

Esercizio 2

Questo esercizio è una naturale estensione dell'esercizio 1 di questa settimana. Di nuovo abbiamo una lista L e un intero x e questa volta vogliamo eliminare da L tutti i nodi con info=x e produrre con i rimanenti nodi di L una nuova lista L1 . I nodi tolti da L devono venire deallocati e L1 deve essere composta con i nodi di L.

Esempio: se L = 2 -> 1 -> 1 -> 3 -> 1 -> 0 -> 2 e x=1, allora L1 = 2 -> 3 -> 0 -> 2.

Come per l'esercizio 1, vogliamo 3 funzioni che facciano quanto richiesto seguendo i 3 approcci già visti nell'esercizio 1 e a lezione.

I prototipi delle 3 funzioni sono come segue:

- 1) `nodo* elimAll(nodo*L, int x)`
- 2) `void elimAll2(nodo*L, int x)`
- 3) `void elimAll3(nodo*&L, int x)`

I test richiedono che il Vostro programma stampi 3 volte la stessa lista, ottenuta con le 3 funzioni richieste. Come per l'esercizio 1, il main produce 3 liste L, L2 ed L3, da usare rispettivamente, come input delle 3 funzioni elimAll, elimAll2, elimAll3.

Correttezza; scrivere PRE e POST per ciascuna delle 3 funzioni da fare.