

Esercizio 4 del 28/10/2014

1) Il programma deve dichiarare 2 array di interi, $T[100]$ e $P[10]$, leggere 2 interi dimT e dimP , ($0 < \text{dimT} \leq 100$ e $0 < \text{dimP} \leq 10$) e poi legge dimT interi in T e dimP interi in P . Successivamente il programma deve determinare quanti match di P esistono in T . Un match di P in T è una **porzione** $T[i..i+\text{dimP}-1]$ di lunghezza dimP di T tale che questa porzione è identica a P .

Esempio 1. Sia $\text{dimT}=16$, $\text{dimP}=4$ e $T=[0,1,1,1,1,2, 0,1,0,1,1,1,1,2,0,0]$ e $P=[1,1,1,2]$, allora ci sono 2 match di P in T ed essi iniziano dalle posizioni 2 e 10 di T .

Ancora con $\text{dimT}=16$ e $\text{dimP}=4$, siano $T=[0,1,2,1,2,1, 2,1,0,1,1,2,1,2,1,2]$ e $P=[1,2,1,2]$. Allora ci sarebbero 4 match di P in T che iniziano nelle posizioni 1, 3, 10 e 12. Si osservi che i match sono a 2 a 2 sovrapposti tra loro. Per esempio, il match che inizia in 1 comprende le posizioni 1,2,3 e 4 di T mentre il match che inizia in 3 comprende le posizioni 3,4,5 e 6 di T . Quindi i 2 match hanno le posizioni 3 e 4 in comune e per questo si dicono sovrapposti.

La pre- e postcondizione del programma da realizzare sono le seguenti:

PRE=(cin contiene $0 < \text{dimT} \leq 100$ e $0 < \text{dimP} \leq 10$, seguite da dimT interi e poi ancora dimP interi)

POST=(il programma stampa su cout il numero di match di P in T)

2) Si tratta di una variante dell'esercizio precedente in cui si vogliono contare i match non sovrapposti di P in T .

Esempio 2. Consideriamo nuovamente i 2 casi dell'esempio 1. Nel primo caso la richiesta di considerare match non sovrapposti non cambia il numero di match che resta 2, visto che i 2 soli match presenti non sono sovrapposti tra loro. Tutto cambia invece nel secondo caso in cui i 4 match sono a 2 a 2 sovrapposti e quindi, contando solo match non sovrapposti, abbiamo solo 2 match anziché 4 come prima.

La PRE è uguale al caso (1) e la POST richiede il numero dei match non sovrapposti.