

MA 151: Homework #5

due Tuesday October 24

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. If the function gives an infinite loop, explain in general terms what the output will be. 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 definitions have been loaded into GHCi:

```
f (x:xs)
| x <= 4    = 1 + f xs
| otherwise = 2 * f xs
```

```
g 0 = 1
g x = x^2 + g (x-1)
```

Now evaluate:

1. `f [3,4,5]`
2. `g 5`
3. `map (>3) [1..5]`
4. `map (drop 1) (map (take 3) ["Hello","mother","dear"])`

Programming problems

Include a type signature for all functions.

- Write a function called `bigWords` which takes a list of strings and gives the list consisting of only those words which had length 8 or greater. Use recursion- no list comprehension.
- Write a function called `range` which takes two integers a and b and functions exactly like `[a..b]`. Use recursion- don't use any list comprehension or `..` in your answer. (Make sure you handle the cases when $b \leq a$.)
- Write a function called `exclaim` which takes a string and puts a `!` after every word. For example `exclaim "hello my name is bill"` is `"hello! my! name! is! bill!"` (Hint: this is like `aLover`. Think of it as replacing every space with something.)
- Write a function called `numerology` which converts a lowercase character to a number so that `numerology 'a'` is 1, `numerology 'b'` is 2, `numerology 'z'` is 26, etc. Use recursion. (Hint: you can use `pred` with characters.)
- Write `listOfLengths` from Homework #3 using `map` without using a list comprehension. Make it polymorphic.