

**Esercizio per casa 3    consegna entro 2/2/2009 a mezzanotte    comando : consegna prog3**

1) Si tratta di scrivere una funzione F che dato un array `int A[5][6]` riempie un array `bool C[5][5]` in modo tale che `C[i][j]==true` se e solo se la riga `A[i]` è contenuta nella riga `A[j]`. Dove una riga è contenuta in un'altra quando ogni elemento della prima riga è contenuto anche nella seconda.

Quindi la riga `[1,1,1,1,1,1]` è contenuta nella riga `[1,2,3,4,5,6]`, ma non viceversa.

La funzione F deve aver tipo: **`void F(int A[][6], bool C[][5]);`** F deve solo calcolare C non stamparla e deve assumere che `A[5][6]` sia definita (non è richiesta alcuna lettura).

Si richiede di specificare, pre e post-condizione di F e anche gli invarianti dei cicli di F argomentando perché sono invarianti.

2) Si tratta di scrivere una funzione **`int PM( char T[][10], int righe_T, char P[], int ele_P )`** in cui `righe_t` è il numero di righe di T e `ele_T` è il numero di elementi di P. La funzione deve calcolare per ciascuna riga di T il numero di match di P nella riga e restituire l'indice della riga che ha massimo numero di match. In caso ci siano righe con uguale numero di match massimale allora PM deve restituire l'indice più piccolo tra quelli di queste righe. Cosa sia un match di P in una riga è spiegato con l'esempio seguente:

*Esempio: sia  $P=['a','v','a']$  e una riga di T sia*

*$['a','c','v','a','v','a','v','a','v','c']$  il numero di match di P in questa riga è 2, il primo match è sulle posizioni  $[2,3,4]$  ed il secondo sulle posizioni  $[4,5,6]$ . Notare che entrambi i match usano la posizione 4 della riga. Si ricordi che gli elementi della riga partono dall'indice 0 e arrivano a 9.*

**Consiglio:** usare una funzione supplementare che viene invocata da PM.

Si richiede di specificare pre- e post-condizioni di tutte le funzioni e anche gli invarianti dei cicli.