

Corso di programmazione 2010-2011

Quarta esercitazione per casa: assegnata l'11 febbraio 2011

Consegna: il 17 febbraio entro le 10 di mattina, con il comando: **consegna settimana4**

Ogni funzione deve avere la sua pre- e postcondizione. Per ciascuno dei 2 esercizi, la funzione di maggiore importanza deve venire corredata con la dimostrazione della sua correttezza rispetto alla relativa pre- e postcondizione.

Introduzione: in un problema di pattern matching abbiamo un testo T , per esempio $\text{char } T[1000]$, ed un pattern P (per esempio, $\text{char } P[\text{dim_P}]$) ed esaminiamo T da sinistra a destra cercando una porzione di dim_P elementi contigui di T che siano identici agli elementi di P . Più precisamente, cerchiamo un k tale che $T[k..k+\text{dim_P}-1] == P[0..\text{dim_P}-1]$. Quando questo si verifica diremo di avere trovato un **match** di P in T e diremo che questo match **comincia nella posizione k di T** . Due match di P in T , che iniziano nelle posizioni k_1 e k_2 di T , sono **sovrapposti**, se gli intervalli $[k_1, k_1+\text{dim_P}-1]$ e $[k_2, k_2+\text{dim_P}-1]$ hanno qualche indice in comune e altrimenti sono detti **non sovrapposti**.

(1) Si chiede di scrivere una funzione $\text{int } F(\text{char } (*T)[15][15], \text{int } \text{limite1}, \text{char } P[], \text{int } \text{dim_P})$ che soddisfi i seguenti punti:

(i) limite1 è il limite della prima dimensione di T , cioè T è un array $\text{char } T[\text{limite1}][15][15]$ ed è completamente definito; inoltre $\text{dim_P} > 0$ e $P[0..\text{dim_P}-1]$ è definito.

(ii) F deve calcolare l'indice minimo i tale che lo strato i -esimo $[15][15]$ di T contenga almeno 2 match di P non sovrapposti.

Il seguente esercizio è opzionale e non banale

(2) Un array a tre dimensioni come $\text{char } T[\text{limite1}][15][15]$ lo si può vedere come una torta di limite1 strati $[15][15]$. Possiamo "tagliare" una tale torta in fette, dove la prima fetta è costituita dalle prime righe di tutti gli strati, la seconda fetta è costituita dalle seconde righe di tutti gli strati e così via fino alla quindicesima fetta.

Si tratta di scrivere una funzione $\text{bool } G(\text{char } (*T)[15][15], \text{int } \text{limite1}, \text{char } P[], \text{int } \text{dim_P})$ per i cui parametri vale quanto detto in (1) (i) e che soddisfi la seguente richiesta:

G deve restituire **true** sse esiste almeno una fetta di T che contiene un match di P

NOTA: è consentito definire anche funzioni ausiliarie oltre F e G .