

INFORMAZIONI IMPORTANTI:

I programmi richiesti devono leggere l'input da un file input1, input2, ecc. e devono scrivere l'output su un file output1, output2, ecc. I valori sui file di input e di output sono sempre separati da uno spazio. In ogni esercizio sarà descritto in dettaglio cosa vada assunto sul contenuto del file di input e cosa vada scritto su quello di output. Il file che contiene la soluzione dell'esercizio (1) deve chiamarsi esercizio1.cpp e quello del (2) deve chiamarsi esercizio2.cpp. Per ogni esercizio trovate nella vostra home un file di input ed un corrispondente file di output. Si tratta solo di esempi. Dovreste provare i vostri programmi anche con altri file di input. I vostri programmi dovrebbero compilare e per gli input dati dovrebbero produrre output uguali a quelli dati.

ESERCIZI:

Dovete cercare di scrivere (come commenti dei programmi) pre- e postcondizioni per ogni programma e invarianti per ogni ciclo.

1) Dati 2 array $\text{char } T[\text{dim}]$ e $\text{char } P[\text{dimP}]$ un match di P in T è una sottosequenza $T[i..j]$ di T tale che $0 \leq i \leq j < \text{dim}$ e tale che $T[i..j] = P[0..\text{dimP}-1]$. Si tratta di scrivere un programma che dichiari $\text{char } T[100]$, $P[20]$ e poi legga da input1 le seguenti cose: 2 interi dim e dimP , seguiti da $\text{dim} + \text{dimP}$ caratteri che vanno messi in $T[0..\text{dim}-1]$ e in $P[0..\text{dimP}-1]$. Si deve assumere che $1 \leq \text{dim} \leq 100$ e che $1 \leq \text{dimP} \leq 20$. Dopo, il programma deve cercare un match di $P[0..\text{dimP}-1]$ in $T[0..\text{dim}-1]$. In caso che un tale match esista, il programma deve scrivere su output1 il valore true (che è 1) e l'indice di T nel quale inizia il match trovato. Se ci fossero diversi match allora va considerato quello con inizio minimo. Se invece non c'è alcun match, il programma deve scrivere false (cioè 0) su output1.

Esempio: Sia $T=[a,c,b,a,c,a]$ e $P=[a,c]$ allora il programma deve scrivere su output1, true e 0 che è l'inizio del match che inizia prima.

2) Il programma richiesto è simile a quello di (1). Deve riempire T e P esattamente come in (1) ma leggendo i valori dal file input2. Dopo il programma deve calcolare il numero di match di $P[0..\text{dimP}-1]$ in $T[0..\text{dim}-1]$ che non sono sovrapposti tra loro. Due match $T[i_1..j_1]$ e $T[i_2..j_2]$ non sono sovrapposti se $i_2 > j_1$ oppure $i_1 > j_2$. Il programma deve scrivere su output2 il numero di match non sovrapposti che ci sono.

Esempio: Si consideri l'esempio precedente. Con quei dati, il programma di questo esercizio deve scrivere su output2 il valore 2 visto che ci sono 2 match non sovrapposti. Se invece $T=[a,a,b,a,a,a]$ e $P=[a,a]$, allora il numero dei match non sovrapposti è 2 e il programma dovrebbe quindi (di nuovo) scrivere 2 su output2. Osserva che ci sarebbero 3 match ma gli ultimi 2 sono sovrapposti alla posizione 4.

NOTARE che gli esempi forniscono contenuti per file di input/output che possono servire per controllo.