FONDAMENTI DI INFORMATICA - PROVA SCRITTA 23-01-2025

CORSO DI LAUREA IN INGEGNERIA BIOMEDICA

Una `GaraPodistica` consiste di tre marce e per ogni marcia si mantengono informazioni sugli iscritti alla marcia. Le marce sono numerate a partire da 1. Un atleta puo' iscriversi a più marce. Ogni iscritto è identificato univocamente dal nome e dall'età. Il nome è una stringa di al massimo 30 caratteri. L'età deve essere compresa fra 18 e 50. Implementare le seguenti operazioni che possono essere effettuate su una `GaraPodistica`:

- `Inizializza(g);`
  Funzione che inizializza una gara podistica `g`. Inizialmente non ci sono iscritti a nessuna marcia.

- `Iscrizione (g, n, s, d);`
  Funzione che iscrive alla marcia `n`, della gara podistica `g` l'atleta di nome `s` ed età `d`. Non ci possono essere due iscritti con lo stesso nome e stessa età alla stessa marcia.

- `Eliminazione(g, n, s, d);`
  Funzione che elimina l'atleta di nome `s` e età `d` dalla marcia `n` della gara podistica `g`. Restituisce `true` se l'operazione ha successo; `false` altrimenti.

- `QuantiIscritti(g, n);`
  Funzione che restituisce il numero di iscritti alla marcia `n` della gara podistica `g`.

- `MediaEta (g, n);`
  Funzione che restituisce l'età media degli iscritti alla marcia `n` della gara podistica `g`.

- `LasciaGara(g, s, d);`
  Funzione che elimina l'atleta di nome `s` e età `d` da tutte le marce della gara `g`. Restituisce `true` se l'atleta era iscritto ad almeno una marcia; `false` altrimenti.

- `Stampa(g);`
  Funzione di uscita di una `GaraPodistica g`, che stampa per ogni marcia, il numero della marcia racchiuso fra parentesi angolate, seguito dai nomi degli iscritti alla marcia nell'ordine di iscrizione. Gli iscritti sono separati da ','. Le marce sono visualizzate su righe separate.
  Nel caso seguente la marcia numero 1 ha 3 iscritti: Rossi è l'atleta che si è iscritto per primo, Bianchi l'atleta che si è iscritto per ultimo. La marcia numero 2 non ha iscritti e la marcia numero 3 ha due iscritti.
  <1> Rossi, Verdi, Bianchi
  <2>
  <3> Neri, Rossi

Mediante il linguaggio C++, realizzare il tipo `GaraPodistica` definito dalle precedenti specifiche utilizzando le strutture. Gestire le eventuali situazioni di errore.

Esempio di funzione main()

Contenuto dello stream d'output standard mostrato nel commento /* */

```
int main(){
  GaraPodistica g;
  Inizializza (g);
  Stampa(g);
/*
<1>
<2>
<3>     */

Iscrizione(g, 3, "Neri", 25);
Iscrizione(g, 1, "Rossi", 18);
Iscrizione(g, 1, "Verdi", 35);
Iscrizione(g, 2, "Neri", 25);
Stampa(g);
/*
<1> Rossi, Verdi
<2> Neri
<3> Neri   */

Eliminazione(g, 2, Neri, 25);
Stampa(g);
/*
<1> Rossi, Verdi
<2>
<3> Neri   */

cout << QuantiIscritti (g, 1) << endl;
/*
2   */

cout << MediaEta (g, 1) << endl;
/*
26.5  */

Iscrizione(g, 1, "Neri", 25);
Stampa(g);
/*
<1> Rossi, Verdi, Neri
<2>
<3> Neri  */

LasciaGara(g, Neri, 25);
Stampa(g);
/*
<1> Rossi, Verdi
<2>
<3>    */


return 0;
}
```