```
#include <iostream>
using namespace std;
#include <string.h>
#include <fstream>

#define DIM 12
#define SP  ' '

struct Atleti {
 char NOME[DIM];
 int  punti;
};

int main(){
  ofstream fout_t ("classifica.txt");
  ofstream fout_b ("classifica.dat", ios::binary);

  Atleti atleta;

  char NOME[DIM];
  int  punti;

  for (int i=0; i<4; i++){
      cout << "\natleta? ";
      cin >> NOME >> punti;

      fout_t << NOME << SP << punti << endl;

      for (int k=0; k<DIM; k++)
          atleta.NOME[k] = SP;    //-- spazio in NOME --

      strcpy (atleta.NOME, NOME);
      atleta.punti = punti;

      fout_b.write( (char *) &atleta, sizeof(Atleti) );
  }
  fout_t.close(); fout_b.close();
  return 0;
}
```

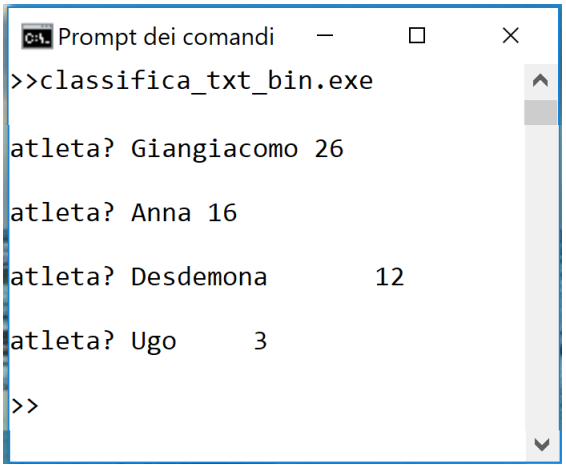
Il programma ***classifica_txt_bin.cpp*** acquisisce da tastiera nome e punteggio di 4 persone e li memorizza in un **file di testo** `"classifica.txt"` (nome logico **fout_t**) e in un **file binary** `"classifica.dat"` (nome logico **fout_b**).

I dati vengono letti nella variabile elementare `punti` e nell'array di caratteri `NOME` (secondo la gestione delle stringhe nel C), dati che vengono poi copiati nei corrispondenti campi del record **atleta** di tipo **Atleti** (dati strutturati); non c'è ambiguità per i nomi delle variabili definite nel main e nella struttura perchè per riferirsi ai campi di un record occorre indicare prima il record (es: **atleta.punti**).

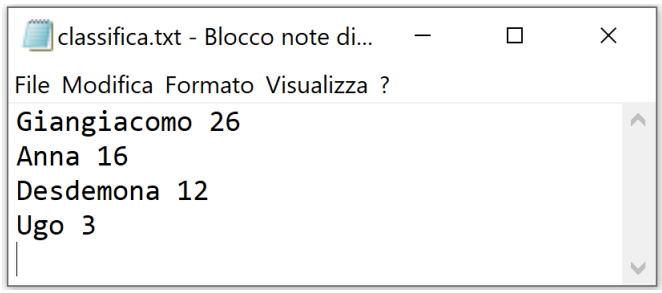
ATTENZIONE: per gli array di caratteri **non è possibile usare gli operatori di assegnazione e confronto**; al loro posto vanno utilizzate le funzioni di copia (**strcpy**) e confronto (**strcmp**) e includere **<string.h>**

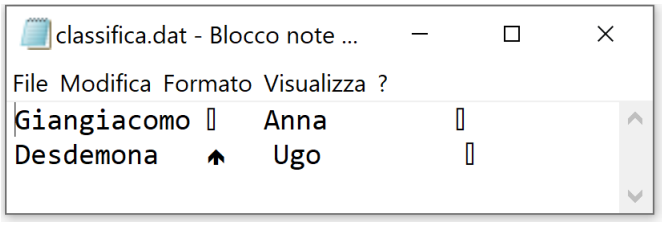
La funzione **strcpy** (http://www.cplusplus.com/reference/cstring/strcpy/) copia dall'array **NOME** all'array **atleta.NOME** un byte alla volta fino al terminatore di stringa ('\0' NULL character 0_{hex} = 0x00). Se non si rimette spazio nei byte successivi, i nomi dal secondo al quarto non puliscono le celle successive.

