

TECNICHE DI PROGRAMMAZIONE, A.A. 2021/2022

Esercitazione di Laboratorio 5

Valutazione: **l'esercizio 2** sarà oggetto di valutazione.

Scadenza: caricamento di quanto valutato - entro le 23:59 del 13/5/2022: andranno caricati insieme i laboratori 4, 5 e 6.

Obiettivi

- Risolvere problemi numerici, di elaborazione testi e di verifica/selezione, iterativi, utilizzando vettori e matrici (*Dal problema al programma: Cap. 4*),

Contenuti tecnici

- Basi di Input Output
- Utilizzo di funzioni
- Costrutti condizionali e iterativi
- Manipolazioni elementari di vettori e matrici (di int e float)

Da risolvere durante il laboratorio oppure prima/dopo il laboratorio stesso

Esercizio 1.

Categoria: problemi su numeri (Dal problema al programma: 4.2) con sotto-problemi di verifica/selezione per la ricerca del massimo (3.4 e 4.5)

Campionato

In un campionato n (max 20) squadre giocano per m (max 20) giornate. Il risultato di ogni partita è contenuto in una matrice ($n \times m$) di numeri interi, ognuno dei quali può valere soltanto 0, 1 o 3.

Ogni i -esima riga della matrice rappresenta i punti ottenuti dall' i -esima squadra negli m giorni di campionato. Ogni j -esima colonna della matrice rappresenta i punti acquisiti dagli n team durante le partite del j -esimo giorno. I punti acquisiti sono 3 per una partita vinta, 1 per un pareggio e 0 per una sconfitta.

Si scriva un programma in C che:

- Acquisisca il contenuto della matrice da un file di testo il cui nome (massimo 20 caratteri) è acquisito da tastiera. Si definisca un opportuno formato per il file.
Per esempio, è possibile strutturare il file in modo che la prima riga riporti il numero di squadre (n) e il numero di giorni (m) separati da uno spazio, ed ognuna delle righe successive rappresenti una riga della matrice con gli spazi come separatori:

```
6 4
0 1 1 0
3 1 0 1
1 0 1 1
1 3 1 0
1 0 1 3
1 3 3 3
```

- Per ognuna delle giornate, stampi l'indice (i.e. il numero di riga) della squadra in cima alla

classifica (i.e. la squadra con la somma di punti più alta dall'inizio del campionato). In caso di più squadre con lo stesso punteggio in cima alla classifica, il programma stampa la prima che trova.

Esercizio 2. (Esercizio da consegnare per il bonus-laboratorio)

Categoria: problemi di codifica ed elaborazione testi (Dal problema al programma: 4.3 e 4.4)

Ricodifica di testo con dizionario

Un file (`sorgente.txt`) contiene un testo composto da un numero indefinito di righe, di lunghezza massima 200 caratteri ognuna. Un secondo file (`dizionario.txt`) è organizzato come segue:

- sulla prima riga è presente un numero intero e positivo S (≤ 30), che indica il numero di possibili ricodifiche (sostituzioni) presenti nel dizionario
- le successive S linee del file riportano, una per linea, le coppie `<ricodifica><originale>` che rappresentano le sostituzioni possibili.
`<originale>` rappresenta la sequenza di caratteri da ricercare nel file sorgente mentre `<ricodifica>` la sua sostituzione nella forma `$<intero>$`

Lo scopo del programma è di ricodificare il primo file di testo (`sorgente.txt`) andando a sostituire le sequenze di caratteri sulla base dei contenuti del secondo file (`dizionario.txt`). In caso di più sostituzioni possibili per una certa sottostringa, il programma scelga la prima sostituzione trovata. Il risultato della ricodifica sia salvato su un terzo file (`ricodificato.txt`).

Esempio:

Il contenuto del file `sorgente.txt` è:

```
apelle figlio di apollo
fece una palla di pelle di pollo
tutti i pesci vennero a galla
per vedere la palla di pelle di pollo
fatta da apelle figlio di apollo
```

Il contenuto del file `dizionario.txt` è:

```
9
$11$ pelle
$2$ pollo
$333$ palla
$41$ alla
$5078$ tta
$6$ tti
$7$ ll
$81$ er
$900$ ere
```

Il file di uscita `ricodificato.txt` conterrà:

```
a$11$ figlio di a$2$
fece una $333$ di $11$ di $2$
tu$6$ i pesci venn$81$o a g$41$
p$81$ ved$81$e la $333$ di $11$ di $2$
fa$5078$ da a$11$ figlio di a$2$
```

Esercizio 3.

Categoria: problemi su matrici di numeri (Dal problema al programma: 4.1 e 4.2)

Rotazione di matrici

(l'esercizio estende un esercizio simile, applicato ai vettori, del laboratorio 4)

Si scriva un programma C che permetta all'utente di eseguire in sequenza operazioni di rotazione di P posizioni su righe e/o colonne di una matrice di interi. Le rotazioni sono da intendersi come circolari sia sulle righe, sia sulle colonne (vedi definizione data in Laboratorio 4 esercizio 2). Le righe e le colonne sono numerate a partire da 1 (attenzione: in C gli indici delle matrici partono da 0!).

Il programma:

- legge la matrice iniziale (max 30x30) da un file, il cui nome (massimo 20 caratteri) è acquisito da tastiera. Il formato del file prevede sulla prima riga 2 interi che indicano il numero di righe nr e di colonne nc . Le successive nr righe del file, una per ogni riga della matrice, contengono ognuna nc interi separati da spazi.
- acquisisce ripetutamente da tastiera una stringa (al massimo di 100 caratteri, contenente eventuali spazi), nella forma

`<selettore> <indice> <direzione> <posizioni>`

Dove `<selettore>` corrisponde alla stringa "riga", "colonna" o "fine", le quali indicano rispettivamente se l'utente vuole effettuare una rotazione su una riga ("riga"), su una colonna ("colonna") oppure terminare il programma ("fine").

`<indice>` è l'indice della colonna o riga selezionata, partendo da 1.

`<direzione>` corrisponde alla stringa "destra", "sinistra", "su" o "giu" ed indica la direzione della rotazione

`<posizioni>` è un numero intero positivo che indica il numero di posizioni P della rotazione.

Dopo ogni rotazione la matrice ottenuta deve essere stampata a schermo.

NB. Per effettuare le rotazioni, implementare una funzione che generalizza quanto sviluppato per l'esercizio 2 del laboratorio 4.

Esempio di file di input:

```
3 3
1 2 3
4 5 6
7 8 9
```

Esempio di comandi acquisiti da tastiera:

```
riga 2 destra 1
colonna 3 giu 2
riga 1 sinistra 4
colonna 1 su 2
fine
```

La figura seguente illustra il risultato delle operazioni in sequenza:

1	2	3
4	5	6
7	8	9

inizio

1	2	3
6	4	5
7	8	9

riga 2
destra
1 posizione

1	2	5
6	4	9
7	8	3

colonna 3
giù
2 posizioni

2	5	1
6	4	9
7	8	3

riga 1
sinistra
4 posizioni

7	5	1
2	4	9
6	8	3

colonna 1
su
2 posizioni