

Esame di Programmazione

13 gennaio 2020

Corso di Laurea in Ingegneria e Scienze Informatiche

A.A. 2019/2020

Tempo a disposizione: 1,5 h

I compiti NON adeguatamente commentati NON saranno corretti. Usare preferibilmente penna blu o verde per i commenti, penna nera per il codice.

Utilizzare i NOMI di campi, strutture e funzioni come specificati nel testo

La segreteria studenti del Corso di Studi in Ingegneria e Scienze Informatiche deve gestire le informazioni relative agli studenti iscritti. Si supponga per semplificare che il Corso di Studi sia composto di soli 5 esami. In particolare, il corso è rappresentato da una struttura che mantenere: la *descrizione* del corso, l'*id* del corso, un puntatore *lp* (listapositivi) a una lista di strutture collegate di tipo *struct_studente*, un puntatore *ln* (listanegativi) a una lista di strutture collegate di tipo *struct_studente*, un puntatore *lt* (listatutti) a una lista di strutture collegate di tipo *struct_studente*.

La struttura *struct_studente* deve contenere i campi: nome, cognome, matricola, voti (vettore statico di 5 elementi), media.

Nella lista puntata da *lp* sono presenti i soli studenti la cui media dei voti è $\geq 18/30$, nella lista puntata da *ln* sono presenti i soli studenti la cui media dei voti è $< 18/30$. Il puntatore *lt* è inizialmente posto a NULL.

Definire le strutture dati necessarie a memorizzare le informazioni descritte. Inoltre, predisporre il codice essenziale del *main()* per poter dichiarare queste strutture. (4 punti)

Implementare la funzione *inserisci_coda_lista* che, presi come parametro tutti i dati di uno studente, lo inserisca in coda alla giusta lista. (4 punti)

Implementare la funzione *modifica_voti* che, presi come parametro la matricola di uno studente e i suoi nuovi voti vada a modificare la struttura corrispondente. Questa funzione deve anche eventualmente spostare lo studente da una lista all'altra se la nuova media lo richiede, inserendolo in coda. (6 punti)

Implementare la funzione *elimina_studente* che, preso come parametro la matricola di uno studente, lo elimini dalla lista a cui appartiene. (4 punti)

Implementare la funzione *inverti_positivi* che, preso come parametro il puntatore alla listapositivi ne inverta l'ordine (così che l'ultimo elemento diventi il primo, il penultimo il secondo, etc.). (6 punti)

Implementare infine la funzione *crea_unica_lista* che, presi come parametro i due puntatori *lp* e *ln*, costruisca una sola lista collegata al puntatore *lt* e ponga a NULL i due puntatori *lp* e *ln*. (6 punti).