

CMPT 383 Comparative Programming Languages

Homework 1

This homework is due by 11:59pm PT on Wednesday Jan 26, 2022. No late submission is accepted. Please save your Haskell code in a single file called `h1_firstname_lastname.hs` (h in lower case, firstname and lastname replaced with your first and last name) 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 in any function.

1. (20 points) Given an `Int` n ($n \geq 1$) and an `Int` m ($m \geq 0$), write a function called `seqMultiple` that returns a list of n boolean values, where the i -th value ($0 \leq i < n$) indicates whether $i + 1$ is a multiple of m . Note that you might want to use the built-in `mod` function: `mod x y` returns the remainder when x is divided by y .

Sample input and output:

```
ghci> seqMultiple 5 3
[False,False,True,False,False]
ghci> seqMultiple 5 2
[False,True,False,True,False]
```

2. (20 points) Given an `Int` n ($n \geq 0$), write a recursive function called `fib` that computes the n -th Fibonacci number. The n -th Fibonacci number is defined as:

$$fib(n) = \begin{cases} 0, & n = 0 \\ 1, & n = 1 \\ fib(n-1) + fib(n-2), & \text{otherwise} \end{cases}$$

Sample input and output:

```
ghci> fib 10
55
ghci> fib 20
6765
```

3. (20 points) Given a list L , write a recursive function called `listReverse` that reverses the list L .

Sample input and output:

```
ghci> listReverse [1, 2, 3]
[3,2,1]
ghci> listReverse "abc"
"cba"
```

4. (20 points) Given two lists of `Int`'s, write a recursive function called `listAdd` that computes their sum. Specifically, the sum of two lists $[x_0, \dots, x_n]$ and $[y_0, \dots, y_m]$ is a list $[z_0, \dots, z_k]$ where $k = \max\{m, n\}$ and

$$z_i = \begin{cases} x_i + y_i, & \text{for } 0 \leq i \leq \min\{m, n\} \\ x_i \text{ or } y_i (\text{whichever exists}), & \text{for } \min\{m, n\} < i \leq k \end{cases}$$

Sample input and output:

```
ghci> listAdd [1, 2] [3, 4]
[4,6]
ghci> listAdd [1, 2, 3] [4, 5]
[5,7,3]
ghci> listAdd [1, 2] [3, 4, 5]
[4,6,5]
```

5. (20 points) Given a list L of values and a value V , write a recursive function called `inList` that checks if V occurs in the list L .

Sample input and output:

```
ghci> inList [1, 2, 3] 2
True
ghci> inList "abc" 'b'
True
```