Exercises on LISP
Exercises

- Define a function \texttt{conta} that counts the occurrences of an element in a list.
- Define a function \texttt{naturali} that, given a number \( n \), returns an ordered list with the first \( n \) natural numbers.
- Define a function \texttt{ribalta} that returns the reverse of a list.
- Define a function \texttt{inserisci} that inserts a number in an ordered list.
- Given a list of integer, define a function \texttt{quadrati} that returns a list with their squares.
- Given a list of elements, define a function \texttt{sonoListe} that returns a list of Booleans that tell whether each element is a list.
- Given a list of groups of people and a list of couples of people, define a function \texttt{gruppi} that returns whether couple(s) are present in one or more groups.
Exercises

- A tree can be represented as:
  - NIL, or
  - (x left right)
  - Where x is the root and left and right are two trees
- Define a function `creaAlbero` that creates and returns a random tree
- Define a function `profondita` that returns the depth of the tree
- Define a function `stampaDF` that prints the elements of the tree in depth-first order
- Define a function `contaNodi` that counts the number of nodes in the tree