Lab Two – Recursive Functions

Luca Gristina
luca.gristina1@marist.edu

October 2, 2023

1 Lab 2 Reflection

Before starting this Lab, I was brainstorming ways that I could be able to solve this problem from what we learned in class and I was struggling to find a solution. I was originally thinking about it like I would in Java and was thinking to just iterate through each of the lists and populate them, until I was reading through chapter 4 in the book and came across list comprehensions. I found that they were way easier to use than mapping and made my function one line of code to create all of the lists.

I started this lab by writing the Erlang code. I used recursion and list comprehensions to create and format the lists to achieve the desired output. When I first made the generateList function the output, although correct, was displayed wrong because of Erlang's pattern matching to the ascii values in the lists. So I had to recursively go back through the lists and format each of them to display the integer values. In the java version of this program that I wrote I made ArrayLists inside of Arraylists. I used nested for loops to calculate the needed values and populate the inner list with the integers and to populate the outer list with the inner lists.

After completing this lab, I found Erlang to be quite powerful in its ability to complete this assignment. Despite this though, coding in Erlang first didn't seem to change my approach for the Java version. I still felt the best way to solve it was the way that I solved it using the nested for loops to populate the lists, but the coding in Erlang gave a good challenge for this lab.

2 Transcript of Shell/Test Cases

```
1> c(lab2).
{ok,lab2}
2> lab2:test().
Formatted List:
[14,28,42,56,70,84]
[13,27,41,55,69,83]
[12,26,40,54,68,82]
[11,25,39,53,67,81]
[10,24,38,52,66,80]
[9,23,37,51,65,79]
[8,22,36,50,64,78]
[7,21,35,49,63,77]
[6,20,34,48,62,76]
[5,19,33,47,61,75]
[4,18,32,46,60,74]
[3,17,31,45,59,73]
[2,16,30,44,58,72]
[1,15,29,43,57,71]
Raw List:
test passed
ok
3> lab2:test2().
Formatted List:
[10,20,30,40,50]
[9,19,29,39,49]
[8,18,28,38,48]
[7,17,27,37,47]
[6,16,26,36,46]
[5,15,25,35,45]
[4,14,24,34,44]
[3,13,23,33,43]
[2,12,22,32,42]
[1,11,21,31,41]
Raw List:
test passed
ok
4> lab2:test3().
test passed
ok
5> lab2:test4().
test passed
ok
6> lab2:main(-5,10).
{error,"Invalid input:
Please enter a non-negative integer."}
7> lab2:main("e",5).
{error,"Invalid input:
Please enter a non-negative integer."}
```

```
$ javac lab2.java
$ java lab2.java

----------------------------------
Test Case 1: example from lab page

[14, 28, 42, 56, 70, 84]
[13, 27, 41, 55, 69, 83]
[12, 26, 40, 54, 68, 82]
[11, 25, 39, 53, 67, 81]
[10, 24, 38, 52, 66, 80]
[9, 23, 37, 51, 65, 79]
[8, 22, 36, 50, 64, 78]
[7, 21, 35, 49, 63, 77]
[6, 20, 34, 48, 62, 76]
[5, 19, 33, 47, 61, 75]
[4, 18, 32, 46, 60, 74]
[3, 17, 31, 45, 59, 73]
[2, 16, 30, 44, 58, 72]
[1, 15, 29, 43, 57, 71]

----------------------------------
Test Case 2: More expected values

[10, 20, 30, 40, 50]
[9, 19, 29, 39, 49]
[8, 18, 28, 38, 48]
[7, 17, 27, 37, 47]
[6, 16, 26, 36, 46]
[5, 15, 25, 35, 45]
[4, 14, 24, 34, 44]
[3, 13, 23, 33, 43]
[2, 12, 22, 32, 42]
[1, 11, 21, 31, 41]

----------------------------------
Test Case 3: negative integer handling

Error: Please input a non-negative integer

----------------------------------
Test Case 4: Wanted to use a character to
test my handling but it wouldn't compile
because its not an integer obviously

Error: Please input a non-negative integer
```