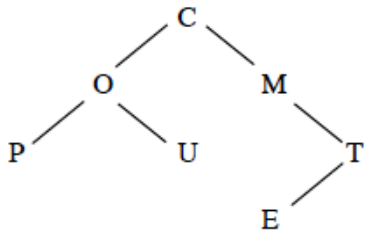
1. [5 points total] Consider the following tree:



a) (3 points) Which of the following apply to this tree (write yes or no beside each item)?

	yes or no
full	_____
complete	_____
nearly complete	_____
BST	_____
binary tree	_____
heap	_____

b) (2 points) Print out the search keys from this tree using a pre-order traversal:

2. [10 points total] For the following questions feel free to use the scratch page for rough work; **make your answers clear**.

8	10	2	4	9	6	12	1	10

a) (2 points) Consider the array above. Draw the tree representing the resulting minimum heap after applying the *heapify* algorithm we discussed in class.

Question 2. *Continued*.

b) (2 points) Next, delete the root and redraw the resulting heap below:

c) (6 points) Finally, apply the heap sort algorithm as we discussed in class. Show the entire, correct heap after each recursive application of the algorithm.

3. [10 points total]

a) (6 points) Using an **order 3** B+ Tree, insert the following key values one by one,in the order shown, showing the state of the tree after each insertion (for simplicity you do not need to draw the sibling pointers). The B+ Tree is initially empty. Use the method shown in class and the lecture notes.

10 112 17 84 2 32

b) (4 points) Given your B+ Tree above, delete 112 and draw the resulting tree.

4. [6 points total]
   a) (5 points) Draw the corresponding binary expression tree using the following:

$$1 * (3 + 4) / 2 - 1 + 10$$

   b) (1 point) After building the tree, print out the tree using Reverse Polish Notation (that is, a post-order traversal).

5. [5 points total] Put the following activites in order from **fastest to slowest**. Write the numbers 1 to 5 beside the activies to indicate this order (1 is the fastest, 5 is the slowest):

   _____Building a heap (heapify)

   _____Searching a BST in the worst case.

   _____Searching a binary tree in the best case.

   _____Searching a heap in the worst case.

   _____Heapsort

6. [4 points total]
   a) (2 points) What is the total minimum number of <key,pointer> pairs (in the leaves) for an entire tree that an order *m* B+ Tree, of height 2, can contain?

   b) (2 points) Given a completely full B+ Tree of order *m*, how many nodes are at a depth of 2? (Remember that the root is at depth 0.)

7. [10 points total] Write a function `heapSearch` with the following specification:

```
typedef string Item_type;
bool heapSearch( Node* root, const Item_type& item );
```

The function should not perform any more comparisons than necessary.

Preconditions: `root` points to the root of a minimum heap; `item` is the key value.

Postconditions: `root`, `item` are unchanged; `true` is returned if the element is found, `false` otherwise.

DO NOT DETACH THIS SHEET.

Scratch space: