1. Suppose the array shown below represents a binary heap. Show the contents of the array after the minimum value (highest priority) is removed. The value in position 0 of the array is not shown (i.e. 9 is in position 1). Your answer should be an array of values.

9, 22, 25, 60, 31, 40, 45, 65, 70, 47, 52, 43

2. Implement the private method toPostorder in the CompleteBinaryTree class. The implementation must be recursive. Remember for a complete binary tree the tree is stored in an array. Your code must move through the array in the pattern of a binary tree. For example if the array contains the values 30, 20, 40, 10, 15, 7, 45, 90, 4 (position 0 is not used or shown) the value returned by the public method should be "90 4 10 15 20 7 45 40 30". Note the values in the array are a complete binary tree but they are not necessarily a binary heap. This does not matter in your answer.
public class CompleteBinaryTree {
    //implements a complete binary tree stored in an array

    int values[]; //stores the tree
        //values are stored in positions
        //1 through currentSize;

    int currentSize; //the current number of
        //items in the tree

    public CompleteBinaryTree(int s) {
        values = new int[s+1];
        currentSize = 0;
    }

    public String toPostorder() {
        //Returns a string of the ints in the tree in post order
        //a space should follow each int
        return toPostorder(1);
    }

    private String toPostorder(int r) {
        //Returns a string of the ints in the subtree rooted at r
        //in post order. A space should follow each int.
    }
}