CS 350 2020
HW 4

Due date: December 1, 23:59

The monad laws are the following.

a . **Left identity.** `return a >>= f` is equivalent to `f a`

b . **Right identity.** `m >>= return` is equivalent to `m`

c . **Associativity.** `m >>= f >>= g` is equivalent to `m >>= (\ x -> f x >>= g)`

---

1 . Prove that the Maybe Monad satisfies the monadic laws. [10]

2 . Prove that the List Monad satisfies the monadic laws. [10]

3 . Write a memoized version of the function to compute the n^{th} Fibonacci number. Assume that the first Fibonacci number is 1. Hint: Use an appropriate monad. [10]

4 . Write an interactive function which takes a sequence of non-negative integers as input, one on each line. The end of sequence occurs when the user inputs -1. You should output the sum of the non-negative integers.

a . Do the above exercise using the **do** notation in Haskell. [10]

b . Do the same exercise, without using the **do** notation. [10]