

# Language Study: Erlang

CMSC 233

## -Homework 2

---

Goals

To enjoy Erlang's functional nature, experiment with modules that have both public and private components, compare and contrast Erlang with an Object-oriented language, and (most importantly) bask in the glory that is recursion.

Instructions

Develop **two** programs, one in Erlang and another in your choice of [Pascal, C, C++, Modula-2, C#, Java, JavaScript, Python] to generate a list of  $M$  other lists wherein each of the "other" lists contains an  $N$ -length sequence of every  $M^{\text{th}}$  integer.

For example, if  $N$  is bound to 6 and  $M$  is bound to 14 then we expect 14 lists of 6 elements each spaced by 14 units.

```
[ [ 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 ] ]
```

(Don't use  $N$  and  $M$  as variables; those are terrible names. Pick better identifiers.)

Lastly, write a paragraph or two reflecting on practical and philosophical differences between your two programs.

Submitting

Print out . . .

- your source code for both programs
- your test cases
- a transcript of two successful runs for each program with expected data
- a transcript of two successful runs for each program with unexpected data that would have caused errors had you not prevented it
- your reflections

. . . and **staple it all together** and hand it in at the start of the class in which it is due. Remember to include your name somewhere where I can easily find it.