

Esercizio 1 del 6/4/2016

Viene dato un main che esegue le seguenti operazioni:

1) dichiara un array `int X[400]` e lo riempie di 0, poi dichiara `int P[50]`;

2) legge da cin 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) legge `n_el` ($0 < n_el \leq 400$) e poi `n_el` valori che metterà in `Y` per strati (che è lo stesso che leggerli nelle prime `n_el` posizioni di `X`);

iii) legge `dimP` (che assumiamo soddisfi $0 < dimP \leq 50$) e poi legge `dimP` valori in `P`.

3) il main contiene un ciclo che invoca 2 funzioni che costituiscono l'esercizio da fare. Si tratta della funzione "`fetta_vuota`" e "`match_fetta`" che sono descritte nel seguito.

la funzione "`fetta_vuota`" ha il seguente prototipo:

```
bool fetta_vuota(int f, int n_el)
```

e la sua PRE e POST sono come segue:

PRE=($0 \leq f < lim3$, $n_el > 0$)

POST=(restituisce true sse la V-fetta `f` in un array `Y[lim1][lim2][lim3]` che contiene `n_el` elementi inseriti per strati non contiene alcun elemento definito)

La funzione "`match_fetta`" risponde alle seguenti specifiche:

PRE=($0 \leq f < lim3$, e la V-fetta `f` di `Y[lim1][lim2][lim3]` con `n_el` valori inseriti per strati contiene qualche elemento definito)

```
void match_fetta(int* X, int lim1, int lim2, int lim3, int n_el, int f, int*P, int dimP)
```

POST=(se sulla V-fetta `f` di `Y[lim1][lim2][lim3]` ci sono `n` match ($n \geq 0$) di `P` contigui e non sovrapposti, allora stampa su cout la frase "`V-fetta f = n match`")

Cosa significhi che si devono considerare match **contigui e non sovrapposti** viene spiegato nell'esempio che segue.

Esempio: assumiamo che $\text{lim1}=3$, $\text{lim2}=4$, e $\text{lim3}=5$ 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	

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

La V-fetta 0 di Y consiste dei seguenti 10 valori:

1 0 1 1 0 1 2 2 1 0

E' facile vedere che questa V-fetta non contiene alcun match contiguo di P. Conterebbe invece il seguente match non contiguo: $P[0]=0$ 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 come le prime tre. Le V-fette di indice 3 e 4 hanno 9 elementi definiti. Per loro il matching andrà fatto solo su questi 9 elementi.

Attenzione: 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 strutturare match_strato in modo che scorra la V-fetta f dal primo elemento in avanti e che, per ciascun elemento invochi un'altra funzione che determina se, a partire da quel punto della V-fetta, esiste un match contiguo e completo di P.

Attenzione: 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 specificarele sue pre e post-condizioni. 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.