1 Hours 05 minutes

1. **Answer the following question (Answer any two question)** 5×2

   a) Show how a stack data type is used to evaluate an expression such as:
   35 /(2+ 3) - 8 / 4

   b) Convert **Q= (A + B)*(C – D) *A* B * C – D + E/F**) into Postfix from showing stack status after every step in tabular form

2. a) Trace the following code. 5×2

```
//MAX=10;

QueueTypeLL<int> q;
    if(q.isEmpty())
    cout<<"Queue is Empty"<<endl;
    else
    cout<<"Queue is not Empty"<<endl;

    q.Enqueue(10);
    q.Enqueue(20);
    q.Enqueue(30);

    if(q.isEmpty())
    cout<<"Queue is Empty"<<endl;
    else
    cout<<"Queue is not Empty"<<endl;

    if(q.isFull())
    cout<<"Queue is Full"<<endl;
    else
    cout<<"Queue is not Full"<<endl;

q.printQueue();//print the full queue

   q.Dequeue();
   q.Dequeue();

q.printQueue();//print the full queue

    return 0;
```

Write the output of the above code?

b) Suppose you have three stacks s1, s2, s2 with starting configuration shown on the left, and finishing condition shown on the right. Give a sequence of push and pop operations that take you from start to finish. For example, to pop the top element of `s1` and push it onto `s3`, you would write `s3.push( s1.pop())`.

start finish

A B

```
B                                                          D
C                                                          A
D                                                          C
---   ---       ---                        ---     ---     ---
s1     s2        s3                         s1      s2      s3
```

3. a) What is Deque? Describe its varieties? 3X2
   b) Write binary search program in a recursive way

```
Int BSEARCH( int a[], l,u,m,key)
{
If(key==a[mid] return mid;
Else if (key<a[mid];
 Bsearch(int a[], l.mid-1, m, key);
Else
  Bsearch (int a[],mid+1, u, m, key);
}
```

4. a) For questions bellow considers the following operations on a Queue data structure that stores 4

```
Queue<int> q ;
q.enqueue(3);
q.enqueue(7);
q.enqueue(8);
q.enqueue(5);
cout<<(q.dequeue( ));          // d1
q.enqueue(6);
cout<<(q.dequeue( ));          // d2
cout<<(q.dequeue( ));          // d3
q.enqueue(2);
q.enqueue(6);
q.enqueue(9);
```

   i) After the code above executes, how many elements would remain in q?
   ii) What value is returned by the last dequeue operation (denoted above with a d3 in comments)?
   iii) print the final queue

5. Construct **a** binary tree from the following traversing sequence? 5
   Inorder sequence: D B E A F C
   Preorder sequence: A B D E C F