Esercizio 4 del 26/3/2013 da consegnare corretto entro il 4/3 a mezzanotte

Viene dato un programma con la definizione della struttura nodo, una funzione che stampa liste concatenate e un main che compie varie operazioni sui file "input" e "output" e che invoca una funzione nodo* crea(int dim, ifstream & INP). La funzione crea non è data e scriverla è parte dell'esercizio. Essa deve costruire una lista concatenata con nodi che hanno come campo informativo interi letti da "input".
Più precisamente, la funzione crea deve soddisfare la seguente pre- e post-condizione:
PRE_crea=("input" contiene (almeno) dim interi a_1,..,a_dim, dim>=0)
POST_crea=(la funzione restituisce una lista corretta di dim nodi tali che il primo nodo contiene nel campo info a_1, il secondo nodo contiene a_2, e l'ultimo contiene a_dim)

Esempio: *se dim=3 e si leggono gli interi 2, 3 e 4, crea deve costruire e restituire col return una lista (corretta) di 3 nodi con campi informativi 2, 3 e 4 (dove 2 è il campo info del primo nodo delle lista e 4 quello dell'ultimo).*

Dopo queste operazioni, il main legge da "input" l'intero dimP (0<dimP<=20) e legge i successivi dimP interi nell'array int P[20]. A questo punto il main invoca una funzione nodo* F(nodo*L, int*P, int dimP). La scrittura di F è la parte principale dell'esercizio. F deve essere una funzione ricorsiva e il suo compitoè di cercare un match **contiguo e completo** di P nella lista L e, qualora lo trovi, di eliminare da L i nodi del match e restituire col return quello che resta di L. I nodi eliminati da L devono venire deallocati. Spieghiamo con un esempio in cui introduciamo anche notazioni utili per scrivere POST_F:

Esempio: *supponiamo che L= 2->3->3->1->4->2->2 e che P=[3,1,4] con dimP=3. C'è un match di P in L che è costituito dai nodi in terza, quarta e quinta posizione. Quindi la funzione F dovrebbe deallocare questi 3 nodi e restituire con il return quello che resta di L e cioè: 2->3->2->2. Questa lista rimanente la indichiamo con (L-P). Se P=[1,4,5], non ci sarebbe alcun match in L e quindi (L-P)=L. Se P=[2,3,3,1,4,2,2], allora (L-P) sarebbe la lista vuota.*

Precisamente, la funzione F deve soddisfare la seguente pre- e post-condizione.

PRE_F=(L è lista corretta e L=vL, P ha dimP elementi definiti con dimP>=0)

nodo* F(nodo*L, int*P, int dimP)

POST_F=(F restituisce col return (vL-P))

Dopo l'invocazione di F, il main stampa la lista restituita da F invocando la funzione stampa che è data. Nel caso limite in cui (L-P) è la lista vuota, il main scrive su "output" la frase, "(L-P) vuota".

Correttezza: Si richiede la dimostrazione induttiva di correttezza della funzione F. Se vengono usate funzioni ausiliarie (cosa consigliata), è necessario definire pre- e post-condizioni per ciascuna di esse. Le funzioni devono essere tutte ricorsive.

Attenzione: F non deve creare alcun nuovo nodo.