Assignment 5

Due on **February 14th, end of the day**. Please follow the submission instructions in the "notes for all labs" on Moodle.

- Read the "Notes for All Labs" document on Moodle. All assignments must be submitted as specified there. **This means .pdf**.
    - Please number your written assignment answers.
- Read Chapter 5 of your text. This assignment mostly follows along with the material covered there.
- All of your code should be in a file called your_id_330_lab5.hs, where your_id is your Earlham user ID.

Written Assignment 5.1: (2 Points)

Describe tuples and lists. What are some important differences?

Written Assignment 5.2: (2 Points)

Why doesn't Haskell let me make a list that contains a mixture of things, like `["Cats", 14, [9], 7.3, 'a']`?

Written Assignment 5.3: (2 Points)

Pictures.hs defines the type `Picture` like this:

```
type Picture = [[Char]]
```

a) What does this mean?

b) Why bother doing this at all? Why not just use `[[Char]]` everywhere we would use `Picture`?

Written Assignment 5.4: (2 Points)

In Haskell, every function returns a single value. What if I want a function to give me two pieces of information? What can I do?

Writing Assignment 5.5: (2 Points)

The **tribonacci sequence** starts with 0, 0, 1..., and every following value is the sum of the previous **three** values. It begins 0, 0, 1, 1, 2, 4, 7, 13, 24, 44, 81...

Here is some code I wrote that computes the nth tribonacci number (1-indexed):

```
trib :: Int -> Int
trib n
  | n == 1    = 0
  | n == 2    = 0
  | n == 3    = 1
  | n > 3     = trib (n - 3) + trib (n - 2) + trib (n - 1)
```

This code correctly produces a result, but it isn't efficient. Explain why, in a few sentences.

Coding Assignment 5.6: (4 Points)

Using tuples, produce a version of `trib` that is more efficient. You will likely use several functions; the name of the one that people should call should be `fastTrib`. Your textbook will be very useful in getting this working.

Coding Assignment 5.7: (8 Points)

Define a data type for a standard deck of playing cards called PlayingCard. Define it as a product type. (See page 104) You do not worry about Jokers for now - just do Two through Ace of each of the four playing card suits (Hearts, Diamonds, Clubs, Spades). If you're not familiar with the makeup of a deck like this, let me know and I can loan you an example.

Consider carefully what fields it should have, and what types those should be. (Hint: You may want to define your own new types.)

Your type should derive Typeclasses as is appropriate for what it represents, as should any new types you create. (If you want to derive Ord for a Suit type that you write, the ordering used in some games is Clubs < Diamonds < Hearts < Spades.)

Write at least two functions that use or produce your PlayingCard type. You may be using your type in future projects, so design carefully.

Written Assignment 5.8: (4 Points)

Explain how you could have written your PlayingCard data type from the last question as a synonym type (using a tuple) instead of as an algebraic type. What are some of the advantages and disadvantages of having written it as an algebraic type? If you were writing code for real use, which way would you choose to write it? Why? (This has no single correct answer, but you should justify your choice.)

Coding Assignment 5.9: (4 Points)

The following should be done with **list comprehensions**.

a) Give a definition of a function

```
doubleAll :: [Integer] -> [Integer]
```

which doubles all of the elements of a list of integers and returns the result.

b) Define the function

```
divisors :: Integer -> [Integer]
```

which returns a list of all of the integer divisors of a positive integer you pass to it. For example, divisors 12

Should return `[1,2,3,4,6,12]`

Hint: You may find the notation introduced in the bottom half of page 110 useful.

Discussion Prep 5.10: (4 Points)

Include **one to three** thoughts or questions you had about the Chapter 5 reading.You may put an asterisk (*) next to the ones that you **most** want to discuss in class, if any.

Preparation for the Future

Moving forward, the expectation will be that you feel comfortable with the topics covered in this assignment. If you do not, you should ask questions on Piazza and/or continue to practice with GHC. If you want more practice, try the various exercises in Chapter 5 of your textbook. (You do not have to submit these, but I would encourage you to do a few.)