

III Compitino di Programmazione del 20/4/2016 Turno 2

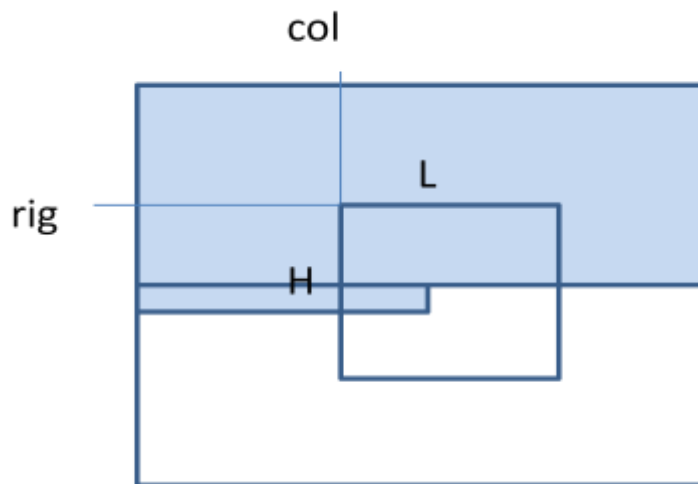
Teoria: (risposte come commento sul file del programma) (4 punti)

-dato l'array `int A[3][4][6][8]`, che tipo ha A e di quanti byte è l'oggetto puntato da A? Se A ha valore A, che valore e che tipo ha l'espressione, `*(A+3)[-2]` ?

-se abbiamo 2 funzioni `int f1(int,float)` e `float f1(float, int)` ed una invocazione `f1(3, 2)` quale delle 2 funzioni disponibili verrà scelta dal compilatore?

Programmazione (19 punti)

In questo esercizio consideriamo il consueto array X di 400 interi che vedremo come un array `int Y[lim1][lim2]` a 2 dimensioni. Poi considereremo un sotto-array di Y e cercheremo di scandire gli elementi delle colonne di questo sotto-array. La nozione di sotto-array è illustrata dalla seguente figura:



Il rettangolo grande è l'array a 2 dimensioni `Y[lim1][lim2]` (che non esiste realmente). Il sotto-array è il rettangolo più piccolo. Esso è individuato dalla riga e colonna (rig,col) in Y del suo primo elemento e dal numero di righe H e dal numero di colonne L. Come sempre l'array Y è riempito con `n_el` valori e quindi avremo una parte definita (la parte azzurra) e una parte indefinita (in bianco). Il sotto-array può avere anche lui delle righe piene e una più corta (come nella figura), ma potrebbe essere anche tutto definito o anche tutto indefinito, a seconda di (rig, col) e di H e L. **In ogni caso, nello svolgimento dell'esercizio si deve assumere che i valori di (rig,col), L ed H definiscano un sotto-array completamente contenuto in Y.** Quindi non si deve fare alcun controllo in questo senso.

Il programma da realizzare ha lo scopo di considerare le colonne del sotto-array facendo attenzione a considerare solo gli elementi definiti delle colonne. Il suo compito è di stampare l'indice della colonna del sotto-array che ha il massimo numero di ripetizioni di uno stesso valore. Alla fine del documento trovate un esempio.

Viene dato un main e anche alcune funzioni. In particolare la funzione F che soddisfa la seguente specifica:

PRE=(X contiene almeno `n_el` valori, gli altri parametri hanno valori sensati, in particolare rig, col, L ed H individuano correttamente un sotto-array di `Y[lim1][lim2]`)

`int F(int*X, int n_el, int lim1, int lim2, int rig, int col, int L, int H, int f)`

POST=(restituisce il massimo n. di ripetizioni di uno stesso valore nella colonna f, se la colonna f è vuota, restituisce -1)

La funzione F invoca 2 funzioni che sono quelle da fare : una funzione FV ed una funzione cambiaM che sono come segue:

La funzione FV assomiglia parecchio alla funzione con lo stesso nome vista negli esercizi della settimana scorsa. Quando la invochiamo passandole un indice f di colonna del sotto-array ed un indice i di elemento per quella colonna, controlla che l'indice corrisponda ad un elemento definito di X e in caso sia così, restituisce il valore dell'elemento cercato, mentre in caso contrario solleva un'eccezione che trasporta il valore false che serve a terminare il ciclo di F. La specifica di FV è la seguente:

PRE=(X contiene n_el valori, gli altri parametri hanno valori sensati, in particolare rig, col, L ed H individuano correttamente un sotto-array di Y[lim1][lim2])

int FV(int*X,int lim1, int lim2, int n_el, int rig, int col, int L, int H, int f, int i)

POST=(se la colonna f del sotto-array individuato da rig, col, L e H ha l'elemento di indice i definito, allora FV restituisce il suo valore, altrimenti lancia una throw con valore false).

La funzione void cambiaM(int y, int M[][2], int & dimM), (notate il parametro passato per riferimento) ha le seguenti PRE e POST:

PRE=(dimM>=0 e M ha almeno dimM righe, si chiami vdimM il valore iniziale di dimM)

POST=(restituisce una nuova M che tenga conto del nuovo valore y, quindi se nella prima colonna M[0..dimM-1][0] c'è già y, allora incrementa di 1 la corrispondente seconda colonna mantenendo dimM = vdimM, altrimenti inserisce una nuova riga con valori (y,1) e dimM=vdimM+1)

Correttezza: (9 punti)

i) scrivere una spiegazione del fatto che FV fa quanto richiesto dalla sua POST.

ii) scrivere un invariante per il ciclo contenuto in cambiaM e fare la prova completa di quel ciclo e dell'intera funzione.

Attenzione: non sono accettate soluzioni in cui venga definito un array delle dimensioni di Y e neppure soluzioni che riempiono X di valori sentinella oppure usino array ausiliari per riconoscere quando si esce dalla zona definita di X.

Esempio.

Si consideri che lim1=6 e lim2=7 e che n_el sia 18 e che l'array Y contenga i seguenti 18 valori:

0	2	1	0	2	3	4
1	1	1	0	2	0	1
3	3	3	0			

se rig=1, col=2, L=3, H=3, allora il sotto-array da considerare sarà questo:

1	0	2
3	0	

e quindi la colonna 1 del sotto-array avrebbe il massimo n. di ripetizioni di un valore (0 ripete 2 volte). Se invece rig=3, mantenendo tutto il resto inalterato, allora il sotto-array sarebbe vuoto e quindi il programma risponderebbe con "Migliore colonna= -1 con -1 ripetizioni".