Esercizio 6 10/11/2014

Si deve sviluppare un programma che dichiara 2 array, int T[10][5] e int P[10], riempie T con 50 valori letti da cin, poi riempie P con 10 valori letti anch'essi da cin. Dopo di che l'esercizio si sviluppa in 3 modi diversi:

1) si chiede di calcolare quante colonne di T sono identiche a P.

2) si chiede di calcolare la colonna di T che presenta il minimo numero di mismatch rispetto a P, più precisamente, se consideriamo la colonna i (0<=i<5) di T, si confronta P[0] con T[0][i], P[1] con T[1][i], P[2] con T[2][i] e così via, e si calcola il numero di questi confronti che falliscono, se questo viene fatto per tutte le colonne, si deve determinare la colonna per cui questo numero è minimo. In caso di più colonne con lo stesso numero minimo di mismatch, si chiede la colonna di indice minore.

3) Questa versione dell'esercizio prevede anche un input diverso. Innanzitutto si leggono da cin 2 interi positivi dimT e dimP (0<dimT<=50 e 0<dimP<=10) e poi si leggono dimT interi in T e dimP interi in P. Gli interi in T vanno letti per riga, nel modo spiegato nel seguente esempio.

Esempio 1: se dimT=32, allora i 32 valori verranno letti in T in modo che i primi 30 valori riempiano le prime 6 righe (6*5=30) e gli ultimi 2 valori vengano messi nei primi 2 elementi della settima riga che sarà quindi solo parzialmente definita.

A questo punto si deve determinare la colonna di T che contiene il massimo numero di match contigui e completi di P. E' importante capire che il match di P su una colonna di T deve interessare solo gli elementi definiti della colonna.

Esempio 2: nella situazione descritta nell'Esempio 1, le colonne 0 e 1 di T avranno 7 elementi definiti, mentre le colonne di indice 2, 3 e 4 avranno solo 6 elementi definiti. Quindi per le prime 2 colonne il match dovrà considerare 7 elementi, mentre per le altre solo 6.

Correttezza:

a) per ognuno dei 3 esercizi scrivere una PRE ed una POST che descrivano con precisione il compito del programma.

b) si cerchi di scrivere l'invariante di ogni ciclo **prima** di definire il corpo del ciclo stesso. Si tratta di sforzarsi di pensare con precisione al compito che il ciclo deve eseguire e descriverlo nell'invariante. Non è importante che l'invariante sia subito completo e nemmeno che sia subito giusto. E' possibile completarlo e migliorarlo, durante lo sviluppo del programma, in successive fasi di raffinamento. Correttezza e programma si aiutano a vicenda.

Quindi ogni ciclo deve avere un suo invariante. Per almeno un ciclo si chiede di scrivere anche la post-condizione del ciclo e di delineare la completa dimostrazione di correttezza del ciclo in 3 parti.