Esercizio 1 del 28/4

Consegnare corretto entro 4/5/2015

Si tratta di realizzare un semplice pattern matching, ma con funzioni ricorsive.

Quindi come al solito il main deve aprire i file "input" e "output", dichiarare int X[400] e int P[20], mettere tutti i loro elementi a 0, leggere n_el (0<n_el<=400) e leggere n_el in X, poi leggere dimP (0<dimP<=20) e leggere dimP valori in P. Dopo questo il programma deve cercare i match di P[0..dimP-1] in X[0..dim-1], considerando match contigui ed anche eventualmente sovrapposti. Quindi vanno considerati sia match non sovrapposti che sovrapposti. Insomma tutti i match che ci sono.

Esempio: supponiamo che X=[2,3,2,3,2,3,2,2,3,0,1,2,3] e P=[2,3,2], allora il pattern P ha 3 match in X: a partire dalla posizione 0, a partire dalla posizione 2, a partire dalla posizione 4 e basta. I 3 match condividono un elemento e quindi sono tutti parzialmente sovrapposti. In un caso come questo il vostro programma deve stampare su "output" le seguenti stringhe:

match n.1 a partire dalla posizione 0

match n.2 a partire dalla posizione 2

match n.3 a partire dalla posizione 4

fine

Se invece il pattern fosse P=[2,3], allora con lo stesso X, l'output da produrre sarebbe:

match n.1 a partire dalla posizione 0

match n.2 a partire dalla posizione 2

match n.3 a partire dalla posizione 4

match n.4 a partire dalla posizione 7

match n.5 a partire dalla posizione 11

fine

Per realizzare le operazioni richieste il main deve invocare una funzione ricorsiva:

PRE=(0<=dim, 0<=dimP, X[0..dim-1] e P[0..dimP-1] sono definiti, i>=0, count_match>=0)

void match(int *X, int*P, int dim, int dimP, int i, int count_match, ofstream & OUT)

POST=(trova tutti i match di P[0..dimP-1] in X[i..dim-1] ed esegue le corrispondenti stampe dove il numero del match parte da count_match, poi count_match+1, ecc.)

Consiglio: conviene seguire lo schema già usato in precedenza per il pattern matching e che ora va coniugato ricorsivamente: la funzione ricorsiva match esamina tutti gli elementi di X e, per ciascuno di essi, invoca una seconda funzione (pure ricorsiva) che tenta il match a partire da quell'elemento.

Correttezza: scrivere Pre e Post della funzione ausiliaria. Tentare una dimostrazione induttiva della sua correttezza.