1) Write a program to implement Link List with method:

a. Method to insert new node at beginning of LL
b. Method to insert new node at last of LL
c. Method to insert new node at any position in LL
d. Method to delete node from beginning of LL
e. Method to delete node from last of LL.
f. Method to delete node from any position of LL
g. Method to display entire LL

Code:-

MyClass.java

```
16 usages
class Node {          //creating node class with members "data" and "next"
    5 usages
    int data;
    28 usages
    Node next;
    3 usages
    Node(int data){     //initializing data and next
        this.data=data;
        next=null;
    }
}

2 usages
class LinkList{     //creating a class linklist
    19 usages
    Node head;

    //a.Method to insert new node at beginning of LL
    2 usages
    void insertNodeAtFront(int data){
        Node newNode = new Node(data);
        newNode.next = head;
        head = newNode;
    }

    //b.Method to insert new node at last of LL
    3 usages
    void insertNodeAtEnd(int data){
        Node newNode = new Node(data);
        if(head==null){
            head=newNode;
        }
        else{
            Node tmp = head;
            while(tmp.next!=null){
                tmp = tmp.next;
            }
            tmp.next = newNode;
        }
    }

    //c.Method to insert new node at any position in LL
    1 usage
    void insertNodeAtPosition(int data,int pos){

        if(pos<0){
            System.out.println("Enter valid position");
        }
        else{
            Node temp1=head;
            Node temp2=head;

            Node newNode = new Node(data);
```

```
            for(int i=1;i<pos-1;i++){
                if(temp1.next!=null && temp2.next!=null)
                    temp1=temp1.next;
                temp2=temp2.next;
            }
            temp2 = temp1.next;
            temp1.next=newNode;
            newNode.next=temp2;
        }

    }


    //d.Method to delete node from beginning of LL
    1 usage
    void deleteNodeFromFront(){
        if(head==null){
            System.out.println("Link List is Empty \"Underflow Situation\"");
        }
        else{
            Node temp =head;
            System.out.println("Delete Node is : " + head.data);
            head = head.next;
            temp.next=null;
        }
    }


    //f.Method to delete node from any position of LL
    1 usage
    void deleteNodeFromPosition(int pos){
        if(head==null){
            System.out.println("Link List is Empty \"Underflow Situation\"");
        }
        else {
            if (pos < 0) {
                System.out.println("Enter valid Position");
            }
            else{
                Node temp1 = head;
                Node temp2 = head;

                for(int i=1;i<pos-1;i++){
                    if(temp1.next!=null && temp2.next!=null){
                        temp1=temp1.next;

                        temp2=temp2.next;
                    }
                }

                temp2=temp1.next;
                temp1.next=temp2.next;
                System.out.println("Deleted Node is : "+temp2.data);
                temp2.next=null;
            }
        }

    }
    //e.Method to delete node from last of LL.
    1 usage
    void deleteNodeFromEnd(){
        if(head==null){
            System.out.println("Link List is Empty \"Underflow Situation\"");
        }
        else{
            Node temp = head;
            while(temp.next.next!=null){
                temp=temp.next;
            }
            System.out.println("Deleted Node is : " + temp.next.data);
            temp.next=null;
        }
    }

```

```java
    //g.Method to display entire LL
    7 usages
    void displayLinkedList(){
        if(head == null){
            System.out.println("Linked List is Empty");
        }
        else{
            Node tmp = head;
            while(tmp!=null){
                System.out.print(tmp.data + "->");
                tmp=tmp.next;
            }
        }
    }
}


//creating main method to create object of class "LinkList"
public class MyClass {
    public static void main(String[] args){
        LinkList l1 = new LinkList();   //creating object of class "LinkList"

        System.out.println("Displaying when linklist is Empty : ");
        l1.displayLinkedList();

        System.out.println();

        System.out.println("Inserting 10,20,30 one by one at end ");
        l1.insertNodeAtEnd( data: 10);
        l1.insertNodeAtEnd( data: 20);
        l1.insertNodeAtEnd( data: 30);

        System.out.println("Displaying the list with values 10,20,30");
        l1.displayLinkedList();

        System.out.println();
        System.out.println();

        System.out.println("Inserting 500 than 600 to the front of list ");
        l1.insertNodeAtFront( data: 500);
        l1.insertNodeAtFront( data: 600);

        System.out.println("Displaying list after inserting 500 and 600 to the front");
        l1.displayLinkedList();

        System.out.println();
        System.out.println();

        System.out.println("Inserting 400 at pos 3 in current list ");
        l1.insertNodeAtPosition( data: 400, pos: 3);

        System.out.println("Displaying list after inserting 400 at position 3");
        l1.displayLinkedList();

        System.out.println();
        System.out.println();

        System.out.println("Deleting Node from last");
        l1.deleteNodeFromEnd();
        System.out.println("List after deleting node with value 30 from last");
        l1.displayLinkedList();
```

```
178
179             System.out.println();
180             System.out.println();
181
182             System.out.println("Deleting Node from front");
183             l1.deleteNodeFromFront();
184             System.out.println("List after deleting node with value 600 from front");
185             l1.displayLinkedList();
186
187             System.out.println();
188             System.out.println();
189
190             System.out.println("Deleting Node at position 3");
191             l1.deleteNodeFromPosition(3);
192             System.out.println("List after deleting node with value 10 at position 3");
193             l1.displayLinkedList();
194
195         }
196     }
197
```

