

FINAL EXAM REVIEW:

1. Study the following function:

```
int Mystery(int n)
{
    int i, j, sum= 0;
    for(i=0; i < 10; i++)
    {
        for(j=0; j < n; j++)
            sum += i * n;
    }
    return sum;
}
```

What value of sum will be returned when the call Mystery (20) is made? What is the complexity of this function?

2. Solve the recurrence relation

$$T(n) = 2 T(n/2) + 2$$

3. LIN is a linear linked list with more than 2 nodes. Write a function which converts LIN into a circular linked list. The function should return a pointer to the last element of the list.

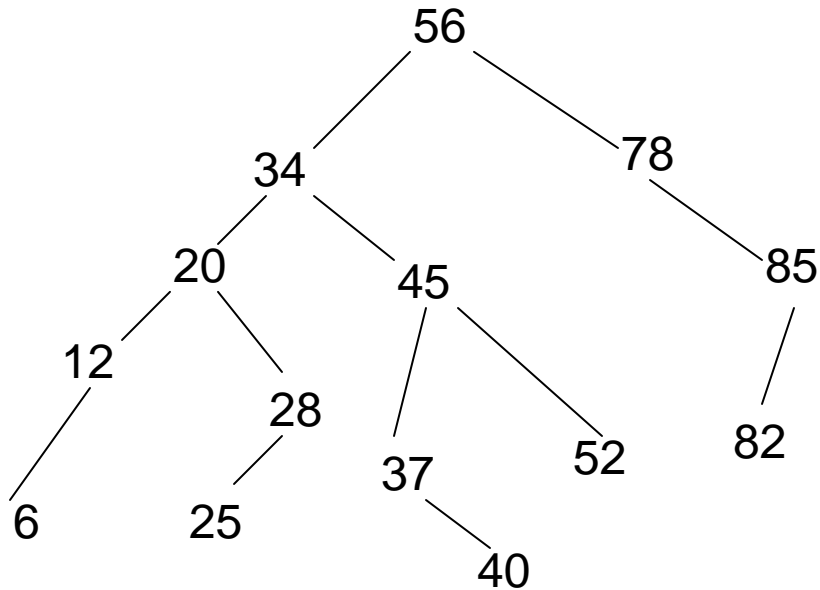
4. Write a function which examines two linear linked lists A and B in memory, removes the first element from B and puts it in front of A.

void shift(struct node ** A, struct node **B)

5. a) Write a recursive routine which counts the number of nodes in a linked list.

5.b) Write a function which deletes the left most node of a binary tree. If the root does not have a left subtree, it deletes the root node. In both cases it returns a pointer to the root of the new tree.

6. Delete the node 34 from the following BST and redraw the tree



7. Convert the decimal number 259.23 into its binary equivalent making use of the octal representation.

8. Convert the following binary number into its corresponding hexadecimal representation.

1 1 0 0 1 0 1 0 1 1 1 0 0 0 1 1 0 1

9. Write a recursive routine which prints the binary equivalent of a decimal number “num”.

10. Execute the algorithm on the tree shown below. Show the exact contents of both the stack and the queue when the algorithm completes execution. Assume that the initial call from the main program is: **P4(root, 60)** and that the tree nodes and pointers are defined as shown. The stack grows from left to right.

```

struct treeNode{
  int data;
  struct treeNode *left, *right;
}
struct treeNode *tree_ptr;
  
