Assignment #5
CS 3060 Programming Languages, Fall 2023
Instructor: S. Roy

Haskell #1

Due Date: Hw5a is due on Nov 15 whereas Hw5b is due on Nov 17 @ 11.59 PM.
Total Points: 60 points

Directions: Using the source provided via Gitlab `https://gitlab.com/sanroy/fa23-cs3060-hw/`, complete the assignment below. The process for completing this assignment should be as follows:

1. You already forked the Repository "sanroy/fa23-cs3060-hw" to a repository "yourId/fa23-cs3060-hw" under your username. If not, do it now.
2. Get a copy of hw5 folder in "sanroy/fa23-cs3060-hw" repository as a hw5 folder in your repository "yourId/fa23-cs3060-hw"
3. Complete the assignment, committing changes to git. Each task code should be in a separate file. As an example, task1.hs for Task 1.
4. Push all commits to your Gitlab repository
5. If you have done yet done so, add TA and Roy as members of your Gitlab repository

Tasks:

1. **Task #1: (15 points)** Part of Hw5a. Write a Haskell program that prints the string "Hello, NAME, you are good." where NAME is your name. Note that you need to compile your .hs file to create an executable, using commands like, "ghc -o myhi prog.hs" and then run the "myhi" executable. Submit a screenshot that shows the above activities (which carries 5 points). *Writing README carries 1 point.*

2. **Task #2: (15 points)** Part of Hw5a. Write a function that accepts a list (lst) of integers as the parameter, and filters out a sub-list (of lst) which contains only the 7's multiples and 5's multiples in lst (if any). As an example, if lst is [10,42,12,35,15,70,20], then the output is [35,70]. *Writing README carries 1 point.*

3. **Task #3: (15 points)** Part of Hw5b. Write a function that accepts a list (lst) of integers as the parameter, and returns x where x is the number of perfect squares in the list lst. *Writing README carries 1 point.* As an example, if lst is [42,9,70,16,625,15,300], then the output is 3 as there are three perfect squares (9, 16, and 625).

4. **Task #4: (15 points)** Part of Hw5b. Write a Haskell function (named toDigitList) that accepts an integer as the parameter, and if the integer is a non-negative integer then it breaks it into its digits and the output is a list of digits. On the other hand, if the input is a negative integer, the function returns an empty list. Use the following examples to test your function:

   (a) toDigitList 1354 gives output [1,3,5,4] (b) toDigitList 0 gives output [0] (c) toDigitList (-52) gives output []

   Writing README carries 2 points.