



Esercitazione di laboratorio n. 4

(Caricamento sul portale entro le 23.59 del 21/11/2016 di entrambi gli esercizi)

Esercizio n. 1: Catalogo di prodotti

Competenze: Vettori di `struct`, strutture dati dinamiche, vettori dinamici, riallocazione dinamica (Puntatori e strutture dati dinamiche 2.5.5, 3.3.3, 3.2.3)

Sia dato un file di testo contenente un catalogo di prodotti (`catalogo.txt`), organizzato come segue:

- sulla prima riga è presente un intero `N` rappresentante il numero di prodotti
- sulle `N` righe successive sono presenti tre stringhe quali il codice del prodotto, il nome del prodotto e la categoria merceologica seguite da un valore reale, il prezzo, e un intero a rappresentare le unità disponibili. La riga è terminata da una data che indica l'ultima volta che il record del prodotto in questione è stato aggiornato
- il codice è nella forma `codXXXXXX`, dove `X` rappresenta una cifra nell'intervallo 0-9
- il nome e la categoria di ogni prodotto sono rappresentati da una stringa, priva di spazi, di massimo 25 caratteri alfabetici (maiuscoli o minuscoli)
- le date sono riportate nel formato `gg/mm/aaaa` (es: 01/10/2016)
- tutti i campi sono separati da uno o più spazi.

Si scriva un programma C tale per cui:

- i contenuti del catalogo sono memorizzati in un vettore di strutture allocato dinamicamente della dimensione opportuna
- i dettagli di ogni prodotto sono memorizzati in una apposita `struct` in cui tutte le stringhe repute necessarie siano allocate dinamicamente.

Una volta memorizzate le informazioni contenute nel file, il programma deve rendere disponibili le seguenti funzioni:

- stampa, a scelta se a video o su file, dei contenuti del catalogo
- ordinamento del vettore per prezzo, ascendente o discendente
- ordinamento del vettore per codice prodotto
- ordinamento del vettore per nome prodotto
- stampa a video dei prodotti, divisi per categoria merceologica.

Le seguenti funzioni sono facoltative:

- aggiornamento della quantità di unità disponibili
- ricerca di un prodotto per codice
- ricerca di un prodotto per nome (anche parziale).

Per quanto riguarda le ricerche, si richiede che siano implementate sia una funzione di ricerca dicotomica sia una funzione di ricerca lineare.

Il programma deve mantenere uno stato relativo all'ordinamento corrente della base dati (ossia, su quale chiave sia attualmente ordinato) e usare di conseguenza la funzione di ricerca più opportuna.



Esercizio n. 2: Allocazione di matrici

Competenze: Strutture dati dinamiche, matrici dinamiche create da funzioni (Puntatori e strutture dati dinamiche 3.3.3)

Un file di testo contiene una matrice di interi con il seguente formato:

- la prima riga del file specifica le dimensioni della matrice (numero di righe nr e numero di colonne nc).
- ciascuna delle nr righe successive contiene gli nc valori corrispondenti a una riga della matrice, separati da uno o più spazi.

Si scriva un programma che:

- allochi dinamicamente la matrice ed effettui la lettura del file. La funzione di allocazione della matrice può:
 - far uso del valore di ritorno per restituire il puntatore alla matrice al main (`int **malloc2dR(...)`, Puntatori e strutture dati pag. 76)
 - restituire il puntatore alla matrice tra i parametri passati per riferimento (`void malloc2dP(int***, ...)`; Puntatori e strutture dati pag. 76)
- una volta acquisita la matrice, generi due vettori (dinamici) di vettori (dinamici) per memorizzare i contenuti della matrice “per diagonal” e “per anti-diagonal”. In entrambi i casi, il vettore principale sia allocato della dimensione pari al numero di (anti)diagonali della matrice stessa. Ogni singolo vettore secondario sia allocato della dimensione esatta per contenere tutti (e soli) gli elementi di una specifica (anti)diagonale della matrice. Si faccia riferimento alla figura seguente per un esempio della struttura dati richiesta.

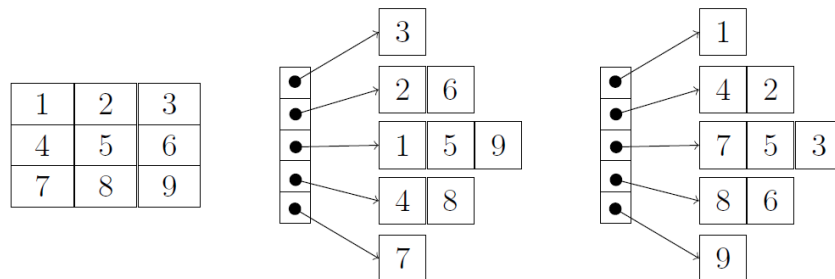


Figura 1: Una matrice di esempio e la sua organizzazione per diagonal e anti-diagonal.