

1 Erlang and Java Program Differences

Solving this list problem was very different in Erlang compared to Java. In Java, I first created a scanner to read the two values in from the user, and then send both of those values to a function that creates the list of lists (which is a two-dimensional array in java). Using nested for loops for the rows and columns of the two dimensional array, each element was populated using the values the user entered to ensure the elements in each row are increasing by the same amount.

In Erlang, I used two functions that recursively call themselves to create the list of lists. The first function, `makeLists`, calls the second function, `create`, and then recursively calls itself in a list decreasing the number of lists each time. The `create` function recursively calls itself subtracting one from the size of the lists each time. Essentially, the first function takes care of the number of lists, and the second function takes care of the number of elements in each list. These two functions can be compared to the two for loops used in the Java program. Thinking in terms of recursive functions rather than tracing through for loops was much harder and more time consuming despite being less code. The closest output I got in Erlang was only getting the last list correct as each list after had a decreasing sequence all the way down to one.