Data Structures (Spring 2020)

Lab Exercise 10

Task 1: Download lab10.cpp from NYU classes and complete following methods of **Heap** (minheap) class.

1. **int Heap::removeMin()**
   Remove the minimum value from Heap keeping the Heap property intact

2. **int Heap::getMin()**
   Return (but don't remove) the minimum value from the Heap

3. **int Heap::parent(int i)**
   Returns the index of the parent of the node i

4. **int Heap::left(int i)**
   Returns the index of the left child of the node i

5. **int Heap::right(int i)**
   Returns the index of the right child of the node i

6. **void Heap::siftup(int k)**
   Sift-up an element at index k

7. **void Heap::siftdown(int k)**
   Sift-down an element at index k
   Hint: You can use following algorithm:
   1. if element has no children do nothing
   2. if element has only a left child which is smaller than element then swap the element with left child
   3. if element has both children then swap the element with smaller child
   4. keep repeating step 1-3 until the element/node reaches to its correct position

Task 2: Write a function mySort(int array[], int) which takes an array as an argument and sort the array using Heap.
Hint: Create an auxiliary Heap object and first insert and then remove elements from it.
