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 (elencati nella sezione "Ulteriori specifiche").

Esercizio - Parte 1 (max 18 punti)

Un file binario contiene i risultati delle due prove (pratica e teorica) di un esame di Programmazione e laboratorio.

Ogni risultato è rappresentato da un record formato da

- matricola del candidato (`int`);
- punteggio ottenuto (`int`);
- tipo della prova (`char`; vale `'P'` per la prova pratica e `'T'` per la prova teorica).

typedef struct {
int matricola;
int punteggio;
char tipo;} Record;

Ad esempio, il file binario allegato `prove.dat` contiene i dati nella tabella 1.

La prova pratica ha punteggio massimo 22 ed è superata con almeno 12 punti. La prova teorica ha punteggio massimo 11 ed è superata con almeno 6 punti. L'esame è superato se sono superate entrambe le prove; il voto è la somma dei punteggi (30 e lode se la somma è maggiore di 30).

Scrivere un programma in linguaggio C, da compilare in un eseguibile di nome `esami`, che

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

./esami prove.dat

- stampi a video una riga per ogni candidato, contenente la matricola e

- se lo studente non ha superato l'esame, l'espressione "non superato"
- se lo studente ha superato l'esame, il voto.

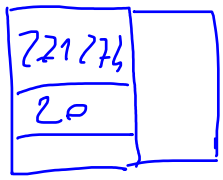
L'ordine dei candidati nell'output non importa.

Ad esempio, se `prove.dat` è il file allegato, l'invocazione

`./esami prove.dat`

deve produrre un output simile al seguente:

```
221274 30 e lode
225556 non superato
223441 26
211942 non superato
219394 non superato
219290 29
219524 19
229174 non superato
223969 non superato
218960 23
228857 30
221188 27
```



typedef struct {
int matricola;
int praticoi
int teoricoi
} Dato,

Per rappresentare i dati relativi agli esami è obbligatorio utilizzare una lista collegata. Si suggerisce di utilizzare un elemento della lista per ogni candidato, nel quale memorizzare matricola, punteggio della prova pratica e punteggio della prova teorica.

Dato != Record

Esercizio - Parte 2 (max 4 punti)

Stampare i candidati in ordine decrescente di punteggio e, a parità di punteggio, in ordine decrescente di numero di matricola.

l2 = insSort(l1).

Ulteriori specifiche

- La lista collegata deve essere implementata come tipo di dato astratto (in modo cioè che il programma principale acceda alla lista solo attraverso le funzioni definite nell'interfaccia della lista).
- 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`.

Matricola	Punteggio	Prova
221274	11	T
225556	8	P
223441	11	T
211942	10	P
219394	5	T
219290	9	T
219524	7	T
219394	20	P
229174	9	P
223969	7	T
218960	13	P
228857	20	P
225556	6	T
219290	20	P
218960	10	T
221274	22	P
221188	11	T
223969	9	P
229174	5	T
228857	10	T
219524	12	P
211942	7	T
223441	15	P
221188	16	P

Tabella 1: Contenuto del file allegato `prove.dat`