Esercizio 3 del 14/3/2017 Questo esercizio è da fare per partecipare ai compiti

Questo esercizio richiede di leggere un numero n_ele (0<n_ele<=400) di interi in un array int X[400] e poi di "vedere" questo array come se fosse un array a 3 dimensioni per esempio come un array int Y[4][5][8] oppure come un array int Z[3][3][10] o ancora come int W[2][5][5].

Esempio 1: Supponiamo che n_el= 66 e che questi siano i 66 valori interi da leggere in X:

1 2 1 0 0 0 2 2 2 2 2 1 1 2 1 0 3 1 0 0 1 1 1 1 1 2 1 0 1 1 0
1 1 2 1 0 1 1 0 1 1 2 1 0 0 0 2 2 0 2 2 2 0 1 2 2 0 1 2 2 2 1
1 2 1 0

"Vedere" X come Y significa vederlo nel modo seguente:

strato 0

r0: 1 2 1 0 0 0 2 2

r1: 2 2 2 1 1 2 1 0

r2: 3 1 0 0 1 1 1 1

r3: 1 2 1 0 1 1 0 1

r4: 1 2 1 0 1 1 0 1

strato 1

r0: 1 2 1 0 0 0 2 2

r1: 0 2 2 2 0 1 2 2

r2: 0 1 2 2 2 1 1 2

r3: 1 0

Quindi Y ha solo 1 strato completamente pieno ed uno con tre righe complete e una riga finale con solo 2 elementi.

Invece, "vedere" X come Z significa vederlo nel modo seguente:

strato 0

r0: 1 2 1 0 0 0 2 2 2 2

r1: 2 1 1 2 1 0 3 1 0 0

r2: 1 1 1 1 1 2 1 0 1 1

strato 1

r0: 0 1 1 2 1 0 1 1 0 1

r1: 1 2 1 0 0 0 2 2 0 2

r2: 2 2 0 1 2 2 0 1 2 2

strato 2

r0: 2 1 1 2 1 0

Cosa significhi "vedere" X come W lo lasciamo come esercizio.

Si osservi che le diverse "visioni" di X risultano in array che possono anche non essere interamente definiti. Questo succede nell'Esempio 1 sia quando vediamo X come Y che quando lo vediamo come Z. E' anche possibile "vedere" X come un array tale che il numero dei suoi elementi sia inferiore a n_ele. Per esempio int K[2][2][10]. In questo caso si dovrebbero considerare solo i primi 40 elementi dei 66 definiti in X e K risulterebbe completamente pieno.

Esercizio : scrivere un programma che dichiara un array int X[400] e legge il valore n_ele (0<n_el<=400) e poi legge n_ele valori inserendoli in X. Successivamente deve leggere anche tre interi positivi lim1, lim2 e lim3 e deve "vedere" X come un array int [lim1][lim2][lim3]. Per esempio, se lim1=4,lim2= 5 e lim3= 8, allora deve "vedere" X come Y dell'Esempio 1, mentre se lim1=3, lim2=3 e lim3=10, allora deve "vedere" X come Z e così via per ogni tripla lim1, lim2 e lim3.

Attenzione: non si deve dichiarare alcun array int [lim1][lim2][lim3], ma solo "vedere" X come un tale array e stampare di conseguenza i suoi valori definiti.

Il programma deve stampare su cout gli strati della "visione" int[lim1][lim2][lim3]di X. Questo deve venire fatto esattamente come indicato nell'Esempio 1 per Y e Z, comprese le stringhe "strato 0", "strato 1", eccetera, all'inizio dello strato 0, 1 eccetera e anche con le stringhe "r0:", "r1:", eccetera, all'inizio della riga 0, 1 , eccetera di ciascuno strato.

Il programma deve usare una funzione stampaS che si occupa di stampare uno strato (con tanto di scritta iniziale "strato n", dove n sarà 0,1 eccetera) che è costituito da una sequenza di righe, ciascuna preceduta dalla scritta "r n:", dove n sarà 0,1,eccetera. StampaS può invocare anche altre funzioni.

Attenzione: Vanno stampati solo gli elementi definiti e solo strati e righe con almeno un elemento definito. Strati completamente vuoti e righe completamente vuote non causano alcuna stampa.

Correttezza: scrivere PRE e POST della funzione stampaS descritta in precedenza. Dimostrare che la funzione fa quello che avete specificato nella relativa POST.