

Università di Catania  
Dipartimento di Matematica e Informatica  
Corso di Studio in Informatica, A.A. 2022-2023  
Compito di Programmazione 1 e Laboratorio F-N  
27 giugno 2023

### Descrizione del programma

Si scriva un programma C che:

- A. Prenda in input da riga di comando un parametro stringa *filename* che contenga un nome di file di output (ad esempio “file\_di\_output.bin”) e un parametro intero *n*. Il programma controlli che il nome del file abbia estensione “bin” e che *n* sia compreso tra 3 e 5. Se i parametri passati non rispettano i requisiti richiesti, il programma stampa un messaggio di errore sullo standard error e termina la sua esecuzione con un appropriato codice di terminazione.

*Si stampino a schermo i valori dei parametri presi in input.*

- B. Chieda all'utente di inserire da riga di comando un numero arbitrario di record bancari separati da caratteri di “invio”. Ciascun record deve seguire il formato:

numeroConto nome cognome saldo

Ciascun record viene inserito in una apposita struct *Record* (si assumano stringhe lunghe al massimo 255 caratteri) e inserito in una lista concatenate *L*. L'inserimento termina quando l'utente dà il carattere di end of file (CTRL + D o CTRL + Z a seconda del sistema).

*Si stampi a schermo il contenuto di L dopo le operazioni di inserimento.*

- C. Dichiarare un array *X* di struct *Record* e copiare i record dalla lista *L* all'interno di *X*. Ordinare mediante selectionsort gli elementi di *X* in ordine ascendente rispetto al seguente valore:

$$(\text{saldo}[i] - m) * (\text{saldo}[i] - m)$$

Dove  $\text{saldo}[i]$  è il saldo dell'*i*-esimo elemento di *X*, mentre *m* è il saldo medio.

*Si stampi a schermo il contenuto di X al termine dell'operazione.*

- D. Inizializzare un file binario in *filename* e inserire al suo interno gli elementi di *X*. Leggere dunque in ordine casuale e stampare a schermo i primi 5 elementi in *filename*.

Nota: gestire opportunamente i casi in cui i file non possono essere correttamente aperti in lettura o scrittura stampando un errore sullo standard error e terminando l'esecuzione del programma.

### Specifiche

Il programma potrà essere articolato in un unico file sorgente, ma dovrà contenere almeno le seguenti funzioni con opportuni parametri formali:

- **decodeParameters**: funzione che prende in input il numero argc e il vettore argv ricevuti in input dalla funzione main(), controlli la presenza e i requisiti degli argomenti e li inserisce in un record (struct) da restituire allo user code (funzione main). La funzione deve gestire correttamente gli errori relativi a input non corretti;
- **readRecord**: legge un singolo record da riga di comando e restituisce una struct *Record*;
- **loadRecords**: legge tutti i record da riga di comando e li conserva nella lista *L*;

- **getArray**: funzione che prende in input la pila  $S$  e restituisce l'array  $X$ , come definito nel testo;
- **sort**: funzione che permette di ordinare l'array  $X$  mediante selectionsort;
- **saveToFile**: funzione che salva il contenuto di  $X$  su file;
- **showFileContent**: funzione che legge il contenuto del file e lo mostra a schermo.

### Note

- **Durata della prova**: 120 minuti
- **È VIETATO** usare variabili globali.
- **Si inseriscano i file sorgenti** direttamente nella propria home directory.
- **Accesso alla documentazione** disponibile tramite il browser al link: <https://devdocs.io/c/>.

### Output di controllo

Si consideri il seguente file “input” (troverete il file nella vostra home directory):

```
123456 John Doe 6253.50
987654 Alice Smith 166.75
567890 Michael Johnson 93826.20
234567 Sarah Williams 177625.00
876543 David Brown 8276.80
345678 Emily Miller 222763.25
654321 James Anderson 29376.50
432109 Olivia Thomas 166524.10
765432 Robert Wilson 888827.75
210987 Sophia Jackson 2736.60
```

Eseguendo il programma con il comando: `./soluzione out.bin 4 < input.txt`

il programma stamperà su standard output il seguente contenuto (lo trovate nel file “output.txt” nella home directory):

```
PUNTO A - valori dei parametri presi in input:
filename = s.bin
n = 4
```

```
PUNTO B - Contenuto di L:
123456 John Doe 6253.50
987654 Alice Smith 166.75
567890 Michael Johnson 93826.20
234567 Sarah Williams 177625.00
876543 David Brown 8276.80
345678 Emily Miller 222763.25
654321 James Anderson 29376.50
432109 Olivia Thomas 166524.10
765432 Robert Wilson 888827.75
210987 Sophia Jackson 2736.60
```

PUNTO C - Contenuto di X dopo l'ordinamento:

432109 Olivia Thomas 166524.10  
234567 Sarah Williams 177625.00  
345678 Emily Miller 222763.25  
567890 Michael Johnson 93826.20  
654321 James Anderson 29376.50  
876543 David Brown 8276.80  
123456 John Doe 6253.50  
210987 Sophia Jackson 2736.60  
987654 Alice Smith 166.75  
765432 Robert Wilson 888827.75

PUNTO D - Contenuto del file binario in ordine casuale:

345678 Emily Miller 222763.25  
567890 Michael Johnson 93826.20  
432109 Olivia Thomas 166524.10  
234567 Sarah Williams 177625.00