

Esercizi Lettura Scrittura File

Roberto Borelli

21 Dicembre 2023

1 Struttura Studente

In relazione all'esercizio sulla struttura studente¹ creare un sistema di salvataggio del database degli studenti. Di seguito ricordiamo come è fatta la struttura studente.

```
struct Studente {
    char nome[50];
    char cognome[50];
    int eta;
    float votoMedio;
};
```

Ricordiamo inoltre che il main gestisce un elenco di al più 50 studenti e viene tenuto in memoria il numero di studenti corrente.

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

Si implementino (nello stesso file in cui si sono implementate le funzioni `stampaListaStudenti` ecc...) e testino le seguenti funzioni.

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

Descrizione: Viene creato il file `studenti.txt`. Nella prima riga viene stampato il numero di studenti (chiamiamolo n). Le successive n righe contengono i dati di ciascun studente. Ad esempio se `listaStudenti` contiene Mario Rossi anni 21 voto medio 27.2, Marco Bianchi anni 22 voto medio 30, Andrea Verdi anni 20 voto medio 22 il file creato dovrà avere il seguente contenuto.

```
3
Mario Rossi 21 27.2
Marco Bianchi 22 30
Andrea Verdi 20 22
```

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

Descrizione: Viene aperto in lettura il file `studenti.txt` che avrà lo stesso formato mostrato nel punto precedente. Si riempra opportunamente l'array `listaStudenti` a partire dalla posizione 0 e si modifichi il valore puntato dalla variabile `numStudenti`. Gestire il caso in cui il file non esista e il caso in cui il file contenga più di 50 item.

2 Numeri Naturali

Il file `numeri.txt` contiene i primi n (per un qualche $n > 1$) numeri naturali a partire da 1. Ogni numero è memorizzato su una singola riga. Ad esempio se $n = 5$ il file contiene il seguente testo.

```
1
2
3
4
5
```

Creare un programma con le seguenti specifiche.

¹<https://robertoborelli.com/pdfs/teaching/Esercizi%20Struttura%20Studente.pdf>

- Se il file `numeri.txt` esiste, il programma lo apre e ci aggiunge in coda il successivo numero naturale. Ad esempio se il file conteneva i primi 5 numeri come mostrato, dopo aver eseguito il programma il file contiene i primi 6 numeri.

```
1
2
3
4
5
6
```

- Se il file non esiste, il programma lo crea e ci stampa una sola linea contenente il numero 1.

3 Query

Il file `query.txt` contiene una sola linea che consiste di una sola parola. Il file `db.txt` contiene un numero arbitrario di linee e ogni linea contiene una parola. Creare un programma che legga questi due file e crei il file `answ.txt` che consiste di due righe. Nella prima linea dovrà essere stampato il numero di parole lette in `db.txt` e nella seconda linea verrà stampato il numero di occorrenze della parola del file `query.txt` nel file `db.txt`. Ad esempio se il file `query.txt` contiene la parola `aaa` e il file `db.txt` contiene il seguente contenuto

```
aaa
aab
aba
abb
baa
bab
bba
bbb
aaa
aaab
aabc
```

allora il file `answ.txt` conterrà nella prima linea il valore 11 e nella seconda linea il valore 3.