Exercises Day 2

September 29

Problem 1: Code refactoring
Take one of the longer programs you wrote for the exercises on day 1 and divide it up into many small functions. Try to make sure that every function you write has a single responsibility and that they are shorter than e.g. 10 lines.

Put the resulting code in an external "library" and set up your build system of choice (e.g. `make`) to compile and link your program.

Problem 2: Matrix Operations
Write a function that add two matrices together and somehow return the result. How would the function signature have to be for that? While you are at it you can also implement matrix multiplication.

Problem 3: Array manipulation library
There is a large number of array manipulation functions one encounters regularly when programming. In this exercise we will write our own array library where we implement a lot of these useful features. For now you can implement everything for arrays of integers for simplicity. Here is a list of the desired functionality

- Reversing an array
- Pivoting an array around a specific point
- Sorting an array (you can e.g. use the bubblesort algorithm, or quicksort if you are cool)
- Searching for an integer in an unsorted array
- Efficiently searching for an integer in a sorted array (recursively?)
- Creating a new array with a filter
- Making a version where the filter can be an arbitrary unary function

Problem 4: Interest rates
Picture for once a life in which you were really bad at mathematics but still wanted to be able to keep track of your savings. How would it be possible to calculate interest rate on your savings using recursive functions?

- Write a function that calculates your total savings after `n` years if you have a fixed interest rate and make additional deposits every year.

Problem 5: Integration library
Write a small numerical integration library that can carry out simple one dimensional integrals. Use whatever algorithm you feel like (forwards, backwards, midpoint, Simpson's). Come up with a good function signature for these integration functions. What would you do if you wanted to pass additional function parameters?

Problem 6: Hangman
Today's game of choice is hangman. Write a quick implementation. You probably need some sort of database of possible nouns to use for your hangman game.