Output:-

MyClass

```
"C:\Program Files\Amazon Corretto\jdk11.0.15_9\bin\java.exe"
Displaying when linklist is Empty :
Linked List is Empty

Inserting 10,20,30 one by one at end
Displaying the list with values 10,20,30
10->20->30->

Inserting 500 than 600 to the front of list
Displaying list after inserting 500 and 600 to the front
600->500->10->20->30->

Inserting 400 at pos 3 in current list
Displaying list after inserting 400 at position 3
600->500->400->10->20->30->

Deleting Node from last
Deleted Node is : 30
List after deleting node with value 30 from last
600->500->400->10->20->

Deleting Node from front
Delete Node is : 600
List after deleting node with value 600 from front
500->400->10->20->

Deleting Node at position 3
Deleted Node is : 10
List after deleting node with value 10 at position 3
500->400->20->
```

2) Write a program to implement Stack Data Structure with following method:

a. Method to push element
b. Method to pop element
c. Method to display Stack elements

Code:-

MyClass.java

```
2 usages
public class MyClass {
    //declaring variables
    2 usages
    int maxSize = 5;
    3 usages
    int[] stack = new int[maxSize]; //declaring stack array
    8 usages
    int top = -1;

    //a.Method to push element
    6 usages
    void push(int data){
        if(top+1==maxSize){
            System.out.println("Overflow");
        }
        else{
            top++;
            stack[top] = data;
        }

    }

    //b.Method to pop element
    3 usages
    void pop(){
        if(top==-1){
            System.out.println("Underflow");
        }
        else{
            System.out.println("Element " + stack[top] +" is deleted");
            top--;
        }

    }

    //c.Method to display stack element
    3 usages
    void displayStack(){
        if(top==-1){
            System.out.println("Stack is Empty");
        }
        else{
            for(int i=0;i<=top;i++){
                System.out.print(stack[i] + " ");
            }
            System.out.println();
        }

    }

    //creating main method to create object of class
    public static void main(String[] args){
        MyClass st = new MyClass();
        st.displayStack();
        st.pop();
        st.push( data: 10);
        st.push( data: 20);
        st.push( data: 30);
        st.push( data: 40);
        st.push( data: 50);
        st.push( data: 60);
        st.displayStack();
        st.pop();
        st.pop();
        st.displayStack();
    }
}
```

Output:-

```
MyClass
"C:\Program Files\Amazon Corretto\jdk11.0.15_9\bin\java.exe"
Stack is Empty
Underflow
Overflow
10 20 30 40 50
Element 50 is deleted
Element 40 is deleted
10 20 30

Process finished with exit code 0
```

3) Write a Program to implement Bubble Sort.

Code:-

MyClass.java

```
2 usages
public class MyClass {
    17 usages
    int[] myArray = new int[5]; //declaring a myArray
    1 usage
    MyClass(){      //initialising myArray
        myArray[0]=5;
        myArray[1]=45;
        myArray[2]=0;
        myArray[3]=9;
        myArray[4]=-15;
    }

    //method to bubble sort
    1 usage
    void bubbleSort(){
        int temp,step=0;
        for(int i=0;i< myArray.length-1;i++){
            step++;
            System.out.println("Iteration "+step);
            for(int j=0;j<myArray.length-1-i;j++){
                if(myArray[j+1]<myArray[j]){
                    temp = myArray[j+1];
                    myArray[j+1] = myArray[j];
                    myArray[j]=temp;
                }

                for(int k=0;k<myArray.length;k++){
                    System.out.print(myArray[k]+" -> ");
                }
                System.out.println();
            }
            System.out.println();
        }
        for(int i=0;i<myArray.length;i++){
            System.out.print(myArray[i]+" ");
        }

    }

    //creating main method to create object of class
    public static void main(String[] args){
        MyClass obj = new MyClass();
        obj.bubbleSort();
    }
}

```

Output:-

```
MyClass
"C:\Program Files\Amazon Corretto\jdk11.0.15_9\bin\java.exe"
Iteration 1
5 -> 45 -> 0 -> 9 -> -15 ->
5 -> 0 -> 45 -> 9 -> -15 ->
5 -> 0 -> 9 -> 45 -> -15 ->
5 -> 0 -> 9 -> -15 -> 45 ->

Iteration 2
0 -> 5 -> 9 -> -15 -> 45 ->
0 -> 5 -> 9 -> -15 -> 45 ->
0 -> 5 -> -15 -> 9 -> 45 ->

Iteration 3
0 -> 5 -> -15 -> 9 -> 45 ->
0 -> -15 -> 5 -> 9 -> 45 ->

Iteration 4
-15 -> 0 -> 5 -> 9 -> 45 ->

-15 0 5 9 45
Process finished with exit code 0
```

4) Write a program to implement Selection Sort.

