MA 151: Homework #2

due Tuesday September 15

Written problems

In each of these, simplify the expressions step-by-step to get the final value. If there is an error, say exactly what the problem is. You should show enough detail to make it clear that you know what is going on. In all cases, you should be able to check your answer by typing the expressions into GHCi.

Assume that the following definition has been loaded into GHCi: \( f \ x = 2 * (x + 1) \)

Now evaluate:

1. if ((f 4) > 3) then (f 2) else 0
2. tail (head [[2,3,4]])
3. drop 5 (take 20 [1..])
4. take 3 (repeat (take 3 (repeat 4)))
5. take 5 ("h":(repeat 'i'))

Programming problems

- Write a function called isEven which takes an integer \( n \) and returns the Boolean True if \( n \) is even and False otherwise. Try to do this without using an if block.
  
  Also create a function called isOdd which takes an integer \( n \) and returns True if \( n \) is odd and False otherwise.
  
  (The above two functions are already built into Haskell as even and odd.)

- Write a function called upToDouble which takes an integer \( x \) and returns the list from \( x \) up to \( 2x \). So upToDouble 5 is [5,6,7,8,9,10].

- Write a function called chopper which takes a list and returns the same list with the first and last 3 elements deleted. (If there are fewer than 7 elements, the answer should be empty.) For example:
  
  chopper "calculator" is "cula"

- Define a function called taketwice which takes one Int parameter \( n \) and a list \( l \) and returns a list of length \( 2n \) consisting of the first \( n \) elements of \( l \) followed by the first \( n \) elements of \( l \) again. For example:
  
  taketwice 3 ['a','b','c','d','e'] is ['a','b','c','a','b','c']

  You should probably use take inside your definition.

- Define a function called monkeyInTheMiddle which takes a string parameter, and inserts the string "MONKEY" in the middle. “The middle” means that the part before "MONKEY" is half (rounding down) of the length of the input string. For example:
  
  monkeyInTheMiddle "calculators" is "calcuMONKEYlators"