



Esercitazione di laboratorio n. 6

Laboratorio valutato nei giorni 11/11/2015 (SQ1 e SQ2) e 13/11/2015 (SQ3 e SQ4)
Caricamento nella Sezione Elaborati del Portale entro e non oltre le 23:59 del 17/11/2015

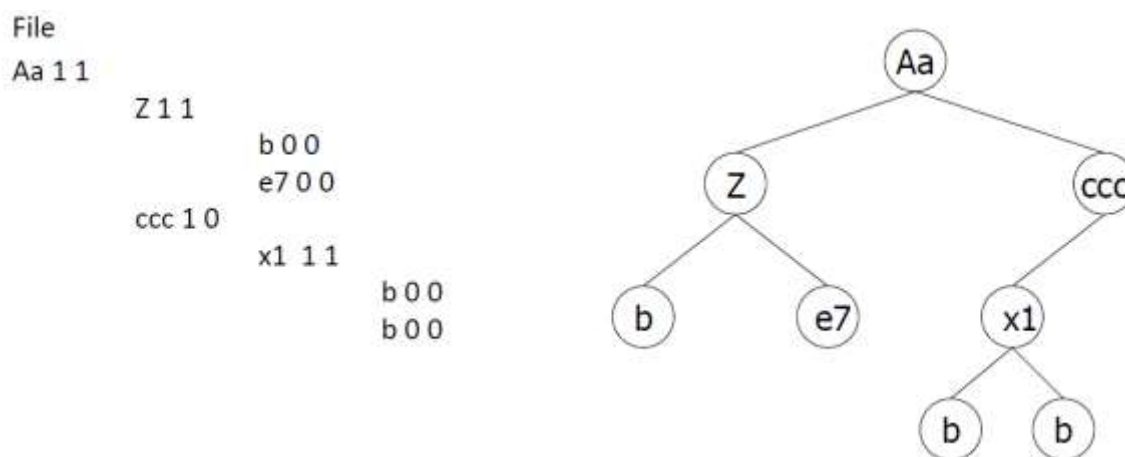
Esercizio n. 1: Operazioni su alberi

Si scriva un programma in linguaggio C adatto ad eseguire le seguenti operazioni su alberi binari:

- conteggio del numero di nodi con esattamente due figli
- stampa di tutti i nodi che si trovino tra due specifici livelli L_1 e L_2 dell'albero, elencando prima tutti i nodi che si trovino al livello meno profondo (più vicino alla radice), e proseguendo via via per livelli più profondi. Si ricorda che i nodi allo stesso livello sono quelli aventi la stessa profondità
- verifica di isomorfismo di due alberi. Due alberi si dicono isomorfi se sono strutturalmente identici, cioè se contengono le stesse chiavi nelle medesime posizioni.

Per svolgere questo esercizio, utilizzare la funzione di lettura messa a disposizione nel sorgente `albero.c`.

Il file di ingresso accettato dalla libreria è composto da una serie di terne, in ragione di una per riga, a rappresentare la chiave (stringa di al massimo 16 caratteri sovralllocata staticamente) di ogni nodo e una coppia di valori binari a indicare la presenza (assenza) dei figli destro e sinistro per uno specifico nodo. Per maggiore leggibilità, i due file di esempio sono caratterizzati da un'indentazione via via crescente all'aumentare della profondità a cui un dato nodo si trovi. Esempio:



Esercizio n. 2: Gruppo di acquisto

Si progetti una struttura dati adatta a memorizzare le informazioni relative ad una sequenza di liste della spesa associate ad un gruppo di acquisto.

Le informazioni relative ai prodotti acquistabili sono riportate in un file testuale (`prodotti.txt`), il quale deve essere a sua volta caricato in memoria. Il file riporta il numero di prodotto diversi disponibili e per ogni prodotto il nome (stringa di massimo 30 caratteri senza spazi) e il prezzo per unità di prodotto (float).

Un secondo file testuale (`liste.txt`) contiene la descrizione di un insieme di liste della spesa, che fanno riferimento ai prodotti di cui sopra. La prima riga del file riporta il numero totale di



liste contenute. Nelle righe successive, per ogni lista della spesa è poi specificato il numero P di prodotti che la compongono, seguito da P righe riportanti il nome e la quantità di ogni prodotto incluso.

Si scriva un programma in linguaggio C che permetta di portare in memoria tutte le informazioni di cui sopra, e renda disponibili le seguenti operazioni:

- stampa a video dei dettagli di tutte le liste
- calcolo del costo per ogni lista della spesa
- calcolo del costo totale per tutti gli acquisti del gruppo
- aggiunta di una nuova lista della spesa
- aggiunta di un prodotto a una certa lista della spesa

Si devono prevedere due strutture dati, entrambe gestite mediante una struct contenitore (wrapper):

- tabella dei prodotti (letta dal primo file). La tabella è un vettore di prodotti. La struct contenitore contiene il puntatore al vettore dinamico e la sua dimensione.
- tabella delle liste della spesa (letta dal secondo file). La tabella è un vettore di liste della spesa. La struct contenitore contiene il puntatore al vettore, la dimensione allocata e quella effettivamente utilizzata. Ogni lista della spesa è un vettore dinamico di struct contenenti puntatori a prodotti (dalla prima tabella) e relativa quantità. Anche per queste liste è opportuno utilizzare una struct contenitore.

Sia le singole liste della spesa che il vettore delle liste debbono essere ri-allocabili per contenere nuovi elementi. La figura sotto-riportata rappresenta la struttura richiesta.

