Compito di Programmazione del 29 marzo 2012

Teoria:

(i)Data la dichiarazione char X[4][8][10], qual'è il tipo di (*X)[-2] e quale dimensione ha l'oggetto puntato da questo puntatore? Inoltre che valore ha l'espressione (*X)[-2] rispetto al valore di X?
(ii) Spiegare cos'è l'overloading (sovraccaricamento) ed i principi dell'overloading resolution. (max 4 righe).

Programmazione: Sia per la programmazione iterativa che per quella ricorsiva, il problema da considerare è quello di smembrare una lista L data nel modo seguente:

Esempio. Sia L = 3->2->3->2->0->2->1->2, e sia A un array di interi [2,1,3] allora L va vista come L0@L1@L2@R, con L0=3->2, L1=3, L2=2->0->2, e R=1->2. Il numero di nodi di L0, L1 ed L2 è uguale ad A[0], A[1] ed A[2], rispettivamente. Vogliamo che L diventi L0@L2 e che i nodi di L1 e di R siano deallocati. Se invece A=[2,3,1,1], allora L0, L1, L2 sarebbero come prima, L3 avrebbe solo il nodo 1 ed R solo il nodo 2. Quindi L deve diventare L0@L2@R mentre L1 ed L3 devono venire deallocate. Si noti che R resta in L quando il numero di elementi di A è pari e viene invece deallocato quando è dispari.

Esercizio iterativo: la funzione iterativa da realizzare deve avere il seguente prototipo:
void F(nodo*L, int *A, int lim), L è la lista da smembrare e A[0..lim-1] è l'array da usare per smembrare L nel modo descritto nell'esempio precedente.

F deve soddisfare le seguenti tre condizioni:

a) deve soddisfare alle seguenti PRE e POST:

PRE=(A[0..lim-1], contiene valori positivi, L è una lista corretta che contiene un numero di nodi maggiore o uguale di A[0]+A[1]+...A[lim-1], lim>=0, L=L0@L1@...L(lim-1)@R, dove ciascun Li ha numero di nodi pari ad A[i]).

POST=(F modifica L nel modo seguente: se lim è dispari allora L diventa L0@L2@..L(lim-1), se invece lim è pari, allora L diventa L0@L2@..L(lim-2)@R, mentre i nodi di L1, L3,.. e nel primo caso di R, vengono deallocati).

b) F deve usare una funzione iterativa nodo* taglia(nodo* L, int k) che soddisfa la seguente coppia di PRE e POST:

PRE_taglia=(L è una lista corretta che contiene almeno k nodi, k>0)

POST_taglia=(restituisce il k-esimo nodo della lista L)[1]

c) F deve anche usare una funzione iterativa void del (nodo*L) che soddisfa la seguente coppia di PRE e POST: PRE_del=(L è una lista corretta), POST_del=(i nodi di L sono deallocati).

Si richiede di scrivere le funzioni F, taglia e del. **Ogni ciclo deve avere un corrispondente invariante.**

NOTE: (i) non si deve creare alcun nodo nuovo. I nodi della lista originale L vanno in parte conservati ed in parte distrutti.

(ii) Si osservi che la PRE implica che A[0]>0 e quindi il primo nodo della nuova lista sarà lo stesso di quello della lista originale. Non ci si deve quindi preoccupare di restituire al chiamante l'inizio della nuova lista, ma solo di collegare correttamente tra loro le sue parti.

---

[1] Quindi se k=1 taglia dovrà ritornare L, se k=2 dovrà ritornare il successore di L e così via.

(iii) Conviene ad ogni iterazione cercare di determinare, con la funzione taglia, due successivi pezzi di L, il primo da tenere ed il secondo da deallocare con del.

Esercizio ricorsivo: si chiede di scrivere una funzione ricorsiva void FR(nodo*L, int*A, int lim) che soddisfa la stessa PRE e POST della parte iterativa. La funzione FR deve usare le due funzioni ausiliarie taglia e del, specificate nella parte iterativa. Non è richiesto di dare versioni ricorsive di queste funzioni. Si possono usare quelle iterative. Le NOTE della parte iterativa sono utili anche per questa parte.

Si **richiede la prova induttiva della correttezza della funzione F rispetto alla PRE e POST.**