

ORDINAMENTO ARRAY mediante funzione SORT

```
#include <iostream>    //-- definizioni per gestione input/output (cin/cout)
#include <algorithm>    //-- contiene i prototipi di alcune funzioni, anche sort
#include "sortVett.h"    //-- oppure "sortVett.hpp"
using namespace std;

#define DIM 6

int iMax=5, A [DIM] = {11, 5, 2, 5, -8}; //--A [DIM -1] = A [5] passa a zero

int main(){
    char sce;
    cout<< "\n\t\t\t\t\tSORT array A di 6 interi\n";
    do {
        scriviVett();
        visualMenu();
        cin >> sce;
        switch (sce) {
            case '1': { fOrd();      break; }
            case '2': { fOrdDecr(); break; }
            case '3': { fOrdCresc(); break; }
            case 'u': return 0;
        }
    } while (1);    //--ciclo infinito -----
}

void scriviVett() {
    cout<< "\ncontenuto array A:\t";
    for (int i=0; i<DIM; i++)
        cout<< A[i] << '\t' ;
    cout<<endl;
    for (int i=1; i<=58; i++)    cout<< '-';
}

void visualMenu(){
    cout<< "\nscegliere tra :";
    cout<< "\t 1 - default\n";
    cout<< "\t\t 2 - decrescente\n";
    cout<< "\t\t 3 - crescente\n";
    cout<< "\t\t u - USCITA\n";
    cout<< "\ndigitare scelta: ";
}

}
```

Il programma **sortVett.cpp** utilizza la funzione **sort** <https://www.cplusplus.com/reference/algorithm/sort/> per ordinare l'array di interi **A**:

1. in ordine crescente (ordinamento di default),
2. decrescente (mediante la funzione d'ordine **confdecr**)
3. crescente (mediante la funzione d'ordine **confronta**).

Il modulo **sortVett.h** (va salvato nella libreria del sorgente) contiene i prototipi (cioè la **dichiarazione**) delle funzioni di programma:

```
sortVett.h - Blocco note di Windows
File Modifica Formato Visualizza ?
void fOrd();
void fOrdCresc();
void fOrdDecr();
bool confronta(const int& uno, const int& due);
bool confdecr (const int& uno, const int& due);
void scriviVett ();
void visualMenu ();
```

Fino a quando non si sceglie di uscire, il programma:

- espone a video gli elementi dell'array **A** - funzione **scriviVett()**
- espone a video il menu di scelta - funzione **visualMenu()**

```
Prompt dei comandi - sortVett.exe
>>sortVett.exe

                SORT array A di 6 interi
contenuto array A:    11    5    2    5    -8    0
-----
scegliere tra :  1 - default
                2 - decrescente
                3 - crescente
                u - USCITA
digitare scelta:  _
```

- resta in attesa della scelta
- elabora la scelta analizzando la variabile **sce** con **switch .. case** e attivando la funzione di ordinamento richiesta.

```
void fOrd(){
    cout<< "\nORDINAMENTO crescente (default)\n";
    sort(A, A + iMax); // ordina elementi da A [0] fino ad A [iMax -1]
}
//-----ordine crescente (default)

void fOrdDecr(){
    cout<< "\nORDINAMENTO decrescente (funz.confdec)\n";
    sort(A, A + iMax, confdec);
}
//--- ordine decrescente definito dalla funzione d'ordine confdec

void fOrdCresc(){// ordina elementi da A [0] fino ad A [DIM -1] = 0 (tutti)
    cout<<"\nORDINAMENTO crescente (funz.confronta)\n";
    sort(A, A + DIM, confronta);
}
//--- ordine crescente definito dalla funzione d'ordine confronta
```

//- la funzione sort richiama le funzioni d'ordine eventualmente indicate come
 //- 3^ parametro, per stabilire se necessario cambiare posizione agli elementi di A

```
bool confdecr (const int& uno, const int& due) {
    if (uno > due) //--- ordine decrescente
        return true;
    else return false;
}
```

//-la funzione booleana confdecr restituisce il valore:
 //----- VERO (true) se il parametro formale uno è maggiore del parametro due
 //-----FALSO negli altri casi, cioè se uno <= due (per gli elementi da ordinare)

```
bool confronta (const int& uno, const int& due) {
    return uno<due; //--- ordine crescente
}
```

//-la funzione booleana confronta restituisce il valore della condizione uno<due:
 //-quindi VERO se il parametro formale uno è minore del parametro due
 //----- FALSO negli altri casi, cioè se uno >= due
 //- la funzione booleana confronta equivale alla relazione d'ordine di default

Con la scelta **1** vengono ordinati i primi 5 elementi dell'array **A**, da **A[0] = 11** ad **A[4] = -8** (NOTA 4 = iMax-1):

```
digitare scelta: 1
ORDINAMENTO crescente (default)
contenuto array A: -8 2 5 5 11 0
-----
scegliere tra : 1 - default
                2 - decrescente
                3 - crescente
                u - USCITA
digitare scelta: _
```

Con la scelta **2** vengono posti in **ordine decrescente** i primi 5 elementi di **A**:

```
digitare scelta: 2
ORDINAMENTO decrescente (funz.confdec)
contenuto array A: 11 5 5 2 -8 0
-----
scegliere tra : 1 - default
                2 - decrescente
                3 - crescente
                u - USCITA
digitare scelta: _
```

Con la scelta **3** vengono posti in **ordine crescente** **tutti** gli elementi di **A**:

```
digitare scelta: 3
ORDINAMENTO crescente (funz.confronta)
contenuto array A: -8 0 2 5 5 11
-----
scegliere tra : 1 - default
                2 - decrescente
                3 - crescente
                u - USCITA
digitare scelta: u
>>
```

La scelta **u** consente l'uscita dal programma.