Esercizio 3 del 17/4

Consegnare corretto entro il 21/4

L'esercizio inizia come gli Esercizi 1 e 2 di questa settimana. Il programma richiesto dichiara un array X di 400 interi, legge lim1, lim2 e lim3 e, come al solito, deve "vedere" X come un array int Y[lim1][lim2][lim3]. Poi il programma deve leggere n_el e di seguito deve leggere n_el interi in Y per strati. Le letture sono tutte dal file "input", n_el è sempre maggiore di 0 e non maggiore di 400, lim1, lim2 e lim3 sono positivi. E' possibile che n_el sia maggiore di lim1*lim2*lim3, ma questo non ci deve preoccupare troppo perché n_el è comunque minore o uguale del numero di elementi di X.

Ci interessano le V-fette di Y. Come in esercizi precedenti, vogliamo considerare (la parte definita) di queste V-fette come un unico array ad una dimensione, ma questa volta vogliamo considerare un ordine diverso degli elementi della V-fetta. Ogni V-fetta è costituita da tasselli e vogliamo considerare l'ordine in cui gli elementi del primo tassello della V-fetta vengono prima di quelli del secondo tassello e così via e all'interno di un singolo tassello gli elementi sono ordinati dall'alto verso il basso. Un esempio dovrebbe chiarire.

Esempio: sia lim1=3, lim2=4 e lim3=5, n_el=48 e supponiamo che i 48 elementi siano i seguenti:

strato 0 strato 1 strato 2

0 0 0 0 0 1 1 1 1 1 2 2 2 2 2

0 0 0 0 0 1 1 1 1 1 2 2 2

0 0 0 0 0 1 1 1 1 1

0 0 0 0 0 1 1 1 1 1

Se consideriamo la V-fetta 0, il suo primo tassello sarà costituito da 0 1 2 dove lo 0 è l'elemento Y[0][0][0], l'1 è l'elemento Y[1][0][0] e il 2 è l'elemento Y[2][0][0]. Il secondo tassello della V-fetta 0 sarà costituita dagli elementi Y[0][1][0], Y[1][1][0] e Y[2][1][0] e così via. Ovviamente i tasselli 2 e 3 della V-fetta 0 saranno di soli 2 elementi (0 e 1). Così come di 2 elementi saranno i tasselli 2 e 3 di tutte le V-fette e anche i tasselli 1 delle V-fette 3 e 4.

Considerando di seguito gli elementi della V-fetta 0 in ordine per tasselli, avremo:

0 1 2 0 1 2 0 1 0 1 dove separiamo i contributi dei 4 tasselli della V-fetta 0 con uno spazio aggiuntivo. Ogni elemento ha un indice, 0,1,2,3,4,..eccetera.

Gli elementi della V-fetta 3 ordinati per tasselli sono: 0 1 2 0 1 0 1 0 1 .

Il problema che vogliamo risolvere è il seguente: fissata una V-fetta f ed un intero e, vogliamo calcolare la distanza dell'elemento di indice e della V-fetta f rispetto all'inizio di Y.

Riprende l'Esempio: Quindi per la V-fetta 0, per e= 7, la risposta è, (lim2*lim3)+2*lim3=30.

Si tratta dell'elemento Y[1][2][0].

Per e=10, visto che la V-fetta 0 non ha 11 elementi, la risposta dovrebbe essere: "10 elemento inesistente della V-fetta 0".

Per la V-fetta 3, per e=7 la risposta è, 3*lim3+3=18. Si tratta dell'elemento Y[0][3][3].

Dopo le letture spiegate prima, il programma deve leggere da "input" 3 gruppi di 4 valori ciascuno: f, e1, e2, e3, dove il primo valore individua la V-fetta da considerare, e gli ultimi 3 individuano i suoi elementi per cui è richiesto il calcolo della distanza dall'inizio di Y.

Per ciascun gruppo di 4 interi il programma deve stampare su "output" le distanze di e1, e2 ed e3 dall'inizio di Y. In caso qualche ej non fosse un elemento presente della V-fetta, il programma dovrebbe scrivere la frase "ej elemento inesistente della V-fetta f".

Consigli:

a) conviene inizialmente considerare il caso in cui Y sia completamente pieno e poi estendere la soluzione per questo caso semplice al problema generale.

b) **il main è dato**. Esso è nel file esercizio3_17_4.cpp e fa tutte le operazioni di lettura e scrittura richieste. Inoltre esso invoca la funzione,

int calcola_dist(int fetta, int e, int lim2, int lim3, int n_el)

che dovete fare voi. La funzione deve restituire la distanza di e dall'inizio dell'array Y. Sarà anche utile definire una funzione che calcoli il numero di elementi definiti di un dato tassello. Dovete usare il file exercizio3_17_4.cpp, aggiungendogli le funzioni richieste e anche le parti di correttezza come commenti.

Correttezza: scrivere Pre e Post delle funzioni che definite. Scrivere un invariante per i cicli della funzione calcola_dist.