1. [10%] Consider the following algorithm:

Algorithm fun(x, y)

    if (x<y)

        return -3

    else

        return (fun(x-y, y+3)+y)

    end if

end fun

What would be returned if fun is called as

a. (3 %) fun(2, 7)

b. (7 %) fun(15, 3)

2. [10%] Change the following infix expression to postfix expression using a stack:

$$A \ + \ B \ * \ C \ - \ D \ / \ E$$

3. [10%] A binary tree has eight nodes. The postorder and inorder traversals of the tree are given below. Draw the tree.

Postorder: FECHGDBA

Inorder: FCEABHDG

4. [10%] After two passes of a sorting algorithm, the following array:

$$47 \quad 3 \quad 21 \quad 32 \quad 56 \quad 92$$

has been rearranged as show below.

$$3 \quad 21 \quad 47 \quad 32 \quad 56 \quad 92$$

Which sorting algorithm is being used(straight selection, bubble, or straight insertion)? Defend your answer.

5. [10%] If values of A, B, C, and D are 2, 3, 4, and 5, respectively, manually calculate the value of the following expression using queues and show all state transformations of the queues.

prefix: $- \ * \ A \ + \ B \ C \ D$

6. [10%] What is the time complexity of the following code?

```
result = 0;
for (i = 0; i < n; i++)
        result = i+1;
```

7. [10%] Draw a minimum spanning tree of the graph G with the following adjacency matrix representation. The values in the matrix are the cost of the edges.

	0	1	2	3	4
0:	0	6	0	5	2
1:	6	0	0	2	0
2:	0	0	0	6	8
3:	5	2	6	0	0
4:	2	0	8	0	0

8. [10%] Draw the hash table after the insertions of the following keys into the table: 12, 5, 88, 128, 17, 10, 33, 45, 27, 14, 64, and 129. The hash function is h(k) = k mod 13 and the collision is resolved by linear probing.

9. [10%] There are 5 elements stored in an *ordered* linked list. These elements are 7, 9, 2, 5, and 8. Their corresponding memory locations are 2100, 700, 1900, 8500, and 300. Each node is a pair (data, location of next element). Draw the linked list with all the information into the nodes.

10. [10%] Draw the resulting binary search tree with the following successive insertion and deletion operations. The tree is originally empty.
(a) insert 41 (b) insert 51 (c) insert 38 (d) insert 12 (e) insert 27 (f) insert 28 (g) insert 36 (h) insert 19 (i) delete 41 (j) delete 19