

Esercizio 3 del 6/4/2016

Dato un array `int A[100]` che contiene `n_el` valori, `n_el ≤ 100`, vogliamo costruire un array `int M[100][2]` che su ogni riga contenga un valore che appare in `A` e il numero di occorrenze in `A` di questo valore. La prima colonna di `A` deve contenere solo valori distinti, quindi per ogni valore di `A` ci sarà una sola riga in `M`. Il numero di righe riempite in `M` sarà uguale al numero di valori distinti che appaiono in `A`. Certamente le 100 righe di `M` basteranno sempre alla bisogna visto che `A` ha al più 100 valori.

Esempio: sia `A=[2,3,1,2,3,0,2,0,1,3]` allora `M=`

2 3

3 3

1 2

0 2

E' dato un `main` che legge `n_el` valori in `A` e invoca la funzione `F` che deve riempire `M`. Il prototipo di `F` è il seguente:

`PRE=(A contiene n_el ≥ 0 valori)`

`int F(in*A, int n_el, int (*M)[2]);`

`POST=(la funzione restituisce il numero n di righe riempite di M e riempie le prime n righe di M con i valori corrispondenti al contenuto di A)`

La funzione `F` dovrebbe usare una seconda funzione che determina se il "prossimo" valore di `A` è già presente in `M` o no e, in caso, in quale riga di `M`. A voi decidere come realizzare una funzione che restituisca 2 risultati: booleano e indice.

Correttezza: Scrivere `PRE` e `POST` della seconda funzione. Scrivere l'invariante di ogni ciclo e la prova del ciclo di `F`.