**Heap – In-Class Practice**

1. Create a class named MinHeap in a file named MinHeap.h
2. The MinHeap class should have the following in the private field:
   1. A string pointer to a dynamically allocated array called heap.
   2. An int called sz for size.
   3. An int called capacity for the max size of the heap.
   4. A helper function called parent that takes an int i and returns the parent node index of the current child index.
   5. A helper function called left that takes an int i and returns the left child index given the parents index.
   6. A helper function called right that takes an int i and returns the right child index given the parents index.
   7. A void function that resizes the heap.
3. The MinHeap class should have the following in the public field:
   1. A constructor that takes an int c for the starting capacity. It should set the capacity to what is passed in, sz to 0, and initialize heap.
   2. A destructor that will delete the dynamically allocated array.
   3. A void insert method that takes a string and inserts it into the heap using the string length for comparison such that the smallest string will be the root.
   4. A remove method that returns root and reheaps the heap using the length of the strings.\*
   5. A isEmpty method that returns a bool: True if sz = 0 and false if sz != 0.
   6. A getSize method that returns sz.
   7. A getHeight method that uses the ceil and log2 functions from the cmath library to return the height of the heap.
   8. A peek method that returns the value at the root of the heap but does not remove it.\*
   9. A clear method that deletes the old heap and resets the size to 0.  
      \*if sz is 0, return the empty string: “”
4. Now create a driver file called Driver.cpp that does the following:
   1. Dynamically allocate a new MinHeap with initial size 4 called heap.
   2. Then insert:
      1. everyday
      2. do
      3. coding
      4. like
      5. I
   3. Now, print the size, height, and root using getSize, getHeight, and peek
   4. Set an int s to the size of the heap.
   5. Use a for loop to print out each node in the heap using the remove method. Be warned you must use the value s in the conditional for the loop and not getSize.
   6. Then insert the word “nonsense” into the heap.
   7. Then clear the heap using clear.
   8. Then use isEmpty to check if the is empty and if it is, print “Heap has been cleared to the screen”
   9. Delete heap.
5. The driver should print “I do like coding everyday” to the screen