

Esercizi del 7/2/2013

#### INFORMAZIONI IMPORTANTI:

Il programma richiesto deve leggere l'input da un file **input1** e deve scrivere l'output su un file **output1**. I valori sui file di input e di output sono sempre separati da uno spazio. Il file che contiene la soluzione dell'esercizio (1) **deve chiamarsi esercizio1.cpp**. Per ogni esercizio trovate nella vostra home un file di input ed un corrispondente file di output. Si tratta solo di esempi. Dovreste provare i vostri programmi anche con altri file di input.

ESERCIZIO: Si chiede di scrivere un main che dichiara un array `int A[400]`, poi apre i file di input e di output e legge da input le seguenti quantità:

- i) l'intero `dim > 0`,
- ii) `dim` valori da leggere nelle prime `dim` posizioni di `A`
- iii) 2 interi `lim1` e `lim2` maggiori di 0
- iv) 2 interi `r1` ed `r2` che sono tra 0 e `lim1-1`
- v) 2 interi `c1` e `c2` tra 0 e `lim2-1`.

Dopo questi input il main invoca 2 volte una funzione `R` passandogli in entrambi i casi `A`, `dim`, `lim1` e `lim2` e `r1` la prima volta ed `r2` la seconda. Per ultimo il main deve invocare 2 volte una funzione `C` passandogli `A`, `dim`, `lim1` e `lim2`, `c1` la prima volta e `c2` la seconda volta. Inoltre entrambe le funzioni devono ricevere lo stream di output che permette di stampare su `output1`. Le 2 funzioni hanno la seguente segnatura:

`void R(int* A, int dim, int lim1, int lim2, int r, ofstream & OUT)` e `void C(int* A, int dim, int lim1, int lim2, int c, ofstream & OUT)`. L'ultimo parametro è passato alle funzioni per far sì che possano scrivere su `output1`. Un tale parametro deve sempre venire passato per riferimento.

La `PRE_R=(A[0..dim-1] definita, dim>0, lim1>0, lim2>0, r in [0..lim1-1])`

Le funzioni `R` e `C` devono "vedere" `A` come un array `int X[lim1][lim2]`. Attenzione che `lim1*lim2` potrebbe essere minore di `dim`.

`POST_R=(stampa su OUT gli elementi definiti della riga r di X)`

Attenzione che la riga `r` di `X` potrebbe non avere alcun elemento definito. In questo caso `R` deve stampare su OUT l'intero `-r`.

La `PRE_C` è simile alla `PRE_R` con `c` in `[0..lim2-1]` al posto di `r`.

`POST_C=(stampa su OUT gli elementi definiti della colonna c)`

Attenzione che la colonna `c` di `X` potrebbe non avere alcun elemento definito. In questo caso `C` deve stampare su OUT l'intero `-c`.

**Esempio:** sia `dim=21` e `A[0..20]=[0000 1111 2222 3333 4444 5]`. Supponiamo che `lim1=3, lim2=10` `r=1` e `c=5`. Allora `R` dovrebbe scrivere su OUT per `r=1` i seguenti 10 valori: 2 2 3 3 3 3 4 4 4 4, mentre `C` dovrebbe

scrivere per  $c=5$  i seguenti 2 valori: 1 3. Se  $r=2$ ,  $R$  dovrebbe stampare solo 5, mentre con  $c=0$ ,  $C$  dovrebbe stampare 0 2 5.