

Esame Scritto di Programmazione del 5/9/2018 durata esame = 2 ore

Attenzione: se la vostra consegna **NON** passa i 4 test, allora **SICURAMENTE** ha qualcosa di sbagliato, se invece li passa, non dovete **ASSOLUTAMENTE** considerarla una garanzia della sua correttezza. Quando qualcuno dei test fallisce, il moodle mostra l'input solo del primo test che fallisce.

Data una lista concatenata L ed un array A che contiene un numero pari di interi non negativi, si vuole smembrare L per creare 2 liste $L1$ ed $L2$ nel modo illustrato dall'esempio seguente.

Esempio: Data la lista concatenata $L = 3 \rightarrow 1 \rightarrow 2 \rightarrow 20 \rightarrow -11$, se $A = [2, 2, 1, 3]$, L verrà scomposta nelle 2 liste $L1 = 3 \rightarrow 1 \rightarrow -11$ e $L2 = 2 \rightarrow 20$ che sono spiegate come segue: $A[0]=2$ indica che i primi 2 elementi di L vanno in $L1$, poi $A[1]=2$ indica che il terzo e il quarto elemento di L vanno in $L2$, infine $A[2]=1$ indica che l'ultimo elemento di L va in $L1$, ed, essendo finito L , l'ultimo valore (3) di A non viene usato.

Vediamo un secondo esempio: per $L = 3 \rightarrow 1 \rightarrow 2 \rightarrow 20 \rightarrow -11 \rightarrow 4 \rightarrow 0 \rightarrow 5$, se $A = [0, 2, 1, 3]$, le due liste saranno $L1 = 2 \rightarrow 0 \rightarrow 5$ e $L2 = 3 \rightarrow 1 \rightarrow 20 \rightarrow -11 \rightarrow 4$. Essendo $A[0]=0$ e $A[1]=2$, i primi 2 elementi di L vanno in $L2$, poi con $A[2]=1$, il terzo elemento di L va in $L1$, e con $A[3]=3$, i successivi 3 elementi di L vanno in $L2$ e finalmente, visto che A è vuoto mentre L ha ancora 2 nodi, questi ultimi 2 nodi di L vanno in $L1$.

Se $L = 3 \rightarrow 1 \rightarrow 2 \rightarrow 20 \rightarrow -11$ e $A = [0, 2, 1, 3, 2, 2]$, allora $L1 = 2$ e $L2 = 3 \rightarrow 1 \rightarrow 20 \rightarrow -11$. In questo caso si osserva che, dopo aver inserito i primi 2 nodi di L in $L2$ e il terzo in $L1$, visto che in L rimangono solo 2 nodi, essi vengono messi in $L2$ perché $A[3]=3 > 2$. Dopo di che, essendo L finita, i successivi elementi di A vengono ignorati.

Nel caso seguente, $L = 3 \rightarrow 1 \rightarrow 2 \rightarrow 20 \rightarrow -11$ e $A = [2, 1, 4, 2]$, avremo $L1 = 3 \rightarrow 1 \rightarrow 20 \rightarrow -11$ e $L2 = 2$. Dopo aver inserito i primi 2 nodi in $L1$ e il terzo in $L2$, L contiene ancora 2 nodi che verranno inseriti in $L1$ visto che $A[2]=4 > 2$. Dopo di che, essendo L finita, $A[3]=2$ viene ignorato.

Esercizio ricorsivo:

Realizzare una funzione ricorsiva **Fric** che soddisfa le seguenti specifiche:

PRE=(*lista(L), lista(L1), e lista(L2) sono ben formate, A contiene dimA elementi non negativi, con dimA pari >=0, vL=lista(L), vL1=lista(L1), vL2=lista(L2)*)

void Fric(nodo*L, int*A, int dimA, nodo*&L1, nodo*&L2)

POST=(*i nodi di vL sono distribuiti correttamente su 2 liste X1 e X2 secondo i valori di A e L1=vL1@X1 e L2=vL2@X2*)

Con @ si denota la concatenazione tra 2 liste.

Consiglio: conviene usare (almeno) 2 funzioni ausiliarie ricorsive come segue:

- i) una funzione ha lo scopo di individuare la porzione di L da "staccare" per poi aggiungerla ad $L1$ o ad $L2$;
- ii) l'altra funzione deve concatenare 2 liste

Esercizio iterativo:

Realizzare una funzione iterativa **Fiter** che soddisfi le stesse specifiche di **Fric**. Per **Fiter** vale lo stesso consiglio dato per **Fric** ma, naturalmente, le funzioni ausiliarie dovranno essere iterative.

Correttezza:

- a) Scrivere la dimostrazione induttiva di **Fric**.
- b) Scrivere l'invariante del ciclo principale di **Fiter**.