INPUT da tastiera

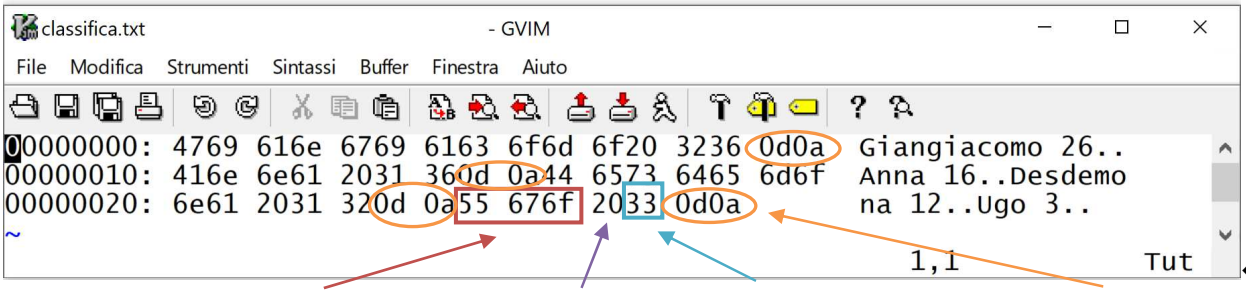


2 FILE di OUTPUT



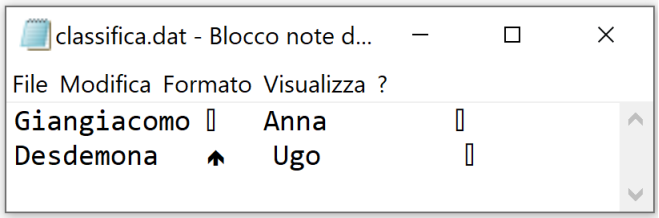


OUTPUT su file ASCII: tutti i caratteri sono scritti in chiaro (anche le cifre), rappresentati in **codice ASCII**

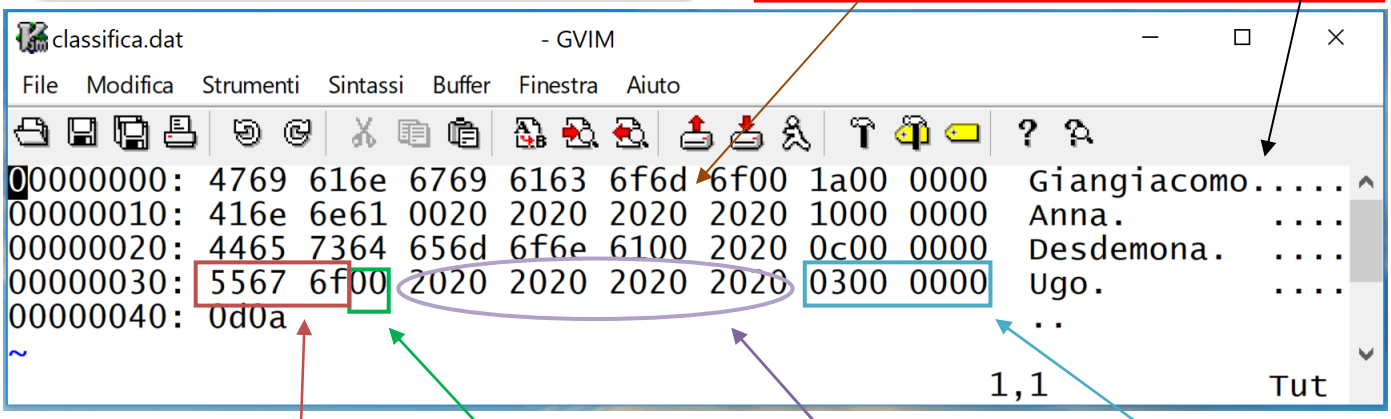


3 caratteri di "Ugo" - 1 carattere a spazio – 1 byte per il carattere '3' – A CAPO (endl)

OUTPUT su file binary **ripulendo atleta.NOME** (istruzione `atleta.NOME[k] = SP;`)



Aprendo il file con l'editor *Vim* e selezionando Strumenti > Converti a esadecimale, sulla destra si vedono i singoli caratteri; al centro, per ogni carattere si vede il codice ASCII espresso in esadecimale.



3 caratteri di "Ugo" - terminatore di stringa - 8 caratteri a spazio – 4 byte per l'intero (3_{16})

OUTPUT su file binary **SENZA ripulire atleta.NOME**