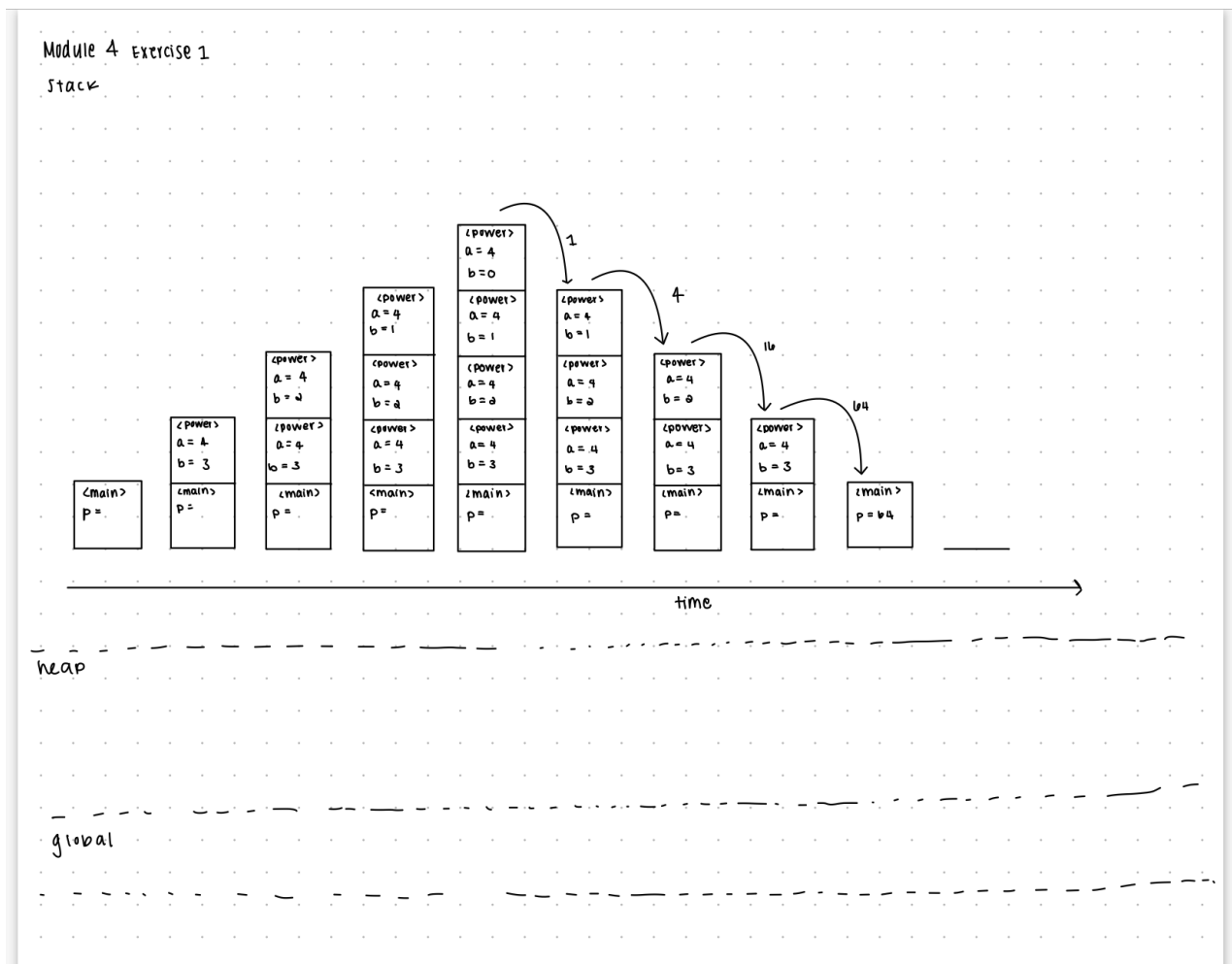


## Module 4, Exercise 1



## Exercise 4

Making “a” a global variable does not impact the output of this program because the program only really relies on “b” as a parameter. With each recursive call, the only element that is being altered is “b,” and we only really need to know that the function should repeat *a* number of times.

## Exercise 5

The output for PowerExample5 matches the output for PowerExample4 exactly despite it lacking any parameters to the recursive function `power()`. The outputs are the same because we declare “a” and “b” to be global variables, whose scope allows them to be accessed anywhere within our program. The recursive function `power()` still modifies `b` with every call, and still references `a` to know how many times to run, however, it access variables that are declared in the global part of memory instead of asking the main function for parameter inputs.

**Exercise 6**

If we switched the two if-statements in the search() method, the program would first search to see if the current value in the array matched the value being searched for, and would then check to see if the index was greater than the length. Overall, switching the two if-statements shouldn't make a significant difference in the program's accuracy or runtime, it's just checking the conditions in a different order.

**Exercise 8**

We need the "or" (||) in the if condition because both a row value of 0 **OR** a column value of 0 mean that there is only one way to (0,0), and we have reached our base case. If there was an "and" (&&) instead, countPaths would return 1 if the row value was 0 even when there might be more ways to reach (0,0) with column values > 0.

There are 56 ways to get from the corner of Park and 55th to the corner of 2nd and 50th.

**Exercise 10**

There are so many calls because the recursive return statement actually makes two recursive calls, and each of those will make two more recursive calls until they reach the base case where  $n \leq 2$ . As long as  $n$  is a large number, requiring many decrements to get to the base case, the amount of recursive calls will be doubled every time the statement is called.