



Esercitazione di laboratorio n. 6

Esercizio n.1: Playlist

Competenze: esplorazione dello spazio delle soluzioni con i modelli del Calcolo Combinatorio (Ricorsione e problem-solving: 3.2, 3.3)

Un gruppo di amici vuole preparare una playlist per un viaggio. Per accontentare tutti, ogni amico ha diritto a proporre fino a cinque canzoni tra cui scegliere. La playlist viene creata scegliendo tante canzoni quanti sono gli amici, e per ognuno di essi scegliendo una ed una sola canzone tra quelle da lui proposte. Tutte le canzoni sono distinte.

Le informazioni sulle canzoni proposte sono memorizzate in un file testuale (`brani.txt`) organizzato come segue:

- sulla prima riga appare il numero A di amici
- seguono A blocchi di righe, uno per ogni i -esimo amico, tali per cui
 - sulla prima riga del blocco appare il numero C_i di canzoni proposte dall'amico
 - seguono C_i stringhe, in ragione di una per riga, riportanti i titoli dei brani proposti

Si assuma che tutti i titoli delle canzoni non contengano spazi e siano lunghi al più 255 caratteri.

Si scriva un programma in C che, letto il file di input, generi tutte le playlist possibili con le regole di cui sopra.

Nota: i contenuti del file di input di esempio in allegato sono indentati per rendere più facile distinguere le porzioni dedicate ad ogni amico.

Si consiglia di individuare preventivamente a quale modello del Calcolo Combinatorio, tra quelli visti a lezione, faccia riferimento il problema e in seguito di adattare il codice presentato in aula.

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(...);`
- restituire il puntatore alla matrice tra i parametri passati per riferimento
`void malloc2dP(int**, ...);`

Una volta acquisita la matrice, il programma deve invocare una funzione

`void separa(int **mat, int nr, int nc, ...)`

che:

- interpreti la matrice come se fosse una scacchiera (su ogni riga e su ogni colonna celle bianche e nere alternate)



- separi i dati delle caselle nere da quelli delle caselle bianche, copiandoli (in ordine arbitrario) in due vettori dinamici, che vanno ritornati al programma chiamante
- i due vettori di interi, di lunghezza opportuna e tale da poter contenere gli elementi delle "celle bianche" e delle "celle nere" separatamente, vanno allocati dinamicamente
- i due vettori e i loro contenuti devono essere visibili al chiamante della funzione `separa()` (andranno quindi dichiarati in modo opportuno, al posto dei puntini ...), che li stampa e li dealloca.

Esercizio n. 3: Azienda di trasporti - multiordinamento

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

Si ripeta l'esercizio 5 del laboratorio 4 utilizzando vettori allocati dinamicamente. In particolare, i vettori vanno allocati in base al numero di dati effettivamente presenti nel file acquisito.

Oltre ai comandi già previsti, si chiede di aggiungerne uno di acquisizione (lettura) di un nuovo file (dato il nome). Si prevede quindi non solo di acquisire una volta sola i dati da file, all'inizio dell'esecuzione, ma di poterlo fare ogni volta che l'utente lo richieda. Il comando di lettura, in generale, prevede l'acquisizione di nuovi dati da un file diverso dal precedente. Occorre quindi liberare (utilizzando la funzione `free`) i vettori precedentemente allocati, e allocarne di nuovi per i nuovi dati da leggere. Si noti che il compito di acquisizione dei dati può essere svolto da una sola funzione, chiamata sia (una prima volta) all'avvio del programma che ad ogni attivazione del comando di lettura.

Valutazione: tutti gli esercizi saranno oggetto di valutazione

Scadenza: caricamento di quanto valutato: entro le 23:59 del 24/11/2020.