Homework: DefineLang and FuncLang

Learning Objectives:

1. Write programs in DefineLang, FuncLang
2. Get familiar with the concepts of recursive functions, high order functions

Instructions:

- Total points: 41 pt.
- Early deadline: Sept 26 (Wed) 2018 at 6:00 PM; Regular deadline: Sept 28 (Fri) 2018 at 6:00 PM.
- Write your Definelang and Funclang programs for the following questions and submit them in one pdf file. Use your Funclang interpreter provided in hw4code.zip to test their correctness.
- Follow the steps in the tutorial of the homework 2 to setup the interpreter.
- How to submit:
  - Submit your document to Canvas under Assignments, Homework 4.
  - Please provide the complete solutions in one pdf file.
  - You can write your solutions in latex or word and then convert it to pdf; or you can submit a scanned document with legible handwritten solutions.

Questions:

1. (3 pt) Write a DefineLang program to do the following: define the ASCII values for the letters 'a' 'b' and 'c'; write DefineLang expressions to compute the ASCII value for the corresponding upper case letter.

2. (3 pt) Pythagoras' theorem is a relation between three sides of a right angle triangle. It states that in a right angle triangle (a triangle whose one side is 90 degrees), the square of the hypotenuse (the side opposite the right angle) is equal to the sum of the squares of the other two sides. This can be represented as:

   (H*H)=(B*B)+(L*L)

   where,
   H : length of hypotenuse
   B : length of base
   L : height of triangle

Define a lambda function isRightAngled such that isRightAngled accepts three parameters height, base, hypotenuse in order and returns true if three sides satisfy Pythagoras' theorem else returns false.

```
$ (isRightAngled 4 3 5)
#t
$ (isRightAngled 3 4 5)
#t
$ (isRightAngled 5 4 3)
#f
```

3. (6 pt) FuncLang programming: recursive functions.

   (a) (3 pt) Compute the factorial of a given number n (n is the input variable).

   (b) (3 pt) Compute the n^{th} number in the fibonacci sequence (n is the input variable).

4. (8 pt) FuncLang programming: list and pair.

   (a) (4 pt) Define a function named *max* that calculates the maximum value of a list,
   ```
   $(max (list))
   0
   $ (max (list 1 10 3 14))
   14
   $ (max (list 11 18 31 14))
   31
   ```

   (b) (4 pt) Define a function even that returns all the even numbers for a given list. Some example usage is show below,
   ```
   $ (even (list))
   ()
   $ even (list 1 2 3 4 5)
   (2 4)
   $ (even (list 5 7))
   ()
   ```

5. (5 pt) FuncLang programming: list and pair.

   Using list-related expressions list, car, cdr, caddr, null?, cons provided in FuncLang, write programs that achieve the following purpose and provide one transcript in the space for answer below:

   (a) (2 pt) Using list expression define a list named *pairs* that contains a list of 4 pairs: (5,1) (4,6) (8,7) (10,15).

   (b) (3 pt) Define a function firstSum that performs an addition on the first element of each pair on the list *pairs*. The function has to extract each element using list-related expressions. Result = 5 + 4 + 8 + 10 = 27.

6. (8 pt) FuncLang programming: high order functions.

   (a) (4 pt) Rajan's book Exercise 5.4.1 in Ch 5.

   (b) (4 pt) Rajan's book Exercise 5.4.2 in Ch 5.

7. (8 pt) FuncLang programming: high order functions and curried functions.

   (a) (2 pt) Construct a global variable mylist that holds a list of three pairs, (1,3) (4,2) (5,6).

   (b) (4 pt) Write a function apply-on-nth that takes three arguments op, lst, n, where op is a function, lst is a list of pairs, n is an integer. The return value should be the result of applying op on the n-th pair in the list. If n is out of range of the list, return -1. You can assume op is a function valid to accept two arguments.

   Some examples of using apply-on-nth with above mylist variable:
   $ (apply-on-nth + mylist 1)
   4 // 1+3
   $ (apply-on-nth - mylist 1)
   4 // 4-2
   $ (apply-on-nth - mylist 8)
   - 1 // third parameter out of range
   $ (apply-on-nth - mylist -1)
   - 1 // third parameter out of range

   (c) (2 pt) Convert the above FuncLang program into the curried form