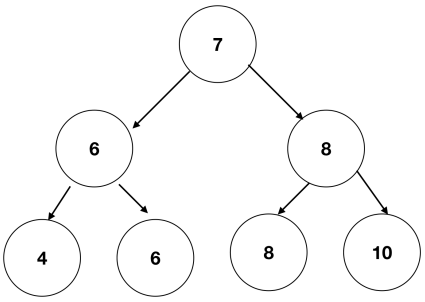
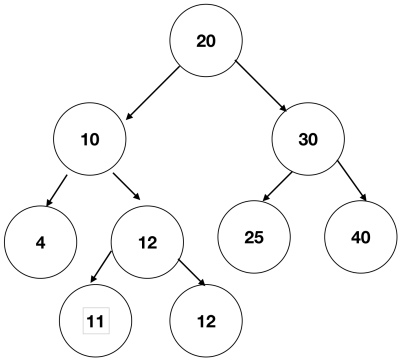
Question 1: 10 Points

Answer each of the following questions about the two trees below:

1. Is it full?
2. Is it complete?
3. Is it perfect?
4. What is the height?
5. Is it a heap?
6. What is the post-order traversal?





1	YES	YES
2	YES	YES
3	YES	NO
4	two	three
5	NO	NO
6	4 6 6 8 10 8 7	4 11 12 12 10 25 40 30 20

Question 2: 10 Points

Given the following array of integers

3, 14, 18, 19, 6, 4, 7, 2, 10, 16, 11, 17, 15, 0, 13, 1, 9, 8, 5, 12

A. Show the array after a quicksort like partition on the number 12.
B. Assuming your partition from part A and a standard quicksort recursive call, show the array after partitioning on 6 and 19

This is one possible partitioning. Others were acceptable.

A 3 6 4 7 2 10 11 0 1 9 8 5 12 14 18 19 16 17 15 13

B 3 4 2 0 1 5 6 7 10 11 9 8 12 14 18 16 17 15 13 19

Question 3: 15 Points

Suppose you have an array-based binary heap with the following data (The data is shown in order in which it appears in the array. That is 18 is in position 0, 15 in position 1, etc):
18, 16, 15, 13, 12, 8, 9, 11, 2, 4, 10, 3, 1, 5, 7, 0, 6
Show the array after each of the following heap operations:

insert(44)
insert (14)
poll()
remove(15)
poll()

Insert 44
44 18 16 15 12 8 9 11 13 4 10 3 1 5 7 0 6 2

Insert 14:
44 18 16 15 12 8 9 11 14 4 10 3 1 5 7 0 6 2 13

Poll
18 16 15 14 12 8 9 11 13 4 10 3 1 5 7 0 6 2

remove 15
18 16 9 14 12 8 7 11 13 4 10 3 1 5 2 0 6

poll
16 14 9 13 12 8 7 11 6 4 10 3 1 5 2 0

Question 4: 20 Points

You are given a function:

```
boolean isInEnglish(String word)
```

which returns true if the string passed into it is an english word and false otherwise.
Write a recursive function,

```
boolean isReducible(String longWord)
```

that returns true if there is a path such that every time you remove a letter, the result is an english word until there are no letters remaining to be removed.
For example, given the input “pips”, then the function would return true since there exists such a path. The path is

pips
pip
pi
i

To remove the ith from a string you can use:

```
String shorterword = word.substring(0,i) + word.substring(i+1);
```

where i is a nonnegative integer and word is a String (for example “pips”).
This function only returns true or false. It need not print anything along the way (that is, you do not need to generate a printout in the style of the example above). You need not write an entire class, just write this method.

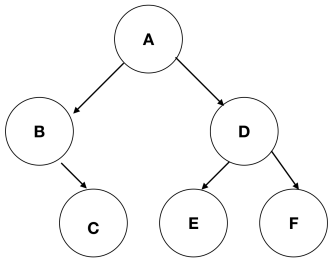
EXTRA CREDIT (3 points): Show the words in the path if the path exists. Show no other words.

```
boolean isReducible(String longWord)
{
 if (longWord.length()==0) return true;
 if (!isInEngligh(longWord)) return false;
for (int i=0; i<longWord.length(); i++)
 {
    String shortWord = longWord.substring(0,i)
                        +longWord.substring(i+1);
    if (isReducible(shortWord))
      return true;
 }
 return false;
}
```

Question 5: 20 Points

Write a method onlyChild that returns a count of the number of nodes in a binary tree that have only one child. For example, in the little tree below right, your function should return 1. You need not write an entire class, just this method. Use the following definition for node. Do not change the node class. You may assume that someone else has built the tree. F

```
public class Node {
      private Node right;
      private Node left;
      public Node() {
          right=null;
          left=null;
      }
      public Node getRight() { return right; }
      public Node getLeft() { return left; }
      public void setRight(Node n) { right = n; }
      public void setLeft(Node n) { left = n; }
}
```




```
public onlyChild(Node n)
{
  if (n==null)
     return 0;
  if (n.getRight()==null) && n.getLeft()==null)
     return 0
  if (n.getRight()==null)
     return 1+onlyChild(n.getLeft());
  if (n.getLeft()==null)
     return 1+onlyChild(n.getRight());
  return onlyChild(n.getRight) + onlyChild(n.getLeft());
}
```

Question 6: 10 Points:

Suppose you have an array-based binary min-heap with *255* elements.

a. At what positions in the array might the third smallest element appear.

b. At what positions in the array might the largest element in the heap appear.

A. The third smallest value can appear in any of positions 2,3,4,5,6 in the array

B. The largest value can appear in positions 127-255

Question 7: 15 Points

Draw a binary tree such that

a. Each node contains a single character
b. A preorder traversal yields “THISISATEST”
c. The tree has height 3.

Typ