

Traccia di Elementi di Programmazione

9 Gennaio 2020

Sviluppare un programma in C che legge da file una matrice di numeri reali.

- Il programma elimina le righe la cui media è un numero negativo
- Per ogni riga della matrice risultante creare un vettore di struct i cui elementi contengano:
 - la media della riga,
 - l'indice di riga,
 - il numero di colonne
 - e i valori della riga
- Realizzare due distinte funzioni che ordinano il vettore di struct per indice di riga e per media crescente
- Salvare il vettore di struct in un file binario
- Rileggere e stampare a video il vettore di struct da file binario.

Esempio

matrice iniziale:

1	-1	0	5
3.0	3.2	1	-3
5	1	3	-15
0	5	1	0
8	3	1	-16

eliminazione righe:

1	-1	0	5
3.0	3.2	1	-3
0	5	1	0

Vettore di struct:

```
{0, 1.25, 4, [1, -1, 0, 5]}  
{1, 1.05, 4, [3.0, 3.2, 1, -3]}  
{2, 1.5, 4, [0, 5, 1, 0]}
```

Traccia di Elementi di Programmazione

9 Gennaio 2020

Sviluppare un programma in C che legge da file delle linee di testo.

- Il programma deve trasformare le linee di testo in un vettore di struct i cui campi sono:
 - la parola trovata
 - Il numero di occorrenze
 - gli indici di linea di ogni occorrenza
 - Gli indice di parola nella linea di ogni occorrenza
- Sviluppare una funzione che ordina gli elementi del vettore di struct in modo che le parole siano in ordine alfabetico.
- Salvare il vettore di struct in un file binario.
- Rileggere e stampare a video il vettore di struct da file binario.

Esempio:

```
sopra la panca la capra campa
sotto la panca la capra crepa
```

Vettore di struct:

```
{"sopra", 1, [1], [1]}
{"la", 4, [0,0,1,1], [1.3.1.3]}
{"panca", 2, [0,1], [2,2]}
{"capra", 2, [0,1], [4,4]}
{"campa", 1, [0], [5]}
{"sotto", 1, [1], [0]}
{"crepa", 1, [1], [5]}
```

Vettore ordinato:

```
{"campa", 1, [0], [5]}
{"capra", 2, [0,1], [4,4]}
{"crepa", 1, [1], [5]}
{"la", 4, [0,0,1,1], [1.3.1.3]}
{"panca", 2, [0,1], [2,2]}
{"sopra", 1, [1], [1]}
{"sotto", 1, [1], [0]}
```