Esercitazione di laboratorio n. 3

(Caricamento sul portale entro le 23.59 del 03/11/2017dell'es. 2 e di almeno uno tra gli es. 1 e 3)

Esercizio n. 1: Individuazione di regioni

Competenze: lettura/scrittura di file, manipolazioni di matrici;
Categoria: problemi di verifica e selezione (Dal problema al programma: 4.5)

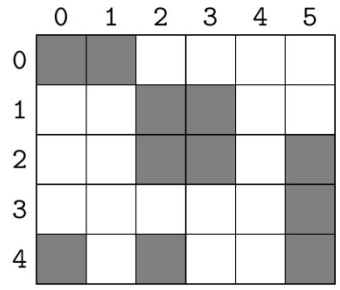
Un file di testo contiene una matrice di interi (0 o 1) con il seguente formato:

- la prima riga del file specifica le dimensioni reali della matrice (numero di righe `nr` e numero di colonne `nc`). Si assuma che entrambi i valori siano al più pari a 50
- ciascuna delle `nr` righe successive contiene gli `nc` valori corrispondenti a una riga della matrice, separati da uno o più spazi
- ogni cella può contenere solamente il valore 0 (associato al colore bianco) o il valore 1 (associato al colore nero)
- le celle nere sono organizzate in modo da formare regioni rettangolari (ogni regione nera è circondata da una cornice di celle bianche, oppure da bordo/i della matrice). A tal fine, si consideri che l'adiacenza delle celle è considerata solo lungo i quattro punti cardinali principali (Nord, Sud, Ovest, Est), non in diagonale.

Si scriva un programma C che:

- legga la matrice dal file di ingresso (il file non contiene errori, quindi ci sono solo rettangoli neri che rispettano i vincoli)
- individui le regioni nere
- per ogni regione produca in output le coordinate dell'estremo superiore sinistro, oltre che i valori di base e altezza e dell'area (espressa come numero di celle che la compongono).

Esempio:	Mappa corrispondente:

`5 6`
`1 1 0 0 0 0`
`0 0 1 1 0 0`
`0 0 1 1 0 1`
`0 0 0 0 0 1`
`1 0 1 0 0 1`
	

Output del programma:

```
Regione 1: estr. sup. SX = <0,0> b = 2, h = 1, Area = 2
Regione 2: estr. sup. SX = <1,2> b = 2, h = 2, Area = 4
Regione 3: estr. sup. SX = <2,5> b = 1, h = 3, Area = 3
Regione 4: estr. sup. SX = <4,0> b = 1, h = 1, Area = 1
Regione 5: estr. sup. SX = <4,2> b = 1, h = 1, Area = 1
```

Esercizio n. 2: Voli

Competenze: elaborazione di testi, ricerca in tabelle di nomi/stringhe, tipi enumerativi
Categoria: problemi di elaborazione testi mediante stringhe (Dal problema al programma: 4.4.3) e problemi di selezione (Dal problema al programma: 4.5.2)

Un azienda di trasporti aerei traccia i propri voli in un file di log, file testuale di nome `voli.txt`, contenente, nella prima riga, intero positivo che indica il numero di successive righe del file stesso (al più 1000), nelle righe successive le informazioni sui voli, uno per riga, con formato:

```
<codice_volo><partenza><destinazione><data_e_ora><vettore>
```

Tutte le stringhe sono lunghe al massimo 30 caratteri.

Si scriva un programma C in grado di rispondere alle seguenti interrogazioni:
1. elencare tutti i voli partiti in un certo intervallo di date
2. elencare tutti i voli partiti da una certa località
3. elencare tutti i voli diretti verso una certa destinazione
4. elencare tutte le località da cui sia decollato almeno un volo facente uso di un certo vettore.

Le interrogazioni di cui sopra siano gestite mediante menu di comandi (si veda il paragrafo 4.4.1, Dal problema al programma). Ogni comando consiste di una parola tra "date", "origine", "destinazione", "vettore" e "fine", eventualmente seguita sulla stessa riga da altre informazioni, ad esempio 2 date per date, una località di partenza per origine, etc. Si utilizzi la strategia di codifica dei comandi mediante tipo enum `comando_e`, contenente i simboli `r_date`, `r_origine`, `r_destinazione`, `r_vettore`, `r_fine`, che consente menu basati su `switch-case`.

Si consiglia di:
- realizzare una funzione `leggiComando` che, acquisito in modo opportuno il comando, ritorni il corrispondente valore di tipo `comando_e`
- realizzare una funzione `selezionaDati` che, ricevuti come parametri la tabella, la dimensione della tabella e il tipo di comando, gestisca mediante menu l'acquisizione delle informazioni aggiuntive necessarie per quel comando e la chiamata di un'opportuna funzione di selezione e stampa dei dati selezionati.

