

# 12BHD INFORMATICA, A.A. 2018/2019

## Esercitazione di Laboratorio 11

---

### Obiettivi dell'esercitazione

- Scrivere programmi complessi con lettura da file e uso delle strutture

### Contenuti tecnici

- Consolidamento della lettura di dati da file
- La definizione di strutture dati
- typedef e .

---

### Da risolvere preferibilmente in laboratorio

Esercizio 1. Scrivere un programma che permetta di memorizzare un insieme di coordinate del piano cartesiano. Nel programma si definisca una struttura dati contenente due campi come segue:

```
struct coordinate
{
    int x;
    int y;
};
```

Il programma deve acquisire da tastiera le coordinate  $x$  e  $y$  di 4 punti, corrispondenti ai punti toccati da un percorso e manipolare i dati ricevuti in base ad un parametro ricevuto da linea di comando:

- Parametro “-m” : calcola e stampa a video la lunghezza del percorso composto dai 4 segmenti.
- Parametro “-a” : calcola e stampa a video la distanza minima tra le coordinate inserite.

Esercizio 2. Si scriva un programma per la gestione di una rubrica di massimo 100 persone. Il programma deve permettere la memorizzazione dei seguenti dati in una struttura:

```
typedef struct nomi {
    - Nome                char nome[20];
    - Cognome             char cognome[20];
    - Numero di telefono fisso  char fisso[20];
    - Numero di telefono mobile char mobile[20];
} Nomi;
```

Il programma deve permettere all'utente di effettuare, tramite scelta da menu, l'inserimento di un nuovo nominativo (tollerando eventuali duplicazioni di nome e cognome, ma segnalandole all'utente e chiedendo un'ulteriore conferma), e la stampa dell'elenco completo.

Esercizio 3. <sup>1</sup>Si scriva un programma in linguaggio C che legga il contenuto di un file dopo averne ricevuto il nome da linea di comando. Il numero di righe del file sia al massimo 80 e ciascuna riga del file contenga i seguenti campi, ciascuno composto al massimo da 20 caratteri e privo di spazi

*<materia> <nome prof> <cognome prof> <periodo> <crediti> <% superamento esame>*

Il programma dovrà stampare a video:

- a. il nome della materia che assegna più crediti in assoluto
- b. per ciascun periodo didattico (considerandone al massimo 4), la materia più difficile da superare

Il programma dovrà infine richiedere l'inserimento da tastiera di un cognome di professore (massimo 20 caratteri) e stampare a video:

- c. la somma dei crediti assegnati dalle materie che insegna
- d. la media di superamento degli esami da lui tenuti.

Approfondimento: nei punti a) e b), se si verificasse il caso in cui siano presenti due o più materie che assegnano il numero massimo di crediti o abbiano nello stesso periodo il minor tasso di superamento, il programma dovrà stampare l'elenco completo delle materie identificate

Esempio: di seguito un possibile contenuto di file:

*Equitazione Donato Cavallo 1 5 50*

*Canottaggio Remo Controcorrente 2 4 70*

*Velocità Tina Svelta 1 10 80*

#### Da risolvere a casa

Esercizio 4. Si scriva un programma che legga da un file (il cui nome è ricevuto come primo parametro sulla linea di comando) alcune informazioni ferroviarie. Per ciascuna linea, il file contiene le seguenti informazioni (ciascuno dei campi non superi i 20 caratteri di lunghezza e sia privo di spazi)

*<stazione\_partenza> <ora\_partenza> <stazione\_arrivo> <ora\_arrivo>*

Il programma può eventualmente ricevere come ulteriore parametro da linea di comando il nome di una città; in tale caso, il programma calcoli e stampi il numero di treni in arrivo ed il numero di treni in partenza da tale città (se inclusa nell'elenco).

---

<sup>1</sup> Questo esercizio sarà svolto in modo multimediale e inserito sul Portale, tra il materiale comune, nelle settimane successive.

- Esercizio 5. Scrivere un programma in linguaggio C per la gestione di una gara che:
- legga da file di tipo testo gara.txt una sequenza, in ordine qualsiasi, di nomi e relativo punteggio. I nomi s'intendono senza spazi in mezzo e il punteggio è un intero; si può ipotizzare che ci siano al più 100 atleti;
  - visualizzi sul video l'elenco ordinato in senso alfabetico, con relativo punteggio;
  - contemporaneamente effettui l'output su un file di tipo testo, garaord.txt degli stessi dati ordinati;
  - visualizzi sul video i nomi dei primi tre classificati, senza riordinare l'elenco.

Si rammenti che il primo classificato è quello che ha avuto il punteggio massimo, il secondo è quello che ha avuto il punteggio massimo inferiore al primo, e così via (occorre, in generale, cercare il massimo inferiore al massimo precedente).

Nota: si consiglia di utilizzare l'inserimento ordinato, visto nelle esercitazioni passate, in un vettore di struct.

- Esercizio 6. Realizzare una programma per la gestione degli esami di questo corso. I comandi che il programma deve gestire sono:

- I <Cognome\_Nome> <Matricola> <Voto>  
per inserire i dati di uno studente.
- C <Cognome\_Nome>  
per cancellare i dati dello studente con quel nome.
- C <Matricola>  
per cancellare i dati dello studente con quella matricola.
- V <Cognome\_Nome>  
per visualizzare i dati dello studente con quel nome.
- V <Matricola>  
per visualizzare i dati dello studente con quella matricola.
- P  
per stampare tutto il data-base.
- L <Nomefile>  
per leggere i dati (già ordinati) dal file con quel nome.
- S <Nomefile>  
per salvare i dati del data-base nel file con quel nome.

Utilizzare un vettore di struct ordinato in base al campo <Cognome\_Nome>.

NOTE: i comandi sono da fornire (e il programma deve accettarli) nella forma fissata dalle specifiche, senza inutili domande ridondanti. Ad esempio, per inserire i dati di uno studente:

*I Rossi Francesco Maria 123456 30*

Pertanto per i comandi C (cancellazione) e V (visualizzazione) il programma deve comprendere da solo (dal dato) se si è inserito un nome oppure una matricola.

Si ipotizzi che i `Cognome_Nome` non contenga cifre, che la `Matricola` sia solo numerica e che l'input sia corretto.