**Data Structures and Algorithms**

**Lab 12**

**Submitted To:**

Mr. Dilshad Sabir

**Submitted By:**

Manaal Waseem

FA18-BCE-074

**In Lab:**

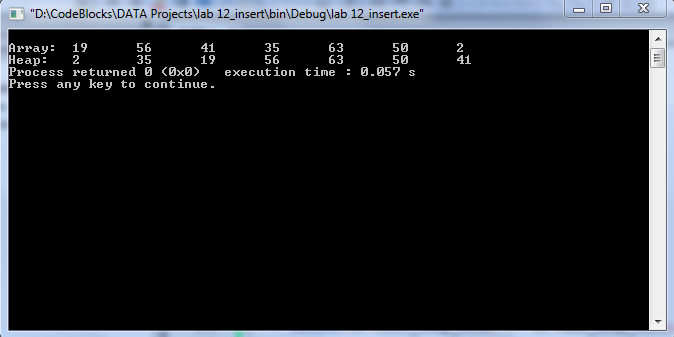
**Task 1:**

***Complete the function ‘void insert\_node(int x, struct heap\_struct \* H)’ in the skeleton code provided.***

**Program:** In this program, function ***“insert\_node”*** inserts nodes from a numeric array into the heap.



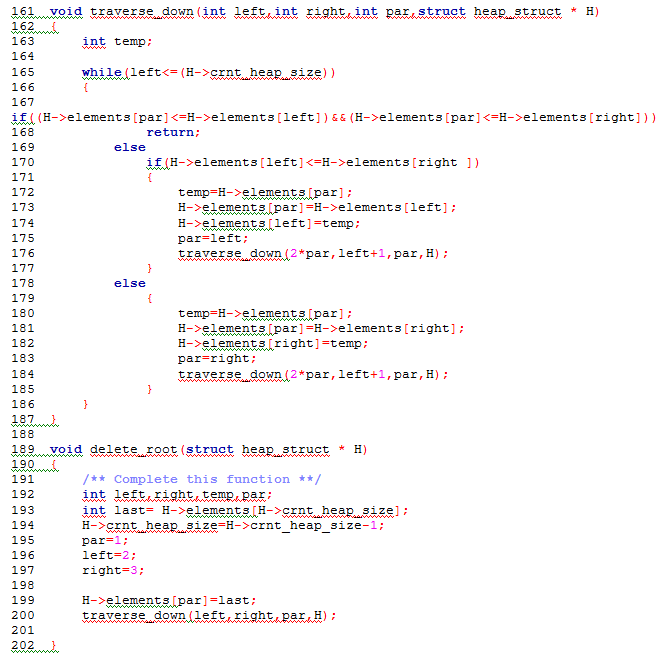
**Output:**



**Task 2:**

***Complete the function ‘void delete\_root(struct heap\_struct \* H)’ in the skeleton code provided.***

**Program:** In this program, function ***“delete\_root”*** deletes the root node from the heap.



**Output:**

****

**Post Lab:**

***Study Binomial Heaps and Fibonacci Heaps.***

* ***Binomial Heaps***

Each node in a binomial heap **H** has a *val* field that stores its value. In addition, each node **N** has following pointers:

* **P[N]** that points to the parent of **N**
* **Child[N]** that points to the leftmost child
* **Sibling[N]** that points to the sibling of N which is immediately to its right

If **N** is the root node, then **P[N] = NULL**. If **N** has no children, then **Child[N] = NULL**, and if **N** is the rightmost child of its parent, then **Sibling[N] = NULL**. In addition to this, every node **N** has a degree field which stores the number of children of **N**.

* ***Fibonacci Heaps***

Fibonacci heaps are especially desirable when the number of extract-minimum and delete operations is small relative to the number of other operations performed. This situation arises in many applications, where algorithms for graph problems may call the decrease-value once per edge. However, the programming complexity of Fibonacci heaps makes them less desirable to use.

A Fibonacci heap is a collection of trees. It is loosely based on binomial heaps. If neither the decrease-value nor the delete operation is performed, each tree in the heap is like a binomial tree. Fibonacci heaps differ from binomial heaps as they have a more relaxed structure, allowing improved asymptotic time bounds.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**THE END**