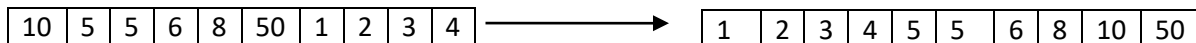


# UN ALGORITMO DI ORDINAMENTO - C++

## IL PROBLEMA DELL' ORDINAMENTO

Dato un array di interi, si vuole costruire un programma che ordini in maniera crescente l'array.

Esempio.



Vogliamo quindi un programma che preso in input un array di una qualsiasi dimensione, lo scandisca e ne ordini i suoi elementi.

## LA STRUTTURA DEL PROGRAMMA

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

//algoritmo di ordinamento
void ordina(int array[], int lunghezzaArray) {

}

void stampa(int array[], int lunghezzaArray) {

}

int main() {
    int ilMioArray[10] = {10,5,5,6,8,50,1,2,3,4};

    stampa(ilMioArray, 10);

    ordina(ilMioArray, 10);

    stampa(ilMioArray, 10);

    return 0;
}
```

Abbiamo quindi:

- Una funzione *stampa* che servirà per stampare tutti gli elementi dell'array.
- Una funzione *ordina* che conterrà il codice per ordinare l'array

Nel main:

- Abbiamo definito un array *ilMioArray*.
- Vengono stampati gli elementi di *ilMioArray*.
- Viene ordinato *ilMioArray*.
- Vengono nuovamente stampati gli elementi di *ilMioArray*.

## BUBBLESORT

Esistono vari algoritmi per ordinare un array, ciascuno che opera in modo diverso. Prendiamo in considerazione l'algoritmo *bubblesort* che pur essendo inefficiente, è facile da capire.

*Idea:*

Scandisco l'array considerando due elementi adiacenti ad ogni passo.

Se trovo due elementi adiacenti non ordinati...

Scambio i due elementi.

Altrimenti

Continuo a scandire

Ripeto il procedimento  $n$  volte dove  $n$  indica la lunghezza dell'array.

*Esempio:*

(9,2,5,8,7)

(9,2,5,8,7) → (2,9,5,8,7) → (2,5,9,8,7) → (2,5,8,9,7) → (2,5,8,7,9)

(2,5,8,7,9)

(2,5,8,7,9) → (2,5,8,7,9) → (2,5,8,7,9) → (2,5,7,8,9) → (2,5,7,8,9)

(2,5,7,8,9)

(2,5,7,8,9) → (2,5,7,8,9) → (2,5,7,8,9) → (2,5,7,8,9) → (2,5,7,8,9)

(2,5,7,8,9)

(2,5,7,8,9) → (2,5,7,8,9) → (2,5,7,8,9) → (2,5,7,8,9) → (2,5,7,8,9)

(2,5,7,8,9)

(2,5,7,8,9) → (2,5,7,8,9) → (2,5,7,8,9) → (2,5,7,8,9) → (2,5,7,8,9)

(2,5,7,8,9)

## CODICE DEL PROGRAMMA

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

//algoritmo di ordinamento
void ordina(int array[], int lunghezzaArray){
    for (int h = 0; h < lunghezzaArray; h++){

        //scandisco l'array
        for (int j = 0; j < lunghezzaArray-1; j++){
            //se trovo due elementi non ordinati
            if(array[j] > array[j+1]){
                //scambio i due elementi
                int temp = array[j];
                array[j] = array[j+1];
                array[j+1] = temp;
            }
        }
    }
}

void stampa(int array[], int lunghezzaArray){
    for (int i = 0; i < lunghezzaArray; i++){
        cout << array[i];
        cout << " ";
    }

    cout << endl;
}

int main(){
    int ilMioArray[10] = {10,5,5,6,8,50,1,2,3,4};

    stampa(ilMioArray, 10);

    ordina(ilMioArray, 10);

    stampa(ilMioArray, 10);

    return 0;
}
```