

## Corso di programmazione 2009-2010

Terza esercitazione per casa: assegnata il 26 gennaio 2010

Consegna il 2 febbraio entro le 8 di mattina, con il comando: **consegna settimana3**

1) Sappiamo che 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.

Si chiede di scrivere una funzione `int F(int A[][5][10], int limite1, int n_ele)`, con  $n\_ele > 0$  che soddisfa le seguenti condizioni:

- i) A ha `limite1` strati ed ha i primi `minore(limite1*5*10, n_ele)` elementi definiti (e gli eventuali altri indefiniti)<sup>1</sup>.
- ii) F deve restituire l'indice dello strato che abbia somma degli elementi massima in A. E' importante che nel calcolo della somma degli elementi di ciascuno strato siano considerati solamente gli elementi di A che sono definiti (ed esclusi quelli indefiniti).
- iii) qualora ci fossero strati diversi con somma degli elementi uguale tra loro e maggiore di quella di tutti gli altri strati, F deve restituire l'indice di uno qualsiasi degli strati massimi.

2) A lezione abbiamo spesso guardato un array a 3 dimensioni come una torta rettangolare composta da strati, ognuno dei quali è un array `int [5][10]`. Continuando con la simbologia della torta, possiamo anche definire le **fette** della torta come composte da righe degli strati: la prima fetta è composta dalle prime righe di tutti gli strati, la seconda fetta è composta dalle seconde righe di tutti gli strati e così via.<sup>2</sup> Chiaramente gli elementi di una fetta non sono allocati in modo contiguo nella memoria come lo sono invece quelli degli strati.

Si chiede di realizzare una funzione `int G(int* A, int limite1, int limite2, int limite3, int n_ele)`, con  $n\_ele > 0$  che soddisfi i seguenti punti:

- i) G riceve il puntatore al primo elemento dell'array A (che ha infatti tipo `int*` in questo caso), ma deve interpretare l'array come un array a 3 dimensioni `int [limite1][limite2][limite3]` che ha i primi `minore(limite1*limite2*limite3, n_ele)` elementi definiti (e gli eventuali altri indefiniti).
- ii) G deve restituire l'indice della fetta la somma dei cui elementi sia massima in A. Quindi è lo stesso esercizio del punto (1), con la differenza che si devono considerare fette anziché strati. In particolare, come in (1), la somma degli elementi di una fetta deve considerare solo gli elementi definiti di A.
- iii) Il caso in cui ci siano più fette a somma massima si risolve come spiegato in (1)(iii).

---

<sup>1</sup> `minore(x,y)` rappresenta il minore tra x e y.

<sup>2</sup> Ovviamente potremmo definire anche fette composte da colonne degli strati, ma in questo esercizio consideriamo le fette come composte dalle righe degli strati.

