

## Es. 8.1 – Gestione delle iscrizioni a una gara

### PRIMA PARTE

Scrivere un programma che gestisca la lista degli atleti iscritti a una gara. Ogni atleta è identificato attraverso il proprio nome, nella forma *nome.cognome*, e il numero di pettorale. Supporre che la lunghezza di *nome.cognome* sia al più 30 caratteri. Utilizzare la seguente struttura dati per il singolo atleta:

```
struct elem{
    char nome[31];
    int pettorale;
    elem* pun;
};
```

Realizzare le seguenti funzioni che agiscono sulla lista:

- *init*: inizializza la lista. Inizialmente la lista è vuota.
- *visualizza*: visualizza il nome e il pettorale degli atleti nella lista, secondo l'ordine di inserimento nella lista.
- *aggiungi*: aggiunge una coppia nome-pettorale alla lista di iscritti e restituisce il valore 1. Se il nome o il pettorale sono già presenti nella lista, l'inserimento non viene effettuato e la funzione restituisce il valore 0.

Definire opportunamente i file *iscrizione.h* e *iscrizione.cpp*, contenenti rispettivamente la dichiarazione e l'implementazione delle funzioni.

Utilizzare il seguente *main* di prova:

```
#include <iostream>
#include "iscrizione.h"
using namespace std;

int main() {
    elem* L;
    init(L);
    visualizza(L);

    aggiungi(L, "mario.bianchi", 24);
    aggiungi(L, "sergio.rossi", 6);
    aggiungi(L, "sergio.rossi", 4); // gia' presente
    aggiungi(L, "gianni.verdi", 32);
    aggiungi(L, "luigi.neri", 6); // gia' presente

    visualizza(L);
    return 0;
}
```

L'output desiderato è il seguente:

```
Atleti iscritti:

Atleti iscritti:
mario.bianchi, 24
sergio.rossi, 6
gianni.verdi, 32
```

## SECONDA PARTE

Definire le seguenti funzioni aggiuntive:

- *cerca*: cerca un nome all'interno della lista. Se il nome è presente, restituisce il valore 1. Altrimenti restituisce il valore 0.
- *elimina*: dato un nome, elimina l'atleta corrispondente dalla lista e restituisce il valore 1. Se il nome non è presente, la lista rimane inalterata e la funzione restituisce il valore 0.

Utilizzare il seguente *main* di prova:

```
#include "iscrizione.h"
#include <iostream>
using namespace std;

int main() {
    elem* L;
    init(L);
    visualizza(L);

    aggiungi(L, "mario.bianchi", 24);
    aggiungi(L, "sergio.rossi", 6);
    aggiungi(L, "sergio.rossi", 4); // gia' presente
    aggiungi(L, "gianni.verdi", 32);
    aggiungi(L, "luigi.neri", 6); // gia' presente

    visualizza(L);

    cout << cerca(L, "mario.bianchi") << endl;
    elimina(L, "mario.bianchi");
    cout << cerca(L, "mario.bianchi") << endl;
    visualizza(L);

    return 0;
}
```

L'output desiderato è il seguente:

```
Atleti iscritti:

Atleti iscritti:
mario.bianchi, 24
sergio.rossi, 6
gianni.verdi, 32

1
0
Atleti iscritti:
sergio.rossi, 6
gianni.verdi, 32
```

## Es. 8.2 – Gestione delle iscrizioni a una gara (implementazione tramite classe)

Scrivere una classe di nome *Iscrizione*, che gestisca la lista degli atleti iscritti a una gara. Ogni atleta è identificato attraverso il proprio nome, nella forma *nome.cognome*, e il numero di pettorale. Supporre che la lunghezza di *nome.cognome* sia al più 30 caratteri. Dotare la classe delle seguenti funzioni membro:

- *Costruttore di default*: inizializza la lista. Inizialmente la lista è vuota.
- *aggiungi*: aggiunge una coppia nome-pettorale alla lista di iscritti e restituisce il valore 1. Se il nome o il pettorale sono già presenti nella lista, l'inserimento non viene effettuato e la funzione restituisce il valore 0.
- *visualizza*: visualizza il nome e il pettorale degli atleti nella lista, secondo l'ordine di inserimento nella lista.
- *cerca*: cerca un nome all'interno della lista. Se il nome è presente, restituisce il valore 1. Altrimenti restituisce il valore 0.
- *elimina*: dato un nome, elimina l'atleta corrispondente dalla lista e restituisce il valore 1. Se il nome non è presente, la lista rimane inalterata e la funzione restituisce il valore 0.

Definire opportunamente i file *class\_iscrizione.h* e *class\_iscrizione.cpp*, contenenti rispettivamente la dichiarazione della classe e l'implementazione delle sue funzioni membro.

Utilizzare il seguente *main* di prova:

```
#include "class_iscrizione.h"
#include <iostream>
using namespace std;

int main() {
    Iscrizione L;
    L.visualizza();

    L.aggiungi("mario.bianchi", 24);
    L.aggiungi("sergio.rossi", 6);
    L.aggiungi("sergio.rossi", 4); // gia' presente
    L.aggiungi("gianni.verdi", 32);
    L.aggiungi("luigi.neri", 6); // gia' presente

    L.visualizza();

    cout << L.cerca("mario.bianchi") << endl;
    L.elimina("mario.bianchi");
    cout << L.cerca("mario.bianchi") << endl;
    L.visualizza();

    return 0;
}
```

L'output desiderato è il seguente:

```
Atleti iscritti:

Atleti iscritti:
mario.bianchi , 24
sergio.rossi , 6
gianni.verdi , 32

1
0
Atleti iscritti:
sergio.rossi , 6
gianni.verdi , 32
```