Homework 1 Due September 4th 11:59 pm

Submission rules:

- All text answers must be given in Haskell comment underneath the problem header.
- You must submit a single .hs file with the following name: firstName-lastName-hw1.hs. Failure to do so will result in -10 points.
- You will lose 10 points if you put a module statement at the top of the file.
- You will lose 10 points for any import statements you have in your file and will automatically miss any problems you used an imported function on.
- If your file doesn't compile you will lose 10 points and miss any problems that were causing the compilation errors.
- This means that any function which is causing compiler errors should be commented out. There will be no partial credit.
- You must use the skeleton file provided and must not alter any type signature. If you alter a type signature you will automatically miss that problem.

Problems

Problem 1 Prelude Types (1 pt each, 15 pts)

Give the type of each of the following builtin Haskell functions.

- `head`
- `tail`
- `fst`
- `snd`
- `length`
- `null`
- `take`
- `drop`
- `sum`
- `product`
- `(++)`
- `(!!)`

- `elem`
- `(:)`
- `last`

Problem 2 More Types (1 pt each, 15 pts)

What is the type of the following Haskell expressions?

- `True`
- `not True`
- `not`
- `1 + 2`
- `(+) 1 2`
- `(+) 1`
- `(+)`
- `[True, False]`
- `[1, 2, 3]`
- `[1, 2, 3] ++ [4, 5, 6]`
- `[]`
- `[[]]`
- `[[], []]`
- `take 10 []`
- `take 10`

Problem 3 Layout (3 pts each, 15 pts)

Fix the error in each of the following pieces of code.

- ```
  f = x + y
    where
      x = 1
       y = 2
  ```
- `g X Y = X + Y`
- `Foo x y = x + y`
- `h = [1,2,3] ++ 4,5,6`
  ```

- ```
  u = x * y
  where
  x = 1
  y = 2
  ```

Problem 4 Classes (3 pts each, 18 pts)

For each problem give a list of functions separated by commas. For example the Eq typeclass defines the (==), (/=) functions.

- Which functions does the Ord typeclass define?
- Which functions does the Show typeclass define?
- Which functions does the Read typeclass define?
- Which functions does the Num typeclass define?
- Which functions does the Integral typeclass define?
- Which functions does the Fractional typeclass define?

Problem 5 Functions (37 pts)

- (5 pts) (Chapter 4 Exercise 1) Define a function, `halve :: [a] -> ([a], [a])`, which takes an even-length list and splits in half. For example:

  ```
  halve [1, 2, 3, 4] == ([1, 2], [3, 4])
  ```

- (10 pts) (Chapter 4 Exercise 2) Define a function, `fourth :: [a] -> a`, which takes a list and returns the fourth element. For example:

  ```
  fourth [1, 2, 3, 4, 5] == 4
  ```

  You must define this function in three ways (name each function fourth1, fourth2, fourth3):

  1. (3 pts) Using the head and tail functions.
  2. (3 pts) Using the !! function.
  3. (4 pts) Using pattern matching.

- (22 pts) (Chater 4 Exercise 8) Luhn algorithm is used to check credit card numbers for simple errors such as mistyping a digit, and proceeds as follows:

  - consider each digit as a separate number;
  - moving left, double every other number from the second last;
  - subtract 9 from each number that is now greater than 9;
  - add all the resulting numbers together;
  - if the total is divisible by 10, the card number is valid.

  Define a function `luhnDouble :: Int -> Int` that doubles a digit and subtracts 9 if the result is greater than 9. For example:

  ```
  > luhnDouble 3
  6
  ```
  ```

```
> luhnDouble 6
3
```

Using `luhnDouble` and the integer remainder function `mod`, define a function `luhn :: Int -> Int -> Int -> Int -> Bool` that decides if a four-digit bank card number is valid. For example:

```
luhn 1 7 8 4 == True
luhn 4 7 8 3 == False
```