1) **(3 marks)** Which of the following trees are heaps? Explain your answers.

i)  
```
          18
         /   
        16    12
       /     / 
      11    12  9
```

ii)  
```
          16
         /   
        11    12
       /     / 
      7     9  12
```

iii)  
```
          6
         /   
        10    12
       /     / 
      14    15 
```

2) **(3 marks)** Show the addition of 7 to the following min-heap:

```
          8
         /   
        10    12
       /     / 
      11    34  40
     /     /   
    14    27   59
```

3) **(3 marks)** Show the deletion of the smallest value from the following min-heap:

```
          8
         /   
        10    12
       /     / 
      11    34  40
     /     /   
    14    27   59
```
4) (3 marks) Show the addition of 32 to the following max-heap:

```
   33
  /   \
12    29
 /     / \
10    2   1  3
 /     /   / \
9     2   1  3
```

5. (8 marks) Given a max-heap Java class IntMaxHeap with existing methods

```java
public IntMaxHeap()
public int size()
public void add(int val)
public int getMax(int val)
```

where `add` only accepts values that are greater than zero and `getMax` deletes the largest value stored in the heap and returns that value, write the body of a new method for this class,

```java
public int nth(int n)
```

which returns the `n`th smallest value in the heap if `0 < n ≤ size` and `-1` otherwise. Note that this method must leave the set of elements stored in the heap unchanged (though it can change this set during the operation of the method). For example, given the max-heap in Question (4) above, the values returned by this method when `n = 1, 3, 7, 9,` and 14 are `1, 3, 12, 33,` and `-1,` respectively.