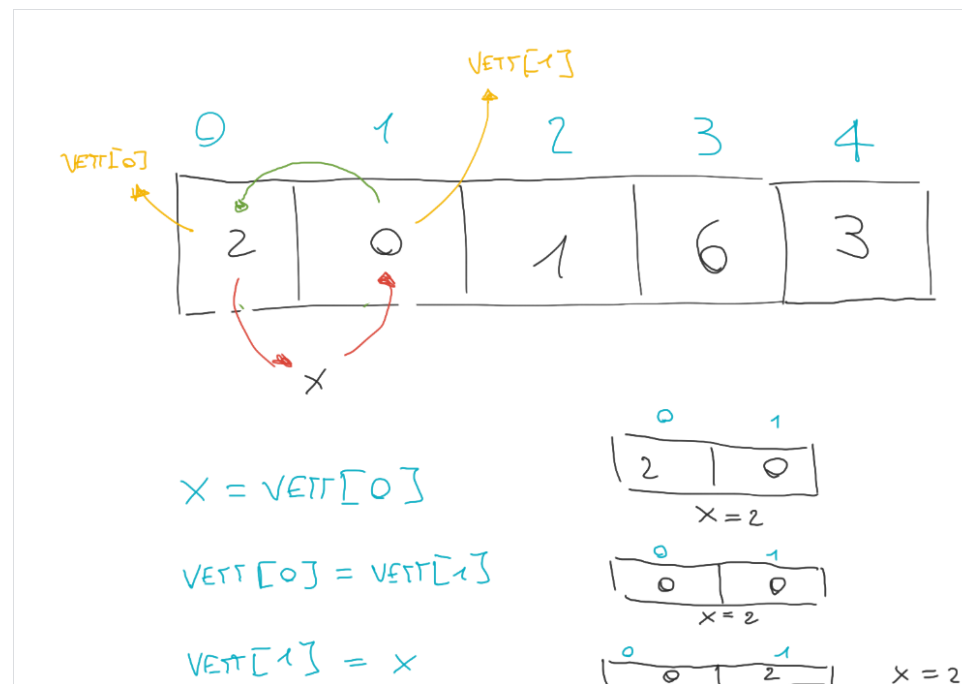
for (int i=0; i<5; i++) {
    cin >> vett[i];
}
```

Ordinamento Array

```
int vett[5] = {2, 0, 1, 6, 3};

for (int i = 0; i < 5; i++) {
    for (int a = 0; a < 5; a++) {
        if (vett[a] > vett[i]) {
            int temp = vett[i];
            vett[i] = vett[a];
            vett[a] = temp;
        }
    }
}
```

Ragionamento per "swappare" gli elementi di un array tra di loro:



Ricerca Massimo/Minimo

```

vett[5] = {2, 5, 1, 0, 3};

int max = vett[0];

for (int i=0; i<5; i++) {
    if (max < vett[i]) {
        max = vett[i];
    }
}

cout << max;

// Output: 5

```

RICERCA MASSIMO

IL MASSIMO INIZIALE PUO' ESSERE QUALSIASI ELEMENTO DELL'ARRAY (PER COMODITA' PARTO DA VETT[0])

0	1	2	3	4
2	1	4	0	3

↑
MASSIMO INIZIALE

MAX = VETT[0];

```

if (max < vett[i]) {
    max = vett[i];
}

```


PASSO 1: $2 < 2?$
 PASSO 2: $2 < 1?$
 PASSO 3: $2 < 4?$ ~ MASSIMO = 4
 PASSO 4: $4 < 0?$
 PASSO 5: $4 < 3?$

MASSIMO = 4

Esercizi

35 Esercizi con gli array

31 Esercizi svolti di programmazione con gli array in C++, risolvibili anche con altri linguaggi: Pascal, Java, Python, C.

 <https://ticoprof.wordpress.com/2016/04/26/esercizi-con-gli-array/>

ticoprof

TICONSIGLIO DI STUDIARE INFORMATICA

ESERCIZIO 1

Leggere un array di interi di 10 posizioni e verificare se è palindromo (ovvero se non cambia ad essere letto dalla prima cella all'ultima o viceversa).

Esempio Output:

inserisci 10 valori: 3 5 7 9 1 1 9 7 5 3

l'array e' palindromo

SOLUZIONE:

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

int main() {
    int a[10];

    cout << "inserisci 10 valori: ";
    for (int i=0; i<10; i++) {
        cin >> a[i];
    }

    bool palindromo = true;

    for(int i=0; i<5; i++) {
        if(a[i] != a[9-i]) {
            palindromo = false;
        }
    }

    if(palindromo) {
        cout<<"l'array e' palindromo";
    }
    else {
        cout<<"l'array NON e' palindromo";
    }
}
```

ESERCIZIO 2

Leggere un array di 10 interi e stampare solo i numeri che appaiono nell'array una volta soltanto.

Esempio Output:

inserisci 10 numeri interi: 1 3 5 3 2 0 5 2 3 4

1 0 4

SOLUZIONE:

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

int main(){
    int a[10];

    //memorizzo l'array
    cout << "inserisci 10 numeri interi: ";
    for(int i=0; i<10; i++) {
        cin >> a[i];
    }

    //per ogni numero dell'array
    for(int i=0; i<10; i++) {
        // voglio contare quante volte appare a[i]
        int volte=0;

        //controllo quante volte compare a[i]
        for(int j=0; j<10; j++) {
            if(a[j] == a[i]) {
                volte++;
            }
        }
        if(volte == 1) {
            cout << a[i] << " ";
        }
    }
    return 0;
}
```