

```

#include <iostream>
using namespace std;
#include <string>
#include <algorithm>
#define DIM 4
#define ACAPO '\n'
#define TAB '\t'

struct Atleta {           //-- STRUTTURA (o RECORD) di nome Atleta
    string nome;
    string naz;
    int punti;
};
//--PROTOTIPI delle funzioni (le IMPLEMENTAZIONI sono a pagina successiva)

void inicializza (Atleta A[], int tot);
void scrittura (Atleta T[], int tot);
bool xnome (const Atleta& x, const Atleta& y);
bool xpunti (const Atleta& a, const Atleta& b);

int main() {

    Atleta atleti [DIM]; // Tabella = Array di Struttura
                        // = Array di tipo Atleta
    inicializza (atleti, DIM);

    cout << "\n\n" << DIM << " atleti registrati\n";
    scrittura (atleti, DIM);

    sort (atleti, atleti+DIM, xnome);

    cout << "\n-- atleti in ordine alfabetico --\n";
    scrittura (atleti, DIM);

    sort (atleti, atleti+DIM, xpunti);

    cout << "\n--- GRADUATORIA atleti ---\n";
    scrittura (atleti, DIM);
}

```

Il programma `sortTabella_atleti.cpp` :

1. inicializza, mediante la funzione `inicializza`, la **Tabella atleti** (Array di Struttura `Atleta`) di 4 elementi (**DIM** vale **4**)
 2. espone la Tabella **atleti** a video mediante la funzione `scrittura` utilizza poi la funzione `sort` <https://www.cplusplus.com/reference/algorithm/sort/> per ordinare la Tabella **atleti** in ordine:
 3. di nome crescente (ordine alfabetico, funzione d'ordine `xnome`),
 4. di punteggio decrescente (funzione d'ordine `xpunti`)
- esponendo a video (mediante la funzione `scrittura`) la Tabella **atleti** dopo ogni ordinamento.

A differenza che per gli Array, per ordinare Tabelle (Array di strutture dati complesse composte da più campi di tipo diverso) la funzione `sort` necessita delle **funzioni d'ordine** stabilite dal programmatore, sia per ordinamenti decrescenti che per ordinamenti crescenti: va stabilito, infatti, in base a quale campo (o quali campi) ordinare i diversi elementi dell'array, spostando l'intero record (struttura) nell'array.

//-- IMPLEMENTAZIONE delle funzioni (i PROTOTIPI precedono il main)

```

void inizializza (Atleta A[], int tot){
    A[0].nome = "Bill";
    A[0].naz = "USA";
    A[0].punti = 7;
    A[1].nome = "aldo";
    A[1].naz = "ITA";
    A[1].punti = 9;
    A[2].nome = "Charlie";
    A[2].naz = "AUS";
    A[2].punti = 4;
    A[3].nome = "Abdul";
    A[3].naz = "EGY";
    A[3].punti = 9;
}

bool xnome (const Atleta& x, const Atleta& y) {
    if (x.nome < y.nome) //--- ordine crescente di nome
        return true; //--- cioè in ordine alfabetico crescente
    else return false;
}

bool xpunti (const Atleta& a, const Atleta& b) {
    if (a.punti > b.punti) //--- ordine per punteggio decrescente
        return true;
    else
        if (a.punti == b.punti //--- a parità di punteggio
            && (a.nome > b.nome)) //--- si ordina per nome decrescente
            return true;
        else return false;
}

void scrittura (Atleta T[], int tot){
    for (int i=0; i<tot; i++){
        cout << "atleta " << i << TAB;
        cout << T[i].naz << TAB;
        cout << T[i].nome << TAB;
        cout << T[i].punti << ACAPO;
    }
}
    
```

OUTPUT del programma

```

CAL Prompt dei comandi - □ ×
>>sortTabella_atleti.exe

4 atleti registrati
atleta 0      USA      Bill      7
atleta 1      ITA      aldo      9
atleta 2      AUS      Charlie   4
atleta 3      EGY      Abdul     9

-- atleti in ordine alfabetico --
atleta 0      EGY      Abdul     9
atleta 1      USA      Bill      7
atleta 2      AUS      Charlie   4
atleta 3      ITA      aldo      9

--- GRADUATORIA atleti ---
atleta 0      ITA      aldo      9
atleta 1      EGY      Abdul     9
atleta 2      USA      Bill      7
atleta 3      AUS      Charlie   4

>>
    
```

Nelle codifica ASCII il carattere **a** del nome **aldo** è maggiore (>) del carattere **C** del nome **Charlie** quindi il record relativo al nome **aldo** (e tutti i suoi dati: naz e punti) viene posto dalla funzione **sort** dopo il record relativo al nome **Charlie** nella **Tabella atleti**

Nel secondo sort il record relativo ad **aldo** viene posto prima del record con nome **Abdul** perché a parità di punteggio (9) i record vengono ordinati per nome in ordine alfabetico decrescente e la **a** del nome **aldo** ha codice ASCII maggiore della **A** di **Abdul**.