**Q1:**

**Input: an array of 1000 integers in data.txt**

**Output: an array reflecting a min-heap**

**Task 1: n=20. Print the resulting A to verify the correctness your code.**

**Task 2: n=1000. Print the first 20 values A[1], A[2], ..., A[20] and the last 20 values A[n-19], A[n-18], ..., A[n].**

**=== Pseudo Code ===**

**Main:**

**1. Read the input data of 1000 integers into array A.**

**2. Let n=some value in [1, 1000]**

**3. for i = n/2 down to 1**

**Heapify(i)**

**Heapify(i)**

**1. left ← 2i**

**2. right ← 2i+1**

**3. Find the smallest element among A[left], A[i], and A[right]**

Note: A right child or a left child may not exist!

**Let k denote the index of the smallest element in Step 3.**

**4. if k != i then**

**swap A[i] and A[k]**

**Heapify(k)**

**Q2:**

**Based on results of task 2 in Q1, we then delete elements from and insert elements into the min-heap. You are given the following sequence of integers**

**0 1018 1021 0 1007 0 0 1026 1001 0 0 1014 0.**

**For a positive integer, then insert it into the min-heap.**

**On the other hand, if it is 0, then we are going to extract the root A[1] from the min-heap. Process the insertion/deletion requests one by one.**

**Print the first 20 values A[1], A[2], ..., A[20] and the last 20 values A[n'-19], A[n'-18], ..., A[n'], where n' is the number of elements in the resulting array.**