CLASS ASSIGNMENT:

The class homework assignment was to create a Binary Heap, using either an array or linked list implementation. The requirements were to have an insert() function, an extract() function(either min or max depending on the type of heap), and a function to display the heap in a visual representation of a binary tree. The program could be written in Java or Python.

The type of heap I made was a min- array Binary Heap. I always wanted to do a Min Heap in the context of this assignment but I originally wanted to do it within a linked list implementation and within Java. That didn't go as well as I would've liked, as I struggled with dealing with the additional Java syntax on top of trying to figure out how to code the project. I eventually decided to switch to Python, even though Java is much closer to a language I am comfortable with using, C/C++. I also decided to change the implementation to an array as I felt more confident in my ability to work with them than to work with linked lists.

The program is split into two different files: the file with all the code and a driver file. I, unfortunately, committed a sin and used global variables in order to keep track of the root, count and an i variable that would be used in a for loop later. At the top, I create an empty array called heap_list (one thing that makes python easier than other languages), a global variable i and set that to zero. Then I created a class called heap and put the insert/extract/ display functions within it.

The insert function is essentially a paragraph of if/else statements divided into three parts. The first part checks if the item inserted is the first number, if it is then it becomes the root. The second part checks if the item inserted is less than the root, if it is then the two swap. The last part adds the item then checks to see if it is in the correct position.

The extract function will just need to pop the min. Python has a .pop() function so that wasn't an issue compared to replacing the root. I took the rightmost leaf(which would be the last item in the array, and replaced the root with it. The program would then filter it to determine where it needs to go.

The display function is just an inner for loop. I could'nt imitate the recursion used in the class example so I used loops instead. The variable Level is declared at zero and the program runs for the length of an array. There is an inner for loop that runs for how big level is plus 1 and it prints spaces equal to level + 1, then level increments by 1. The outer loop ends with a print statement printing the heap list at position x. This program is simple but I tried to imitate the recursive code of the class example which gave me a hard time solving it.