IEEE NITK

Functional Programming with Erlang

Assignment 1

Last Date for Submission: **18-May-2017**

Instructions

1. The main intention of this assignment is to get you accustomed to using the git version control system and contributing on Github.
2. Create an account on Github (if not done so already) and install git on your computer. Understand how both Github and git works.
3. Fork the summer project repository (https://github.com/IEEE-NITK/SummerProjects17)
4. Clone the local fork into your system.
5. Create a folder inside **FP-Erlang / Week 2 / Assignment 1** with your first name (A sample submission folder is created already)
6. Add all your modules (.erl files) there along with a README.md file describing all your modules.
7. Commit and push to your local fork
8. Create a pull request to the parent repository. Your first pull request will be treated as the final submission to the assignment.
9. **Do not** make any changes to any other files or folders in the repository unless directed to do so.
10. Additional innovative functions defined in the modules in addition to those specified below will be highly appreciated :)

Module 1 - myList.erl

Define wrapper functions for the following features which are already defined in the inbuilt **lists** module in Erlang.

1. Appending 2 lists.
2. Returning last element of list.

3. Finding out if an element is a member of a list.
4. Sorting a given list.
5. Returning sum of elements in a given list

Module 2 - complexCalc.erl

A calculator module for complex numbers. Define the following 4 functions that takes 4 numbers as arguments (Real and Imaginary Components of 2 numbers) and returns a single list containing the real and imaginary component of the result.

1. add() - Adds the 2 complex numbers manually
2. subtract() - Calls the add() function for computation
3. multiply() - You can use only the '*' operator, add() and subtract() functions for computation
4. divide() - You are not allowed to use any of these 3 operators ['+' , '-', '*'] to implement this function.

Next, define these 2 functions by taking help of the inbuilt **math** module in Erlang.

1. arg() - Takes a 2-membered list corresponding to a complex number as input parameter and returns the argument in degrees.
2. argInv() - Takes a single number as an input parameter and returns a complex number (2-membered list) whose argument in radians corresponds to the input (For the sake of the problem, take the magnitude of the complex number = 1).
3. absolute() - Takes a 2-membered list corresponding to a complex number as input paramater and returns the magnitude.
4. print() - Takes a 2-membered list corresponding to a complex number as input parameter and prints details of the complex number in the format "Number : a + b i, Real Component: a, Imaginary Component: b, Argument: theta, Magnitude: z". You are allowed to innovate on the format.

Useful Resources:

1. http://erlang.org/doc/man/lists.html
2. http://erlang.org/doc/man/math.html
3. https://webclub-nitk.github.io/blogs.html
4. https://www.tutorialspoint.com/erlang/