COMP 3649 Tutorial –Haskell Basics

1. Haskell's "standard prelude" is the set of library definitions that are pre-loaded by default. It includes a `sum` function that, given a list of numbers, produces the sum of all numbers in the list.

   For example:

   ```
   ghci> sum [5,10,15]
   30
   ```

   One way of defining it yourself would be as follows:

   ```
   sum []     = 0                -- sum.1
   sum (x:xs) = x + (sum xs)     -- sum.2
   ```

   Perform a detailed evaluation trace for the example expression above. Follow exactly the same format as shown in the most recent lecture notes.

2. Without asking GHCi the type of `sum`, state a declaration of the type of `sum` that you might write if you had to provide it yourself.

3. Define and test a `prod` function that returns the product of all the integers in a given list.

   For example:

   ```
   ghci> prod [5,10,15]
   750
   ```

   Start by specifying the type declaration.

4. Define and test a `scaleBy` function that multiplies a given integer (argument 1) by each integer in a given list of integers (argument 2) to produce a new list of integers.

   For example:

   ```
   ghci> scaleBy 5 [1,2,1,3,7]
   [5,10,5,15,35]
   ```

   Start by specifying the type declaration.

   Hint: define the function recursively over the second argument, with one base case and one recursive case.