Homework: DefineLang and FuncLang

Learning Objectives:

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

Instructions:

1. Total points: 52 pt
2. Early deadline: Feb 23 (Wed) 11:59 pm, Regular deadline Feb 25 (Fri) 11:59 pm.
3. Download hw4code.zip from Canvas
4. Set up the programming project following the instructions in the tutorial from hw2 (similar steps)
5. How to submit:
   - For questions 1–6, you can write your solutions in latex or word and then convert it to pdf. Please provide the solutions in one pdf file.
   - Submit the pdf file to Canvas under Assignments, Homework 4

Questions:

1. (3 pt) [DefineLang programming] Define a constant `pi` with the usual value of 3.14. Define a radius `r` with a value of 10. Using the definition of `pi` and `r`, calculate the volume of a sphere. Recall that the formula for volume of a sphere is $\frac{4}{3} * pi * r^3$.
2. (3 pt) [FuncLang programming] Compute the nth number in the fibonacci sequence (n is the input variable).

   Eg:
   (fibonacci 3) = 2
   (fibonacci 4) = 3
   (fibonacci 5) = 5

3. (22 pt) [FuncLang with list programming]

   (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))
()

(c) (4 pt) Write a function `count` that returns the number of times a given number occurs in the list:
$ (count 2 (list 1 2))
1
$ (count 2 (list 1 3 5))
0
$ (count 2 (list))
0
$ (count 2 (list 2 2 2 2))
4

(d) (4 pt) Write a function `unique` that removes duplicate elements in a list and returns a list of unique elements.
$ (unique (list))
()
$ (unique (list 1 10 3 14))
(1 10 3 14)
$ (unique (list 11 18 31 18))
(11 31 18)

(e) (6 pt) Write a function `fibseq` that returns the fibonacci sequence of a given input length.
$ (fibseq 3)
(0 1 1)
$ (fibseq 4)
(0 1 1 2)

4. (5 pt) [FuncLang with list and pair programming]

(a) (2 pt) Define a list named `address` that contains a list of 3 pairs: ("City", "Ames") ("State", "Iowa") ("Country", "USA").

(b) (3 pt) Write a function, `getaddress` that takes `address` as an input and returns a single list of the whole address.
$ (getaddress address)
("Ames" "Iowa" "USA")

5. (11 pt) [High order function programming] Given the following definitions of *pair* and *apair*

   (define pair (lambda (fst snd) (lambda (op) (if op fst snd))))
   (define apair (pair 2 3))

   (a) (4 pt) Write a FuncLang program to determine if the two elements of *apair* are equivalent.

   (b) (4 pt) Modify *pair* to *quadraple* to support four elements. *quadraple* is a high order function.
   (define quadraple (lambda (fst snd trd frth) ....))
   (define atuple (quadraple 2 3 4 5))

   (c) (3 pt) Write programs *first*, *second*, *third*, and *fourth* to select the first, second, third, and fourth element of a given tuple
   $ (first atuple) 2
   $ (second atuple) 3
   $ (third atuple) 4
   $ (fourth atuple) 5

6. (8 pt) [High order function programming and currying]

   (a) (2 pt) Construct a global variable "pairedList" 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 pairedList variable:
   $ (apply-on-nth + pairedList 1)
   4 // 1+3
   $ (apply-on-nth - pairedList 2)
   2 // 4-2
   $ (apply-on-nth - pairedList 8)
   - 1 // third parameter out of range
   $ (apply-on-nth - pairedList -1)
   - 1 // third parameter out of range

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