4) Making variable a global does not change the output of the algorithm because a is set equal to 3 and then reset to be equal to 2 later in the program. Also, the method can still access the variable because it is global; it spans all functions.

5) B controls the recursion because it is incremented down by one every time the power function is called. This function works because a and b are established as global variables; they can be used in the void and power. However, this code has a bug in the print line because the intermediate result always states a^0 = value. This is because b has been decreased to 0 for the recursive function to work.

6) The function would not change if the two if statements were swapped. These if statements are testing two different statements, so the order does not matter here.

8) The or is necessary because the base case occurs when you are at an edge. If there was an and statement, the base case would be at the point we are trying to reach, and the amount of paths should be 0, which is not an effective base case for this problem. There are 56 ways to get from Park and 55th to 2nd and 50th.

10) There are so many calls in the Fibonacci problem because the function calls fib(n-1) and fib(n-2) for every number even if it has already solved the problem for that number.