

## Primo Compitino di Programmazione 14/2/2013 I Turno

**Teoria:** dato l'array `char X[5][10][10][10]` specificare per ciascuna delle seguenti espressioni, il loro tipo, in caso si tratti di un tipo puntatore, la dimensione dell'oggetto puntato e il loro valore (l'R-valore di X va indicato con X)

`-(X[-4]+2)-3`

`-X[-2][-2]+2`

**Programmazione.** Dato un array A 10x10 di interi del quale solo i primi `dim` (`dim ≤ 100`) elementi sono definiti, si tratta di determinare la colonna che contiene il minimo numero di valori distinti. E' importante tenere presente che si dovranno considerare solo gli elementi definiti di A. Chiariamo subito la cosa con un esempio.

**Esempio:** sia `dim=23` e questi gli interi che vengono letti nelle prime `dim` posizioni di A, [0 1 1 0 1 2 0 1 2 3 0 1 0 1 0 1 0 1 0 0 1] in cui abbiamo inserito uno spazio extra dopo ogni 10 valori per aiutare la comprensione. Visto che `dim=23`, le prime 2 righe di A saranno completamente definite (10 valori ciascuna) e la terza riga avrà solo i primi 3 elementi definiti. La colonna 0 conterrà [0 0 0] e quindi un solo valore ripetuto 3 volte, mentre la colonna 5 conterrà [2 1] e quindi 2 valori distinti. E' evidente che nessuna colonna può fare meglio della colonna 0 in cui compare un solo valore (0 ripetuto 3 volte).

Per determinare la colonna di A in cui appaiono il numero minimo di valori distinti, si devono realizzare 2 funzioni:

`PRE_F=(A è array 10x10 con i primi dim elementi definiti, dim>0)`

`int F(int A[][10], int dim, int & kol);`

`POST_F=F` restituisce un intero `v` col `return` e un altro intero `kol`, passato per riferimento, questi due valori restituiti sono tali che: (la colonna `kol` di A ha un numero pari a `v` di valori distinti e nessuna altra colonna di A ha un n. inferiore a `v` di valori distinti).

La funzione F deve invocare una funzione ausiliaria CC con il seguente prototipo e PRE e POST-condizioni:

`PRE_CC=(X è array 10x10 con i primi dim elementi definiti, dim>0) && (c è una colonna di X che sicuramente contiene qualche elemento definito)`

`int CC(int X[][10], int dim, int c)`

`POST_CC=( CC restituisce il numero di valori distinti contenuti nella colonna c di X (considerando solo gli elementi definiti di c))`

**Correttezza:** si chiede di inserire asserzioni (in particolare invarianti per i cicli) nel corpo di F e di usarle per dimostrare la correttezza di F rispetto alle `PRE_F` e `POST_F` date. Per questa prova potete assumere che CC sia corretta rispetto a `PRE_CC` e `POST_CC`.

Si faccia attenzione alla condizione ( $c$  è una colonna di  $X$  che sicuramente contiene qualche elemento definito) contenuta in `PRE_CC`. Essa richiede che  $F$  invochi  $CC$  solo dopo aver verificato questa proprietà.

**Attenzione:** Nella vostra home, assieme a questo documento trovate anche un file `esercizio1.cpp` che contiene un `main` che potrete usare per testare le vostre funzioni. **Il main esegue tutto l'input e l'output richiesto.** Il `main` invoca la funzione  $F$  (che dovete fare voi) in modo consistente con il prototipo fissato prima. Nella vostra home trovate un file `input1` ed il corrispondente `outputX1` che può servire in fase di test. Si osservi che il nome del file è `outputX1` anziché `output1`, come usato nel `main`, per proteggere il file da sovra-scritture non desiderate in fase di test.

Come sempre il programma da consegnare deve essere su un file che si chiama `esercizio1.cpp`. Il comando di consegna è (come sempre): `consegna esercitazione` e va eseguito avendo come directory corrente una directory che contiene solo il file dell'esercizio (ed eventualmente i file di i/o, ma niente di più).

Teoria e correttezza vanno inseriti nel file `esercizio1.cpp`, assieme al programma, come commenti, ma non all'interno del programma, bensì prima o dopo il programma.