

Esercizio a tempo n. 3 del 28/3/2019

Come per il precedente esercizio a tempo, il main dichiara un array `int X[100]`, poi legge `n_ele (>0)` e di seguito legge `n_ele` valori in `X`. Di seguito legge `lim1`, `lim2` e `lim3` e nell'esercizio si dovrà "vedere" `X` come fosse a 3 dimensioni `int T [lim1][lim2][lim3]`.

Si legge ancora `nP` e si leggono `nP` valori nell'array `P`. `P` è il pattern e l'esercizio chiede di cercare un match di `P` nelle V-fette di `T`.

Esempio. Supponiamo che `n_ele = 26` e che i valori letti in `X` siano,

```
1 2 1 1
0 1 2 2
1 1 0 0
0 1 0 1
1 1 1 1
0 1 1 0
1 2
```

Supponiamo che `lim1=3`, `lim2=4` e `lim3=4`, che `nP=4` e `P=[1,1,1,2]`.

Esaminando le V-fette da sinistra a destra, non è difficile vedere che la V-fetta 0 non ha copie di `P`, mentre la V-fetta 1 ne ha 1 che inizia nell'elemento 3 della V-fetta. In questo caso il programma dovrebbe produrre la seguente stampa:

La V-fetta 1 contiene un match che inizia nell'elemento 3.

In caso in cui nessuna V-fetta contenga un match di `P`, il programma dovrebbe stampare: Nessun match trovato

Si chiede di estendere il main dato in modo che faccia i match richiesti e produca l'output appropriato nei diversi casi.

E' caldamente consigliato di seguire la soluzione dell'esercizio a tempo n. 2 proposta in classe e messa sul moodle. In particolare è consigliato l'uso di una funzione.

Correttezza: Inserire gli invarianti dei cicli principali. Scrivere Pre- e Post-condizione della funzione e dimostrare che la funzione è corretta rispetto a queste Pre e Post.