

```

1  -- Name:      Terry Weiss (466751950)
2  -- Email:     ttweiss@syr.edu
3  -- Section:   M013 LSB-105
4  -- Project:   Homework 3
5
6  -- Exercise 1 - Generates a list of `n` pairs ascending from i
7  squarePairs :: Int -> Integer -> [(Integer,Integer)]
8  squarePairs n i
9      | n <= 0      = []
10     | otherwise   = (i, i*i) : squarePairs (n-1) (i+1)
11
12 -- Exercise 2 - Generates a list from `m` down to `n` in increments of `diff`
13 countDownBy :: Int -> Int -> Int -> [Int]
14 countDownBy m n diff
15     | diff <= 0 || m < n = []
16     | otherwise         = m : countDownBy (m - diff) n diff
17
18 -- Exercise 3 - Generates a list of interval lists of incremental
19 --               length from `m` to `n`
20 steps :: Int -> Int -> [[Int]]
21 steps m n
22     | n < m      = [[]]
23     | otherwise  = list m
24   where list :: Int -> [[Int]]
25         list i
26             | i > n      = []
27             | otherwise  = countUp m i : list (i+1)
28
29         countUp :: Int -> Int -> [Int] -- countUp function created in class
30 countUp a b
31     | a > b      = []
32     | otherwise  = a : countUp (a+1) b
33
34 -- Exercise 4 - Generates a string `n` characters `c` with `!` at element `i`
35 indexChar :: Int -> Int -> Char -> String
36 indexChar n i c
37     | n <= 0      = ""
38     | i == 1      = '!' : remainder
39     | otherwise   = c   : remainder
40   where remainder = indexChar (n-1) (i-1) c
41
42 -- Exercise 5 - Generates a list of strings `n` characters `c` with
43 --               `!` replacing the i-th character left-to-right.
44 diag :: Int -> Char -> [String]
45 diag n c = list 1
46   where list :: Int -> [String]
47         list i
48             | i > n      = []
49             | otherwise  = indexChar n i c : list (i+1)
50

```