



Esercitazione di laboratorio n. 6

(Le soluzioni verranno valutate in Laboratorio. Solo se caricate nella Sezione Elaborati del Portale entro e non oltre il 18/11/2014 concorreranno all'assegnazione dei punti supplementari)

Esercizio n. 1: liste, pile, code, code a priorità

Si definisce **sequenza lineare** un insieme finito di elementi consecutivi. A ciascun elemento è associato univocamente un indice

$$a_0, a_1, \dots, a_i, \dots, a_{n-1}$$

e su ogni coppia di elementi è definita una relazione predecessore/successore:

$$a_{i+1} = \text{succ}(a_i) \qquad a_i = \text{pred}(a_{i+1})$$

L'accesso a ogni elemento può essere:

- diretto, cioè dato l'indice i l'elemento viene letto con costo $O(1)$
- sequenziale, cioè dato l'indice i , il costo di leggere l'elemento di quell'indice è $O(n)$.

Si chiama **vettore** una sequenza lineare ad accesso diretto, **lista** una sequenza lineare ad accesso sequenziale.

Gli elementi delle liste sono in generale formati da più campi. Se uno di questi, detto **chiave di ordinamento**, ammette una relazione d'ordine totale, la lista può essere:

- **ordinata**
- **non ordinata.**

Le operazioni fondamentali sulle liste sono:

- **ricerca** di un elemento il cui campo **chiave di ricerca** è uguale a una chiave data
- **inserzione** di un elemento che può avvenire
 - **in testa** alla lista non ordinata
 - **in coda** alla lista non ordinata
 - **nella posizione** tale da garantire l'invarianza della proprietà di ordinamento per una lista ordinata
- **estrazione** di un elemento:
 - che si trova **in testa** alla lista non ordinata
 - che ha **un campo** con contenuto uguale a quello di una chiave di cancellazione (tale operazione richiede solitamente una ricerca preventiva dell'elemento da cancellare).

Si dice **stack** o **pila** una lista non ordinata con inserzione in testa ed estrazione dalla testa che realizza quindi una politica **LIFO** (last-in first-out).

Si dice **coda** o **queue** una lista non ordinata con inserzione in coda ed estrazione dalla testa che realizza quindi una politica **FIFO** (first-in first-out).

Si dice **coda a priorità** o **priority queue** una struttura dati dove l'inserzione garantisce che, estraendo dalla testa, si ottenga il dato a priorità massima (o minima). Tra le possibili implementazioni che realizzano questa politica si considera ora **solo** quella basata su liste ordinate.



Si scriva un programma C che, tramite menu, offra all'utente di gestire una lista (non ordinata), una pila, una coda o una coda a priorità di elementi del tipo di seguito specificato di un elemento (la scelta va fatta come prima operazione (appena il programma viene avviato)).

Ogni elemento contiene le seguenti informazioni:

- campi nome, cognome e matricola dello studente (possibili chiavi di ricerca, estrazione e ordinamento), ciascuno rappresentato da una stringa, priva di spazi, di massimo 35 caratteri alfabetici (maiuscoli o minuscoli)
- campo carriera, sotto-struttura che consiste di un intero che rappresenta i crediti superati e in un float che rappresenta la media (campo identificato come priorità per la coda prioritaria; a parità di media è indifferente l'ordine).

Per la lista non ordinata siano possibili le seguenti operazioni:

- visualizzazione della cardinalità della lista
- inserzione in testa di un elemento
- inserzione in coda di un elemento
- ricerca di un elemento con chiave di ricerca alternativamente pari a ciascuno dei campi dell'elemento stesso. In caso di match multipli, si ritornino tutti gli elementi
- estrazione di un elemento con chiave di cancellazione alternativamente pari a ciascuno dei campi dell'elemento stesso. In caso di match multipli, si cancellino tutti gli elementi
- visualizzazione del contenuto della lista
- distruzione della lista.

Per la pila siano possibili le seguenti operazioni:

- visualizzazione della cardinalità della pila
- inserzione in testa di un elemento
- estrazione dell'elemento di testa
- visualizzazione del contenuto della pila
- distruzione della pila.

Per la coda siano possibili le seguenti operazioni:

- visualizzazione della cardinalità della coda
- inserzione in coda di un elemento
- estrazione dell'elemento di testa
- visualizzazione del contenuto della coda
- distruzione della coda.

Per la coda a priorità siano possibili le seguenti operazioni:

- visualizzazione della cardinalità della coda a priorità
- inserzione di un elemento nella posizione corretta in base a un campo identificato come priorità
- estrazione dell'elemento a priorità massima
- visualizzazione del contenuto della coda a priorità
- distruzione della coda a priorità.

Non è lecito assumere alcun limite superiore al numero di elementi.

Si richiede un'implementazione modulare con un client e quattro implementazioni distinte (ognuna con file .c e file .h) per lista non ordinata, pila, coda e coda prioritaria.