



Esercitazione di laboratorio n. 9

Laboratorio valutato nei giorni 02/12/2015 (SQ1 e SQ2) e 04/12/2015 (SQ3 e SQ4)
Caricamento nella Sezione Elaborati del Portale entro e non oltre le 23:59 del 08/12/2015

Esercizio n. 1: Sportelli postali

Si realizzi un programma C che, attraverso un'apposita interfaccia utente, permetta di gestire più strutture dati di tipo coda a priorità (PQ). Le N code a priorità siano organizzate in un vettore di code. N sia passato sulla riga di comandi. Ogni coda sia identificata mediante un indice intero nell'intervallo 0 ... N-1.

Si richiede di implementare le code a priorità come ADT di I categoria mediante liste concatenate non ordinate con inserimento in coda, estrazione per priorità massima e, a parità di priorità, dell'Item inserito meno recentemente.

Il tipo Item sia caratterizzato da una priorità di servizio (intero), una tipologia di servizio (stringa di massimo 30 caratteri) e un codice cliente (stringa di massimo 10 caratteri). Il tipo Item deve essere implementato come quasi ADT, quindi con la struct definita in item.h, dunque visibile.

Le operazioni permesse devono essere quelle di:

- stampa del numero di dati in una coda e dei relativi dettagli
- stampa di tutti i dettagli di tutte le code
- inserimento di un nuovo elemento in una coda
 - liberamente scelta dall'utente tramite il client
 - automaticamente scelta come quella a coda con attualmente meno elementi
- estrazione di un elemento da una coda
- salvataggio di dati su file
- caricamento di dati da file.

In questo esercizio, il programma deve essere realizzato su tre moduli distinti:

- l'interfaccia utente (il client)
- un modulo PQ con le funzioni per la gestione della coda
- un modulo Item con le funzioni per la gestione dei singoli dati.

Si noti che l'interfaccia utente è "client" sia di PQ che di Item, mentre PQ è client di "Item".

Esercizio n.2: Spellcasting

Riferimento - Ricorsione e problem-solving (Sez.7.1)

Un gioco di ruolo richiede che ogni giocatore, perché una sua magia abbia successo, estragga alla cieca sei rune colorate da un sacchetto. Il sacchetto contiene esattamente sei rune: una rossa, due verdi e tre blu. Una magia ha successo se la runa rossa non appare tra le prime due rune estratte e se non viene estratta per ultima.

Si scriva un programma in C in grado di generare tutte le sequenze di estrazioni favorevoli al giocatore.

Esercizio n.3: Commensali felici

Riferimento - Ricorsione e problem-solving (Sez.7.1)



Un organizzatore di eventi deve distribuire N ospiti in K tavoli per un banchetto. L'organizzatore sa che ogni tavolo può accogliere al massimo M ospiti e che nella sala c'è posto per tutti.

Per ogni coppia di ospiti, inoltre, è noto un indice di reciproco gradimento, espresso mediante un intero nell'intervallo $[-5 \dots 5]$.

La sommatoria del gradimento reciproco per tutte le coppie di ospiti di un tavolo definisce il livello di umore per il tavolo stesso.

L'obiettivo dell'organizzatore è assegnare un tavolo ad ogni ospite massimizzando l'umore globale della sala, valutato come la media dell'umore tra tutti i tavoli.

Si scriva un programma in linguaggio C che, una volta acquisite le informazioni necessarie dal file `banchetto.txt`, produca la soluzione cercata.

Il file `banchetto.txt` è organizzato come segue:

- sulla prima riga è riportata la terna di interi $N \ K \ M$
- seguono N righe composte da N interi ciascuna a rappresentare l'indice di reciproco gradimento tra due ospiti. La matrice risultante è ovviamente simmetrica.