

Esercizio 1, I parte

Chiamiamo di Fibonacci una lista concatenata di interi in cui gli elementi della lista, a partire dal primo, contengono i numeri della sequenza di numeri Fibonacci F_n, F_{n-1}, \dots, F_1 , dove n è la lunghezza della lista,

Scrivere una funzione ricorsiva che riceve come input una lista concatenata e verifica se si tratta di una lista di Fibonacci.

Facoltativo. Discutere se la funzione scritta è tail recursive.

Nota. Si ricorda che la sequenza di Fibonacci è definita da:

$$F_0 = 0 \quad F_1 = 1 \quad F_{n+2} = F_{n+1} + F_n$$

Esempio: la lista concatenata : 8 -> 5 -> 3 -> 2 -> 1 -> 1 -> NULL è la lista di Fibonacci con 6 elementi.

Sol.

```
int ListFib(ListaPtr L);
/*postc: restituisce 1 se la lista è una lista di Fibonacci, 0 altrimenti */

int ListFib(ListaPtr L)
/*postc: restituisce 1 se la lista è una lista di Fibonacci, 0 altrimenti */
{int k;
if (!L) return 0;
if (L->next == NULL)
if ( L->elem == 1) return 1; else return 0;
if (L->next ->next == NULL )
if ( L->elem == 1 && L->next ->elem == 1) return 1; else return 0;
if (L -> elem == L->next -> elem + L->next ->next ->elem )
return ListFib(L ->next); else return 0;}
```

Es 1, II parte

Si scriva una funzione **int meFoglie (TreePtr t)**; che dato un albero binario t, contenente interi positivi, restituisca la media dei valori memorizzati nelle foglie dell'albero.

Si possono usare funzioni ausiliarie richiamate dalla funzione meFoglie. E' richiesto però che la media venga calcolata tramite un'unica visita dell'albero.

Ricordatevi di indicare pre e post condizioni.

Sol.

```
int meFoglie(TreePtr t);
/*prec: l'albero contiene interi positivi && t!= NULL;
postc: restituisce la media dei valori nelle foglie, 0 se l'albero e' vuoto*/

int meFoglieAus(TreePtr t, int *sum );
/*postc restituisce il numero delle foglie e ne calcola in sum la somma dei valori
*/

int meFoglie(TreePtr t)
/* prec: l'albero contiene interi positivi && t!= NULL;
postc: restituisce la media dei valori nelle foglie, 0 se l'albero e' vuoto*/
{int nF,sum=0;
if (!t) return 0;
nF = meFoglieAus(t,&sum);
return sum/nF;}

int meFoglieAus(TreePtr t, int *sum )
/*prec: t!= NULL;
postc: restituisce il numero delle foglie e ne calcola in sum la somma dei valori
*/
{int nF = 0;
if (foglia(t)) {*sum += t->elem; nF++;}
else nF = meFoglieAus(t->left,sum) + meFoglieAus(t->right,sum);
return nF;}
```