

Esercizio 1 del 20/3

Questo esercizio richiede di "vedere" un array ad una dimensione come un array a 3 dimensioni (una torta) e di leggere dei valori interi in questo array a 3 dimensioni **per fette verticali**, cioè i valori vanno inseriti prima nella V-fetta 0 di questo array, poi nella V-fetta 1 e così via. Dopo questa operazione, si dovrà stampare l'array per strati.

Esempio: supponiamo di avere `int X[400]` e di "vederlo" come `int Y[3][6][5]`.

Supponiamo poi di leggere `n_ele=38` e di leggere 38 valori che verranno inseriti nelle V-fette di Y. Ogni V-fetta consiste di una colonna di 6 elementi per ciascuno strato e Y ha 3 strati, quindi ogni V-fetta conterrà 18 interi. Quindi con 38 valori riempiamo interamente $38/18 = 2$ V-fette e i 2 valori che restano andranno nella prima colonna della terza V-fetta.

Se i 38 valori in "input" sono i seguenti:

1 1 2 1 3 1 2 4 5 6

0 0 1 3 5 2 1 0 0 1

3 4 2 3 2 3 2 1 0 0

2 3 1 0 0 2 1 3

inserendoli in Y per V-fette, avremo la seguente situazione nei 3 strati di Y:

strato: 0

1 0 1

1 1 3

2 3

1 4

3 2

1 3

strato: 1

2 2

4 3

5 2

6 1

0 0

0 0

strato: 2

1 2

3 3

5 1

2 0

1 0

0 2

Si osservi che nell'esempio per ogni strato abbiamo mostrato solamente gli elementi letti.

Esercizio: Viene dato un programma che dichiara `int X[400]` e lo riempie tutto col valore `-1` e poi legge da cin le seguenti quantità:

- 1) `lim1`, `lim2` e `lim3` (tutti maggiori di 0 e tali che $\text{lim1} * \text{lim2} * \text{lim3} \leq 400$) e nel seguito si deve vedere `X` come un array `int Y[lim1][lim2][lim3]`;
- 2) `n_ele` ($0 < n_ele \leq 400$); attenzione che `n_ele` può anche essere maggiore di $\text{lim1} * \text{lim2} * \text{lim3}$.

Si richiede di completare il programma con una funzione `leggiV` capace di leggere `n_ele` valori nelle V-fette (0,1, eccetera) dell'array `Y` (ovviamente gli elementi non toccati da queste letture resteranno `-1`). `leggiV` deve avere il seguente prototipo ed essere corretta rispetto alle seguenti pre e post-condizioni:

PRE=(cin contiene `n_ele > 0` interi qualsiasi)

void `leggiV(int* a, int lim1, int lim2, int lim3, int n_ele)`

POST(`a` viene vista come `Y[lim1][lim2][lim3]` e riempita per V-fette con `M` valori, dove $M = \min(\text{lim1} * \text{lim2} * \text{lim3}, n_ele)$)

Oltre a `leggiV`, va scritta una seconda funzione `stampaS` col compito di stampare tutti gli elementi di `Y` (anche quelli `-1`) per strati. Quindi prima lo strato 0, poi lo strato 1 e così via. Naturalmente ogni strato verrà stampato per riga. Quindi, lo strato 0 dell'esempio precedente va stampato come segue:

strato: 0

1 0 1 -1 -1

1 1 3 -1 -1

2 3 -1 -1 -1

1 4 -1 -1 -1

3 2 -1 -1 -1

1 3 -1 -1 -1

La stampa di ciascuno strato va preceduta dalla stringa "strato n" con $n=0,1,\dots$ come nell'esempio precedente.

Il prototipo, la pre e la post-condizione di stampaS sono:

PRE=(a[0.. $\text{lim1}*\text{lim2}*\text{lim3}-1$] è definita)

void stampaS(int* a, int lim1,int lim2,int lim3).

POST=(vede a[0.. $\text{lim1}*\text{lim2}*\text{lim3}-1$] come Y[lim1][lim2][lim3] e la stampa per strati)

Correttezza: scrivere un invariante dei cicli della funzione leggiV e dimostrare la correttezza della funzione rispetto alla pre e post-condizione date.

NOTA: sia leggiV che stampaS possono invocare altre funzioni ausiliarie.