Esercizio 1 del 2/4/2019

Scopo di questo esercizio è scrivere un programma che elenchi tutte le occorrenze di un pattern in un testo. Il testo viene rappresentato come un array bidimensionale char T[100][100] dove ogni riga dell'array contiene una riga del testo. Le righe del testo sono di lunghezza variabile e terminano con il carattere speciale '\0' (carattere con codice ASCII uguale a 0, da non confondersi con la cifra '0'). I caratteri che compongono il testo possono essere lettere o spazi.

Il pattern è una sequenza di char di lunghezza variabile che termina con il carattere '\0', e puo` contenere lettere, spazi e il carattere '?' che rappresenta un carattere "jolly" che fa match con "qualsiasi carattere". Il pattern viene memorizzato nell'array char P[100].

Diremo che il pattern di lunghezza l, P[0…l-1] occorre nella riga i di T a partire dall'indice j, se P[0..l-1]=T[i][j..j+l-1]. Sia in T che in P non si deve mai considerare il carattere '\0'.

Viene dato un main che effettua la lettura di P e T, ed inserisce il numero di righe di T nella variabile int n_righe. Per l'input il main usa la funzione getline che non è stata vista nel corso e che legge un'intera riga di input, aggiungendo il carattere '\0' alla fine. La definizione della funzione si trova all'indirizzo: http://www.cplusplus.com/reference/istream/istream/getline/.

Il programma deve produrre tutte le occorrenze del pattern nel testo, ordinate per riga e poi per colonna. Ogni occorrenza i,j viene stampata nella forma riga: i colonna: j. Se non c'`e nessuna occorrenza del pattern, il programma stampa Pattern non trovato.

Si chiede di risolvere l'esercizio implementando la funzione bool match(char* P, char* S) che ritorna true se e solo se il pattern P occorre nell'array lineare S a partire dalla posizione 0. Si deve assumere che P ed S terminano con il carattere speciale '\0'.

Per esempio, se il pattern è P = aba\0:

• la chiamata match(P, S) con S = ababbac\0 ritorna true;

• la chiamata match(P, S) con S = bbbabac\0 ritorna false (l'occorrenza aba non `e in posizione 0 di S);

• la chiamata match(P, S) con S = ab\0 ritorna false (S è più corto del pattern).

Correttezza: scrivere un invariante per tutti i cicli della funzione match e dimostrarne la correttezza rispetto alla pre- e post-condizione seguenti :

PRE: P è un pattern che termina con '\0', S è una sequenza di char che termina con '\0'

POST: la funzione ritorna true se e solo se P occorre in S a partire dalla posizione 0

Esempio: dato il pattern P = ?el ed il testo T di 3 righe che segue

nel mezzo del cammin di nostra vita

mi ritrovai per una selva oscura

che la diritta via era smarrita

il programma produce l'output

riga: 0 colonna: 0

riga: 0 colonna: 10

riga: 1 colonna: 20