Part 1

- The goal is to write **tail-recursive** functions 1) that filters the list ‘filterList’. As a result of the first function a list with elements that satisfy predicate should be returned.
  In example:
  filterList(List("Hello", "there", "people"), _.contains('l'))
  would return list with two elements contains **List(“Hello”, “people”)**.
  2) the goal is to write the function that inverts the order of the elements in the list (produces another list).

PART 2:
SIMPLE CLASS FOR BOOKKEEPING

- The goal is to write few classes that would allow us to do personal accounting: the Bookkeeping, Expense and Income
- Expense should contain the amount and description,
- Income should be largely the same but would also contain date
  - Above two classes have to be “functional”
  - They should have a method converting them to string of the form:
    +number - - - “description”
    -number - - - “description” with quotes
- The object of class Bookkeeping should allow to add objects of both, Expense, and Income (they need to be kept separate internally)

  ```
  val bk = new Bookkeeping()
  bk -= Expense(20, “lunch”)
  bk -= Expense(200, “jacket”)
  bk += Income(1000, “subsistence”, “10 December”)
  ….
  ```
- There should be methods in Bookkeeping to print all expenses & incomes as well as obtain the balance = sum of incomes - sum of expenses
- There are 2 points to win for Expense/Income classes with all required functionality, and 2 points for the class Bookkeeping