Laboratorio di Programmazione
Lezione 11

Esercizio 1

- Si definisca una funzione **left-rot** per ruotare a sinistra con rientro gli elementi di una lista. Esempio:

  ```
  (left-rot '(1 0 0 1 1 0)) -> (0 0 1 1 0 1)
  ```

- Si sfrutti la **left-rot** per scrivere una funzione che consenta un numero 'arbitrario' di rotazioni a sinistra.

Esercizio 2

Si scriva un programma **sum-binary** per sommare due numeri binari passati come liste (senza sfruttare la funzione di libreria **reverse**).

Esercizio 3

Si implementino mediante le liste le seguenti funzioni per *Alberi Binari*.

```
(leaf-tree <item>)              : nodo -> albero
(leaf? <tree>)                  : albero -> booleano
(grow-tree <item><tree><tree>) : nodo x albero x albero -> albero
(root-node <tree>)              : albero -> nodo
(left <tree>)                   : albero -> albero
(right <tree>)                  : albero -> albero
```

Esercizio 4

Si scriva una funzione **tree->inexp** per ottenere espressioni infisse (in forma di stringhe) mediante una visita simmetrica dell'albero di valutazione.

Esercizio 5

Si scriva una funzione **tree->rpn** per ottenere espressioni RPN (in forma di stringhe) mediante una visita posticipata dell'albero di valutazione.

Esercizio 6

Si scriva una funzione **rpn->tree** che trasformi espressioni RPN (passate come stringhe) in alberi binari.