Esempio:
Dato il seguente file "voli.txt"

```
8
AZ007 TRN MPX 2017/09/01:12:00:00 B737
AZ189 MPX CIA 2017/09/05:06:15:00 B747
AZ293 JFK MPX 2017/10/02:11:30:00 A320
AI199 DEN LAX 2017/09/12:09:15:00 A330
AI912 HND SYD 2017/09/21:09:40:00 B737
AZ021 MPX JFK 2017/10/05:10:37:00 A320
AZ283 TRN CIA 2017/08/01:01:10:00 B737
AI222 MPX LAX 2017/10/31:02:19:00 A320
```

I risultati, a fronte dei parametri di esempio proposti, sarebbero:

1. Voli `2017/09/012017/09/30`
   `AZ007`
   `AZ189`
   `AI199`
   `AI912`
2. Origine `MPX`
   `AZ189`
   `AZ201`
   `AI222`
3. Destinazione `LAX`
   `AI199`
   `AI222`
4. Vettore `B737`
   `TRN`
   `HND`
   `TRN`

Suggerimento:
Supponendo di aver definito `MAXN` come `1000` e dichiarato un vettore di `structTabella`, di dimensione `MAXN`, in grado di contenere i dati presenti su file, la tabella del log dei voli va acquisita mediante una funzione (`leggiTabella`), che ne ritorna il numero di dati effettivamente letti come valore di ritorno. La funzione deve poter essere chiamata con un'istruzione del tipo:

```
nDati = leggiTabella(Tabella, MAXN);
```

Esercizio n. 3: Cammino nel labirinto

Competenze: lettura/scrittura di file, manipolazioni di matrici;
Categoria: problemi di verifica (Dal problema al programma: 4.5)

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

- la prima riga del file specifica le dimensioni reali della matrice (numero di righe `nr` e numero di colonne `nc`). Si assuma che entrambi i valori siano comunque al più pari a 50
- ciascuna delle `nr` righe successive contiene gli `nc` valori corrispondenti a una riga della matrice. Se il valore è '-', la cella rappresenta una zona percorribile. Se il valore è 'X', la cella rappresenta un muro. Ingresso e uscita dal labirinto sono indicate, rispettivamente, con 'I' e 'U'.

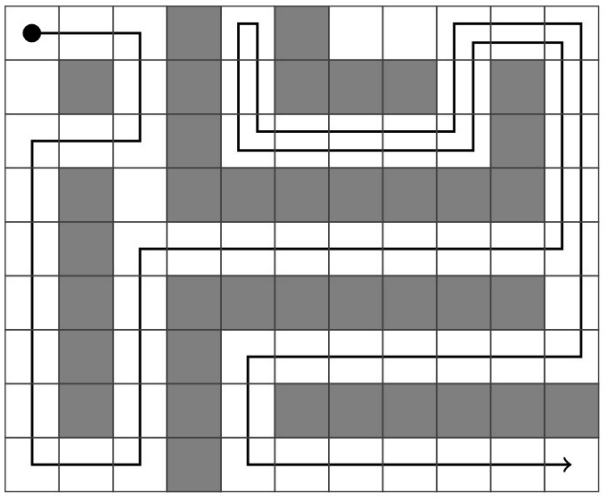
Si scriva un programma C che:

- legga tale matrice dal file di ingresso (nome ricevuto come `argv[1]`)
- acquisisca da file (nome ricevuto come `argv[2]`) una sequenza di movimenti nella forma di coppie di interi, positivi negativi o nulli,<±H, ±V>, a rappresentare lo spostamento verso destra (+) o sinistra (-) in orizzontale e verso l'alto (+) o il basso (-) in verticale
  - per semplicità si assuma di non ammettere spostamenti in diagonale
  - (facoltativo) si ammettano anche spostamenti in diagonale, sempre prestando attenzione alla presenza di muri lungo il cammino
- verifichi se si tratta effettivamente di un cammino, cioè se connette o meno le celle di ingresso e di uscita passando per celle permesse e, in caso affermativo:

- ne calcoli la lunghezza
- verifichi se il cammino è semplice.

Suggerimento:
Si consiglia di effettuare la verifica del cammino, mentre se ne fa l'acquisizione da tastiera.

Esempio:

Labirinto su file	Rappresentazione grafica con cammino
9 11 I--X-X----- -X-X-XXX-X- ---X-----X- -X-XXXXXXX- -X--------- -X-XXXXXXX- -X-X------- -X-X-XXXXXX ---X------U	

Il cammino, non semplice di lunghezza 70, rappresentato nell'immagine sarebbe associato alla seguente sequenza di spostamenti:

```
+2 +0
+0 -2
-2 +0
+0 -6
+2 +0
+0 +4
+8 +0
+0 +4
-2 +0
+0 -2
-4 +0
+0 +2
+0 -2
+4 +0
+0 +2
+2 +0
+0 -6
-6 +0
+0 -2
+6 +0
```