

Istruzioni

- Tempo disponibile: 120 minuti.
- Non è permesso l'uso di dispositivi elettronici (a parte il PC della propria postazione).
- Il programma sarà valutato per
 - Identificazione delle strutture dati e degli algoritmi appropriati alle specifiche
 - Corretta implementazione di strutture dati e algoritmi
 - Utilizzo efficiente delle risorse
 - Stile (chiarezza, utilizzo di costrutti appropriati, corretta strutturazione)
- I programmi non compilabili saranno valutati 0 punti.
- Fare l'upload di tutti i file che compongono il programma.
- Visual Studio Code è installato su tutte le postazioni. Si può comunque usare l'editor che si preferisce fra quelli installati.

Esercizio - Parte 1 (max 18 punti)

Un file binario contiene i dati degli esami sostenuti da uno studente. Ogni record contiene

- giorno dell'esame (`int`);
- mese dell'esame (`int`);
- anno dell'esame (`int`);
- nome dell'insegnamento (array di 50 `char` compreso il terminatore, senza spazi);
- voto conseguito (`int`).

Ad esempio, il file allegato `esami.dat` contiene i dati mostrati in Tabella 1. Gli esami non sono in ordine di data e possono comparire più esami relativi allo stesso insegnamento, anche di esito positivo (si suppone che i voti meno recenti per data siano stati rifiutati).

Si scriva un programma in linguaggio C, da compilare in un eseguibile di nome `libretto`, che mostri gli esami superati dallo studente. In particolare, il programma deve

- prendere come argomento della linea di comando il nome di un file binario del formato sopra indicato;

Giorno	Mese	Anno	Insegnamento	voto
30	6	2022	architettura-degli-elaboratori-e-laboratorio	0
3	2	2021	calcolo-delle-probabilita-e-statistica	3
21	5	2020	lingua-inglese	18
25	5	2021	fisica	23
23	6	2021	economia-e-gestione-aziendale	18
5	8	2021	lingua-inglese	28
18	9	2022	calcolo-delle-probabilita-e-statistica	24
3	9	2020	programmazione-e-laboratorio	1
25	1	2020	matematica-discreta	0
27	9	2020	matematica-discreta	14
27	1	2020	economia-e-gestione-aziendale	24
13	1	2022	fisica	0
7	9	2022	istituzioni-di-matematica	17
4	7	2020	fisica	26
22	12	2022	lingua-inglese	22
7	12	2020	lingua-inglese	5
30	7	2022	economia-e-gestione-aziendale	22
12	5	2020	programmazione-e-laboratorio	3
25	5	2021	fisica	25
8	4	2020	fisica	1
22	6	2020	calcolo-delle-probabilita-e-statistica	0
22	1	2021	istituzioni-di-matematica	15
2	2	2022	fisica	4
23	4	2021	fisica	12
8	5	2021	lingua-inglese	31
5	9	2021	programmazione-e-laboratorio	10
23	3	2021	programmazione-e-laboratorio	11
14	12	2021	lingua-inglese	4
24	9	2021	architettura-degli-elaboratori-e-laboratorio	18
7	12	2022	fisica	29

Tabella 1: Contenuto dell'allegato [esami.dat](#)

- creare una lista collegata contenente un elemento per ogni insegnamento; ogni elemento della lista dovrà contenere, oltre al nome dell'insegnamento, la data e il voto dell'esame *di esito positivo* (cioè con voto maggiore o uguale 18) e *più recente* relativo a quell'insegnamento;
- per ogni insegnamento, stampare a video una riga contenente il nome dell'insegnamento, la data di superamento e il voto;
- stampare infine la media aritmetica (non ponderata) dei voti.

Ad esempio, se `esami.dat` fosse il file allegato, l'invocazione `./libretto esami.dat` dovrebbe produrre un output simile al seguente:

```
lingua-inglese 22/12/2022 22
fisica 7/12/2022 29
economia-e-gestione-aziendale 30/7/2022 22
calcolo-delle-probabilita-e-statistica 18/9/2022 24
architettura-degli-elaboratori-e-laboratorio 24/9/2021 18
```

Media: 23.00

Ulteriori specifiche

- La lista collegata deve essere definita come tipo di dato astratto, cioè separando interfaccia e implementazione.
- Verificare la correttezza della linea di comando e la corretta apertura dei file; in caso di errore, stampare un messaggio e terminare l'esecuzione.
- Il programma deve essere costituito dai seguenti file:
 - `main.c` contenente (tra eventuali altre) la funzione `main`;
 - `listaEsami.c` con la definizione delle funzioni su liste (ed eventuali altre);
 - `listaEsami.h` con le definizioni dei tipi di dato e le dichiarazioni delle funzioni definite in `listaEsami.c` e utilizzate in `main.c`;
 - `Makefile` che permetta di costruire l'eseguibile con un singolo comando `make`.

Esercizio - Parte 2 (max 4 punti)

Fare in modo che il programma legga da un file di testo, dato come secondo argomento alla linea di comando, il numero di CFU assegnato a ogni insegnamento (es. 12 per `programmazione-e-laboratorio`) e che stampi la media ponderata oltre a quella non ponderata.