

Esercizi sulle strutture

Roberto Borelli

13 Dicembre 2023

1 Descrizione dell'Esercizio

Creare un gestionale di studenti (al massimo 50) utilizzando un array di strutture. Ogni struttura `Studente` contiene informazioni come nome, cognome, età e voto medio. L'obiettivo è implementare una serie di funzioni per interagire con questo elenco di studenti.

2 Struttura Studente e Main

```
#include <stdio.h>
#include <string.h>
struct Studente {
    char nome[50];
    char cognome[50];
    int eta;
    float votoMedio;
};

int main(){
    struct Studente elencoStudenti[50] = { }; // array vuoto di studenti
    int numStudenti = 0;
}
```

3 Prototipi dei Metodi da implementare

1. `void inserisciStudente(struct Studente listaStudenti[], int *numStudenti);`

Descrizione: Aggiunge un nuovo studente all'array di studenti, chiedendo all'utente di inserire nome, cognome, età e voto medio. Dopo la chiamata alla funzione, la variabile nel main `numStudenti` dovrà essere stata incrementata di 1. Se nell'array sono già presenti 50 studenti, la funzione stampa un errore.

2. `void stampaListaStudenti(struct Studente listaStudenti[], int numStudenti);`

Descrizione: Stampa i dettagli (nome, cognome, età e voto medio) di tutti gli studenti presenti nell'array.

3. `struct Studente studenteVotoMigliore(struct Studente listaStudenti[], int numStudenti);`

Descrizione: Restituisce lo studente con il voto medio più alto all'interno dell'array. Se non ci sono studenti la funzione stampa un errore. Se ci sono due o più studenti con la media massima, ne viene restituito solo uno, ma la funzione stampa un messaggio avvisando che il migliore non è unico.

4. `void rimuoviStudente(struct Studente listaStudenti[], int *numStudenti, char nome[]);`

Descrizione: Rimuove uno studente dall'array utilizzando il suo nome come parametro di ricerca. Se nell'array non è presente uno studente con tale nome la funzione stampa un errore. La variabile `numStudenti` deve essere decrementata se necessario.

4 Esempio di esecuzione

```
// CODICE DI ESEMPIO

#include <stdio.h>
#include <string.h>
struct Studente {
    char nome[50];
    char cognome[50];
    int eta;
    float votoMedio;
};

int main(){
    struct Studente studs[50] = { };
    int n = 0;

    stampaListaStudenti(studs, n);

    inserisciStudente(studs, &n);
    inserisciStudente(studs, &n);
    inserisciStudente(studs, &n);

    printf("%d\n", n);

    stampaListaStudenti(studs, n);

    Studente best = tudenteVotoMigliore(studs,
    &n);

    printf("%s\n", best.nome);

    rimuoviStudente(studs, &n, best.nome);

    printf("%d\n", n);

    stampaListaStudenti(studs, n);

    return 0;
}

// COMMENTI E INPUT/OUTPUT

// Viene stampato
// "Non ci sono studenti registrati"

// Supponiamo di fare i seguenti inserimenti
// inseriamo Mario Rossi 21 anni, 27.2
// inseriamo Marco Bianchi 22 anni, 30
// inseriamo Andrea Verdi 20 anni, 22

// Viene stampato: "n=3"

// Viene stampato:
// "Gli studenti registrati sono
// Mario, Rossi, 21, 27.2
// Marco, Bianchi, 22, 30
// Andrea, Verdi, 20, 22"

// lo studente migliore e' Marco bianchi

// Viene stampato "Marco Bianchi"

// Viene rimosso lo studente Marco Bianchi

// Viene stampato: "n=2"

// Viene stampato:
// "Gli studenti registrati sono
// Mario, Rossi, 21, 27.2
// Andrea, Verdi, 20, 22"

// il programma termina
```