



## ***Esercitazione di laboratorio n. 7***

(Caricamento sul portale entro le 23.59 del 12/12/2016 dell'esercizio 3 e di almeno uno tra gli esercizi 1 e 2)

**Nota:** per gli esercizi n.1 e n.2 si consiglia di individuare preventivamente a quale modello ricorsivo, tra quelli visti a lezione, essi facciano riferimento e in seguito fare riferimento al codice presentato in aula opportunamente adattato.

### **Esercizio n. 1: Cover Set**

Sia  $U$  l'insieme degli interi compresi 1 e 9:  $U = \{1, 2, 3, 4, 5, 6, 7, 8, 9\}$ . Sia  $S$  una matrice  $n \times 10$  di interi che su ciascuna delle  $n$  righe contiene un sottoinsieme di  $U$  terminato dal valore 0. La matrice viene ricevuta da una funzione come parametro formale con numero di righe ignoto. Il numero  $n$  di righe compare come ulteriore parametro.

Dato un intero  $k$ , ulteriore parametro della funzione, si scriva una funzione ricorsiva in C che visualizzi, se esiste, una collezione di  $k$  sottoinsiemi la cui unione sia  $U$ . A fronte di più soluzioni possibili, è sufficiente individuarne una qualsiasi.

La funzione abbia il seguente prototipo:

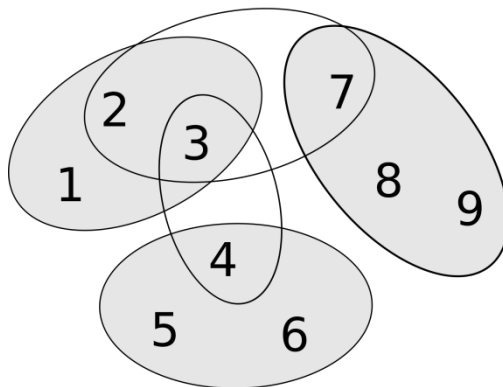
```
void cover(int S[][10], int n, int k);
```

Esempio:

Si considerino i seguenti  $n=5$  sottoinsiemi:

$$S_0 = \{1, 2, 3\}, S_1 = \{2, 3, 7\}, S_2 = \{7, 8, 9\}, S_3 = \{3, 4\}, S_4 = \{4, 5, 6\}$$

A fronte di  $k=3$ , una soluzione possibile, in cui gli insiemi scelti sono evidenziati in grigio in figura, sarebbe  $\{S_0, S_2, S_4\}$ .



### **Esercizio n. 2: Playlist**

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.

### **Esercizio n.3: Ricettario**

Si realizzi un programma C che, attraverso un'apposita interfaccia utente, permetta di costruire una struttura dati adatta a valutare ricette, in base agli ingredienti necessari.

Una ricetta è caratterizzata da:

- nome (stringa di massimo 255 caratteri, senza spazi)
- tempo di preparazione, espresso in minuti
- elenco di ingredienti, ognuno in una certa quantità espressa in grammi.

Ogni ingrediente è caratterizzato da:

- nome (stringa di massimo 255 caratteri, senza spazi)
- costo al chilo (numero reale positivo)
- apporto calorico al grammo (numero reale positivo)

Le informazioni complete sugli ingredienti disponibili sono memorizzate in un file (`ingredienti.txt`), organizzato come segue:

- sulla prima riga è presente il numero  $I$  di ingredienti
- sulle  $I$  righe successive sono riportati i dettagli di ogni ingrediente, in ragione di una terna `<nome> <costo> <calorie>` per riga

Le informazioni relative a una ricetta sono acquisite da standard input (il formato è libero). Il programma, dopo aver letto e acquisito in una opportuna tabella gli ingredienti disponibili, continua mediante un opportuno menu (si preveda un comando per "fine"), ad acquisire nuove ricette. Data una ricetta, si cercano nella tabella degli ingredienti le informazioni sugli ingredienti richiesti, in modo da poter valutare costo e apporto calorico della ricetta. I dati calcolati vanno visualizzati su standard output.

In questo esercizio il programma deve essere realizzato su due moduli distinti (con due file `.c` distinti e un `.h`):

- il main, con interfaccia utente e menu (il client)
- un modulo con le funzioni per la gestione degli ingredienti.

Il modulo per gli ingredienti deve fornire devono offrire le funzionalità di:

- acquisizione da file della tabella degli ingredienti. La tabella deve essere realizzata come lista concatenata di ingredienti, avente una struct wrapper (involucro) contenente puntatore a head e tail. La lista va creata con inserimenti in coda.
- ricerca di un ingrediente

Il main, oltre a coordinare l'acquisizione degli ingredienti (chiamata a funzione), deve iterare su acquisizioni di ricette e calcolo, mediante ricerca degli ingredienti, di costo e contenuto calorico.