Project #7
CS 2210 – Fall 2022
Christopher LaFave

I. Requirements: Create an array-based heap that can handle add and removing.

II. Design: I wrote 4 methods, add, remove, bubbleUp, and bubbleDown.
Add and bubbleUp handle the inserts,
Remove and bubbleDown handle the retrevals.

III. Security Analysis: This was focused on integers only, so any other type of data might break it or mess with memory.

IV. Implementation: The add method added the input to the end of the array heap and then bubbled it up to sort it in. Before adding, it checked if it was about to go over the array's memory and increased its size if necessary.

The remove method swaps the root with the last element in the array heap and bubbles it down. It stores the root to be returned before swapping. After bubbling down, it adjusts the size and returns the correct value.

The bubbleUp method checks for nulls first off, and then checks the keys of pos and its parent, and swaps them if pos is smaller. If it is not, it ends the method.

The bubbleDown method also checks for nulls first, and then checks what is the smallest between pos and its children. It swaps if necessary, and if pos is the smallest, it ends the method.

V. Testing: I tested it with the test program provided and my inserting 100,000 numbers between 0 and 699,999 onto the heap and taking it off.

VI. Summary/Conclusion: My program runs properly and gets all the correct outputs.

VII. Lessons Learned: I became more comfortable with how heaps work and became much better at sorting through Binary Trees. I also learned to start these projects sooner because I am turning this in half an hour late (sorry!)