HOMEWORK 1: CS 4450

DUE: 3:00PM, 9/22/17

Directions

- Name your submission file as follows. For groups, submission files must be named `{groupname}_homework1.hs`. For example, a group named `reynolds` would submit `reynolds_homework1.hs`. For individuals, submission files must be named `{pawprint}_homework1.hs`. For example, an individual with a pawprint `tnrn9b` would submit `tnrn9b_homework1.hs`.
- Your submission *must* load and typecheck in Haskell Platform to get any points.
- Every function you define must have a type signature in order to get full credit for the problem.
- Functions must be documented. Examples provided in the assignment file.
- Name all functions and data types exactly as they appear in the assignment. You cannot import any additional libraries or enable any additional language features unless granted explicit permission from the instructor.
- The code you submit must be your own. *Exceptions*: you may (of course) use the code we provide however you like, including examples from the slides.
- No late submissions! Don't submit the assignment hours after it is due and claim that Canvas wasn't working. If Canvas truly isn't working, you should submit the assignment via email minutes after it is due (3:00 PM) – not hours after it is due.

Problems

1. Define a function `prob1` that takes a character, `‘a’,..,‘z’` or `‘A’,..,‘Z’` , as input and returns the next character occurring in alphabetical order. Furthermore, `‘z’` and `‘Z’` should return `‘a’` and `‘A’`, respectively. Also, given any other character input should return that input unchanged.

```
*HW1> prob1 ‘a’
‘b’
*HW1> prob1 ‘1’
‘1’
```

2. Define a function `prob2` that takes a digit, `‘0’,..,‘9’` as input and returns the numeric value of that digit as an `Int` type. Furthermore, any non-digit input should return `-1`.

```
*HW1> prob2 ‘0’
0
*HW1> prob2 ‘a’
-1
```

3. Define a function `prob3` that takes three arguments; two functions and a value to serve as their input and returns a tuple of the their outputs.

```
*HW1> prob3 prob1 prob2 ‘a’
(‘b’,-1)
```

4. Define a function `prob4` that takes three arguments; a `Bool` and two return values. If the `Bool` is `True`, the second argument is returned, otherwise the third argument is returned.

```
*HW1> prob4 (1 == 1) ‘a’ ‘b’
‘a’
*HW1> prob4 False ‘a’ ‘b’
‘b’
```

5. Define a function `prob5` that takes a year (a number of type `Integer`) from the Gregorian calender and returns `True` if that year is a leap year and `False` otherwise.

```
*HW1> prob5 1900
False
*HW1> prob5 2000
True
```

Grading

Function	**Points**
`prob1`:	10
`prob2`:	10
`prob3`:	10
`prob4`:	10
`prob5`:	10
Total	**50**