

## Esercizio 1 del 13/4

### Consegnare corretto entro il 21/4

E' un esercizio che inizia come alcuni esercizi precedenti, ma poi diventa un esercizio di pattern matching.

Il programma da fare deve dichiarare un array `int X[400]`, riempirlo con 400 valori -1, e poi dichiarare `int P[50]`, e infine deve leggere da "input" i seguenti valori:

i) 3 interi `lim1`, `lim2` e `lim3` (tutti positivi), nel seguito il programma deve "vedere" `X` come un array `int Y[lim1][lim2][lim3]`;

ii) `n_el` ( $0 < n\_el \leq 400$ ) e poi `n_el` valori che metterà in `Y` per strati;

ii) `dimP` ( $0 < dimP \leq 50$ ) e `dimP` valori che metterà in `P` partendo dall'inizio di `P`.

Si deve considerare tutte le V-fette di `Y` con almeno un elemento letto da "input" (quindi `i-1` non vanno considerati) e per ciascuna di esse si deve calcolare il numero di match di `P[0..dimP-1]` su quella V-fetta. Dopo aver considerato la fetta `n` il programma deve scrivere su "output" la scritta "V-fetta `n` = `M` match" che indica che la V-fetta `n` ha `M` match di `P[0..dimP-1]`.

I match da considerare sono **contigui e non sovrapposti**. Questo viene spiegato nell'esempio che segue.

**Esempio:** assumiamo che `lim1=3`, `lim2= 4`, e `lim3=5` e quindi `X` deve essere "visto" come `int Y[3][4][5]`. Assumiamo anche che `n_el=48` e che i 48 elementi letti da "input", già distribuiti sugli strati di `Y`, siano come segue:

strato 0	strato 1	strato 2
1 2 1 0 0	0 2 2 3 1	1 2 0 1 1
0 0 1 1 1	1 0 2 1 0	0 0 0
1 1 0 1 0	2 1 2 2 0	
1 2 2 2 2	2 2 1 2 1	

Ovviamente lo strato 2 in realtà sarebbe completato con i -1. Questi valori -1 sono inseriti solo per facilitare l'individuazione degli errori in fase di test. Quindi il programma deve ignorarli e per questo non li abbiamo inseriti nell'esempio.

Assumiamo inoltre che  $\text{dim}P=5$  e che i 5 elementi letti in P siano: 0 1 2 2 0.

La V-fetta 0 di Y contiene i seguenti elementi:

1 0 1 1 0 1 2 2 1 0

Quindi non contiene alcun match contiguo di P. Conterrebbe invece un match non contiguo in cui  $P[0]=0$  viene trovato nel secondo elemento della V-fetta 0,  $P[1]=1$  viene trovato nel terzo elemento della V-fetta, poi  $P[2]=P[3]=2$  sono trovati nel settimo e ottavo posto della V-fetta e infine  $P[4]=0$  viene trovato nell'ultimo elemento della V-fetta. L'esercizio richiede di considerare solo match contigui di P e quindi quello appena descritto non va considerato. Quindi il programma richiesto per la V-fetta 0 dovrà scrivere su "output": "V-fetta 0= 0 match".

La V-fetta 1 è: 2 0 1 2 2 0 1 2 2 0

Ci sono 2 match contigui. Il primo inizia nel secondo elemento della V-fetta 1 e si estende a comprendere il sesto elemento. Il secondo match invece parte dal sesto elemento della V-fetta e si estende fino alla fine. Quindi i 2 match hanno il sesto elemento in comune e quindi sono sovrapposti tra loro. Visto che nel problema vanno contati solo match non sovrapposti, la V-fetta 1 ha un solo match e quindi il programma deve stampare la frase: "V-fetta 1=1 match".

Va osservato che non tutte le V-fette hanno 10 elementi definiti (escludendo i -1), come le prime due. Le V-fette di indice 3 e 4 hanno 9 elementi definiti. Per loro il matching andrà fatto solo su questi 9 elementi.

E' molto importante osservare che i match vanno effettuati sulle V-fette scordando il fatto che esse sono costituite da colonne che risiedono su strati diversi di Y. Per esempio, per la V-fetta 1, il primo match inizia nello strato 0 e finisce nello strato 1. In teoria un match potrebbe estendersi anche su più di 2 strati.

Si consiglia di usare una funzione che sia capace di scorrere una qualsiasi V-fetta contando i match (contigui e non sovrapposti) che vi si trovano e che usa un'altra funzione che cerca se, a partire da un certo punto della V-fetta, esiste un match contiguo e completo.

**Avvertenza:** dichiarare un array `int Y[lim1][lim2][lim3]` e copiarci dentro i valori letti in X è considerata una soluzione sbagliata.

**Correttezza.** Per ogni funzione introdotta specificare pre e post-condizione. Dimostrare che una delle funzioni significative del vostro programma è corretta rispetto alla pre e post-condizione specificata.

Scrivere l'invariante e la post-condizione del ciclo del main che considera le V-fette e dimostrare la correttezza del ciclo.