Exam 1 correction completing 6 problems

Logan Montgomery

1.)

product [] = 0

product (x:xs) = x*xs

I went wrong in trying to make the function with list comprehension and I don't know list comprehension well enough to make it work. I also made a recursive function with no base case.

2.)

A) List of Bools

B) tuple of bools

C) List of Strings, This one I originally thought was just a string such that each char was represented in a list but it is in fact a list comprised of strings.

D) List of Chars

E) List of functions that return lists, This one I just said was a list of functions not stating that the functions had a common output.

3.)

A) [a] -> a

B) Num a => a -> a I had not stated that 'a' must be of type Num

C) Eq a => [a] -> Bool This needs a to be of Equality type and will return a Bool

D) a -> b -> c -> (a, b, c) The inputs don't have to be of the same type will return a tuple of different type things

E) (c, a, b) -> (a, b, c) Similar to d the elements in the input tuple don't have to be the same type and what s returned will maintain the original element type

4.)

foo = (\x -> (\y -> (\z -> (y * z)+x))) On the exam I had not defined the lambda correctly I only lambda the x value and forgot all of the ->'s in my expression

5.)

hundcube = sum [x^3 | x <- [1..100]]

My original function I tried to create a list, I most likely mis-read the instructions and tried to create a list of each of the cubes rather than summing them. My function does not do this properly either because I used mismatched variable types.

6.)

triangleNumber x = sum [n | n <- [1..x]]

On the exam I had forgot that putting an operator in the list comprehension will create a list and sum each iteration like I wanted rather I had to apply the sum function to the list comprehension.