Corso di programmazione 2010-2011

Terza esercitazione per casa: assegnata il 4 febbraio 2011

Consegna il 10 febbraio entro le 10 di mattina, con il comando: consegna settimana3

Le 2 funzioni richieste devono essere corredate da pre- e postcondizione ed ogni ciclo deve avere il suo invariante. Per ogni funzione si deve delineare una dimostrazione di correttezza.

Introduzione: in C e C++ un array a 3 dimensioni viene immagazzinato in memoria come una sequenza contigua di strati a 2 dimensioni che a loro volta sono immagazzinati come sequenze contigue di array a 1 dimensione che sono sequenze di elementi. Questa allocazione definisce quindi un ordine tra gli elementi di A. Nel seguito, quando facciamo riferimento ai primi n elementi di un array, intendiamo i primi n elementi rispetto a questo ordine.

1) Si chiede di scrivere una funzione bool F(int A[][8][9], int B[][8][9],int limite1, int n_ele), con n_ele > 0. Si deve assumere che i parametri di F soddisfino le seguenti condizioni: A e B hanno limite1 strati, inoltre limite1*5*10 >= n_ele ed i primi n_ele elementi di A e di B sono definiti mentre gli eventuali altri elementi sono indefiniti. Questa condizione va espressa nella precondizione.

F deve restituire true se A ha uno strato identico ad uno strato di B e false altrimenti.

Spiegazione: Diremo che 2 strati sono identici quando ogni riga del primo è identica alla corrispondente riga del secondo e 2 righe sono identiche se sono la stessa sequenza di valori. E' necessario fare attenzione al fatto che devono essere considerati solo gli elementi definiti dei 2 array. Visto che solo i primi n_ele elementi dei 2 array vanno considerati, in generale, i 2 array A e B saranno costituiti da alcuni strati completamente pieni di elementi definiti seguiti da uno strato solo parzialmente riempito[1]. Ovviamente questo strato finale di A può essere uguale solo all'analogo strato finale di B, mentre uno strato pieno di A può essere identico a qualsiasi strato pieno di B.

2) A lezione abbiamo visto che un array con qualsiasi numero di dimensioni può essere visto e percorso come fosse un array ad una sola dimensione grazie all'allocazione contigua degli array garantita dal C e C++. In questa visione, dato un array A[lim1][lim2][lim3][lim4], chiameremo sottosequenza di A che inizia in (i,j,k,h) una sequenza di elementi di A che inizia con l'elemento A[i][j][k][h] e continua con alcuni elementi di A immediatamente successivi a A[i][j][k][h]. Una sottosequenza ha lunghezza pari al numero dei suoi elementi. La quadrupla (i,j,k,h) è detta punto di partenza della sottosequenza.

Si chiede di scrivere una funzione void G(int * A, int *B, int lim1, int lim2, int lim3, int lim4, int &i, int&j, int&k, int&h, int&lung) per cui si deve assumere che: A e B puntano al primo elemento di un array a 4 dimensioni e con limiti lim1, lim2, lim3 e lim4 e che tutti gli elementi dei 2 array sono definiti.

G deve calcolare la quadrupla (i,j,k,h) che individua il punto di partenza a partire dal quale iniziano sottosequenze di A e B che sono identiche tra loro e che hanno lunghezza massima tra tutte le sottosequenze identiche di A e di B che iniziano in uno stesso punto di partenza. Il punto di partenza va restituito attraverso i 4 parametri i, j, k e h passati per riferimento, mentre lung sarà la lunghezza della

[1] Si deve prestare attenzione al fatto che l'ultimo strato con elementi definiti, potrebbe essere anche completamente pieno di elementi definiti.

sottosequenza trovata. Qualora non ci fosse alcun punto di partenza a partire dal quale A e B abbiano sottosequenze identiche, allora il parametro lung dovrebbe assumere valore 0 (mentre i valori di i,j,k e h possono essere qualsiasi). Ci fossero vari punti di partenza da cui iniziano sottosequenze di A e di B con lunghezza uguale tra loro e maggiore di tutte le altre, G dovrà restituire il primo punto di partenza[2] tra quelli che danno sottosequenze identiche con lunghezza massima.

Esempio: per motivi evidenti consideriamo 2 array A e B a 2 dimensioni. Le righe sono separate da virgole.

A=[aaaa, aabd,bbwb,baaa] e B=[acca,acbd,bbwc,aaaa]

E' facile vedere che dal punto di partenza (1,2) iniziano le sottosequenze di A e B identiche tra loro e più lunghe. Le 2 sottosequenze sono costituite da bdbbw. Ci sono anche altri punti di partenza da cui iniziano sottosequenze identiche, per esempio (0,0) con lunghezza 1, (0,3) con lunghezza 2 e anche (3,1) con lunghezza 3.

[2] Secondo l'ordine di allocazione degli elementi degli array.