```

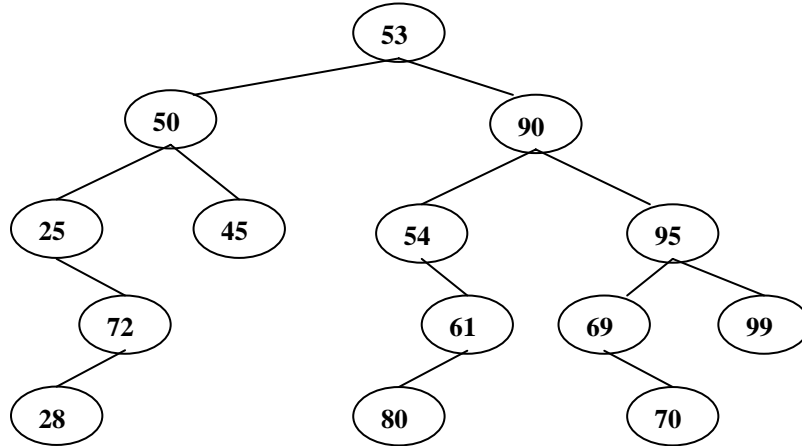
```

void P4(struct tree_ptr *node_ptr, int key) {
    if (node_ptr != NULL){
        if (node_ptr->data <= key){
            push(node_ptr->data);
            P4(node_ptr->right, (key + 20));
            P4(node_ptr->left, (key - 20));
        }
        else {
            enqueue(node_ptr->data);
            P4(node_ptr->left, (key - 10));
            P4(node_ptr->right, (key + 10));
        }
    }
}

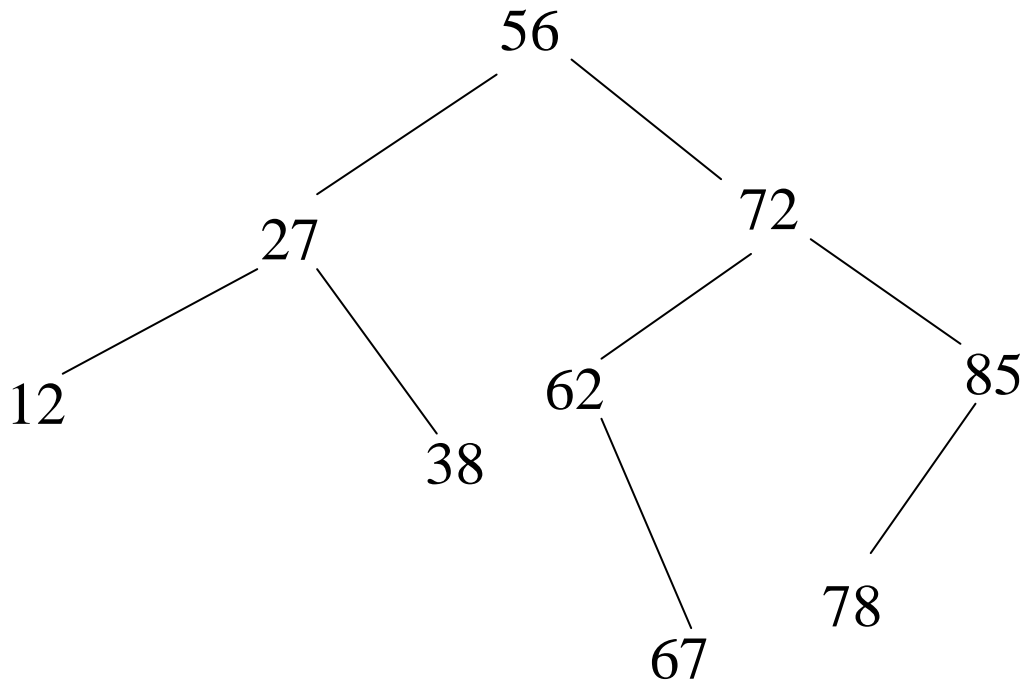
```

The stack grows to the right.

stack										
queue										



11. In the following AVL tree , insert node 93 and then insert 65, so that the tree still remains an AVL tree. Balance if necessary.



12. An array A containing elements in random order is subjected to bubble sort and the result is stored in array B. What would be the time complexity if array B is now subjected to quick sort? Justify in one line.

13. What is the complexity of performing merge sort on data stored as a linked list?

14. A list of numbers are stored in an array. The same list of numbers are also stored in a linked list. Would it be more efficient or less efficient to carry out bubble sort on the linked list? Why?

15. When Merge Sort is applied on the following array show the arrays that are going to be merged whenever the merge routine is called.

[28 67 43 10 2 85 52]

16. Given an array [30 23 5 15 40 89 65 8], indicate the output of the following sorting methods after pass 2:

- i) Selection sort
- ii) insertion sort
- iii) bubble sort