

CMPT 383 Comparative Programming Languages

Homework 2

This homework is due by 11:59pm PT on Tuesday Jan 28, 2025. No late submission is accepted. Please save your Haskell code in a single file called `H2_SFUID.hs` (replacing `SFUID` with your 9-digit SFU student number) and submit it to Canvas.

Requirements of this homework:

- Write type signatures for all functions using the `::` operator.
- Do not use the `if-then-else` expression unless specified in the question.
- Please ensure the command `ghci <path_to_your_file>` can load your code without errors. Failure to comply with this instruction may result in deducted marks.

1. (10 points) Write a recursive function `myFoldl` that implements the standard `foldl` for lists.

Sample input and output:

```
ghci> myFoldl (+) 0 [1,2,3,4]
10
ghci> myFoldl (flip (:)) "" "abcd"
"dcba"
```

2. (10 points) Write a recursive function `myFoldr` that implements the standard `foldr` for lists.

Sample input and output:

```
ghci> myFoldr (*) 1 [1,2,3,4]
24
ghci> myFoldr (:) "" "abcd"
"abcd"
```

3. (20 points) Write a function `alternativeMap :: (a -> b) -> (a -> b) -> [a] -> [b]` that alternatively applies two argument functions to elements in the list.

Sample input and output:

```
ghci> alternativeMap (+10) (+100) [1]
[11]
ghci> alternativeMap (+10) (+100) [1..6]
[11,102,13,104,15,106]
```

4. (20 points) Use `foldr` to implement a function called `myLength` that takes a list as input and returns an `Int` denoting the length of the list.

Sample input and output:

```
ghci> myLength [1..10]
10
ghci> myLength "abc"
3
```

5. (20 points) Write a function called `myFilter` using `foldl` to implement the standard `filter`. Note that you can use `if-then-else` for this question.

Sample input and output:

```
ghci> myFilter even [1..10]
[2,4,6,8,10]
ghci> myFilter (>5) [1..10]
[6,7,8,9,10]
```

6. (20 points) Write a function `sumsqeven` in **point-free style** that takes a list of `Int`'s and returns the sum of squares of all even numbers in the list.

Sample input and output:

```
ghci> sumsqeven [1,2,3,4]
20  -- 2^2 + 4^2 == 20
ghci> sumsqeven [2,4,6]
56  -- 2^2 + 4^2 + 6^2 == 56
```