**Lab 2 Report**

This lab is about Priority Queues implemented as heaps. I use either an array or a vector in your implementations.I create 2 classes, and they are

* QuadHeap
* DHeap

**Task 1**

This task is asked us to find the algorithm that in an array, how the tree/heap relationship build. If each node has n children so how can I get the index of children? I defined a function and it returns the array of children index.

childIndexArray[i] = parentIndex \* numOfChild + i + 1;

So if a node of a heap has 4 children, it will be like:

childIndexArray[i] = parentIndex \* 4 + i + 1;

/\*@return address of index array head\*/

**Task 2a**

It is a real heap and data is arrangement in an array physically and in a tree logically. Insert method is quite simple, but need a extra help function called hepifyup. It is help compare the child node and the parent node and change the position if needed.

deleteMin() is a little difficult. And There are two ways to define this function whether the heap is. If the root is minimum data, it need the tail data change the position to the root and do the heapifydown operation. I write the MinHeap actually.

**Task 2b**

It is just a simple test to check whether the task 2a did well.

**Task 3**

Reference the core, it is similar to Task1, but change 4 to arbitrary number.