Code:-

MyClass.java

```
2 usages
public class MyClass {
    15 usages
    int[] myArray = new int[5];     //declaring array

    1 usage
    MyClass(){          //initialising array
        myArray[0]=7;
        myArray[1]=8;
        myArray[2]=6;
        myArray[3]=5;
        myArray[4]=1;

    }

    //method of selection sort
    1 usage
    void selectionSort(){
        int min;
        int minIndex;
        for(int i=0;i<myArray.length;i++){
            min = myArray[i];
            minIndex=i;

            //code to print each pass
            for(int k=0;k<myArray.length;k++){
                System.out.print(myArray[k] + " -> ");
            }

```

```
            for(int j=i+1;j<myArray.length;j++){
                if(myArray[j]<min){
                    min=myArray[j];
                    minIndex = j;
                }
            }
            myArray[minIndex] = myArray[i];
            myArray[i]= min;

            System.out.println();

        }
    }

    //creating main method to create object of class
    public static void main(String[] args){
        MyClass obj = new MyClass();
        obj.selectionSort();
    }
}

```

Output:-

```
MyClass
"C:\Program Files\Amazon Corretto\jdk11.0.15_9\bin\java.exe"
7 -> 8 -> 6 -> 5 -> 1 ->
1 -> 8 -> 6 -> 5 -> 7 ->
1 -> 5 -> 6 -> 8 -> 7 ->
1 -> 5 -> 6 -> 8 -> 7 ->
1 -> 5 -> 6 -> 7 -> 8 ->

Process finished with exit code 0
```

5) Write a program to implement Insertion Sort

Code:-

MyClass.java

```
2 usages
public class MyClass {
    12 usages
    int[] myArray = new int[6];     //declaring array

    1 usage
    MyClass(){          //initialising array
        myArray[0] = 7;
        myArray[1] = 4;
        myArray[2] = 15;
        myArray[3] = 1;
        myArray[4] = 8;
        myArray[5] = 2;
    }

    //method of insertion sort
    1 usage
    void insertionSort(){
        int temp;
        for(int i=0;i<myArray.length-1;i++){
            temp = myArray[i+1];
            for(int j=i;j>-1;j--){
                if(myArray[j]>temp) {
                    myArray[j + 1] = myArray[j];
                    myArray[j] = temp;
                }
                else{
                    break;
                }
            }
        }
```

```
    }

    //creating main method to create object of class
    public static void main(String[] args){
        MyClass obj = new MyClass();
        obj.insertionSort();
    }
}

```

Output:-

```
MyClass
"C:\Program Files\Amazon Corretto\jdk11.0.15_9\bin\java.exe"
4 -> 7 -> 15 -> 1 -> 8 -> 2 ->
4 -> 7 -> 1 -> 15 -> 8 -> 2 ->
4 -> 1 -> 7 -> 15 -> 8 -> 2 ->
1 -> 4 -> 7 -> 15 -> 8 -> 2 ->
1 -> 4 -> 7 -> 8 -> 15 -> 2 ->
1 -> 4 -> 7 -> 8 -> 2 -> 15 ->
1 -> 4 -> 7 -> 2 -> 8 -> 15 ->
1 -> 4 -> 2 -> 7 -> 8 -> 15 ->
1 -> 2 -> 4 -> 7 -> 8 -> 15 ->

Process finished with exit code 0
```

6) Write a program to implement Linear Search.

Code:-

```
MyClass.java
2 usages
public class MyClass {

    2 usages
    void linearSearch(int[] myArray,int data){
        int flag=0;
        for(int i=0;i< myArray.length;i++){
            if(data == myArray[i]) {
                System.out.println("Data found at index : " + i);
                flag = 1;
            }
        }

        if(flag==0){
            System.out.println("Data not in Array");
        }
    }

    public static void main(String[] args){
        MyClass obj = new MyClass();
        int[] myArray = {30,54,12,66,49,66};
        obj.linearSearch(myArray, data: 66);
        obj.linearSearch(myArray, data: 10);
    }
}

```

Output:-

```
MyClass
"C:\Program Files\Amazon Corretto\jdk11.0.15_9\bin\java.exe"
Data found at index : 3
Data found at index : 5
Data not in Array

Process finished with exit code 0
```

7) Write a program to implement Binary Search.

Code:-

MyClass.java

```
public class MyClass {
    2 usages
    void binarySearch(int[] myArray,int data){
        int start = 0;
        int end = myArray.length-1;
        int flag=0;

        while(start<=end) {
            int mid = (start + end) / 2;
            if (data == myArray[mid]) {
                System.out.println("Data found at Index : " + mid);
                flag=1;
                break;
            } else if (data < myArray[mid]) {
                end = mid - 1;
            } else if (data > myArray[mid]) {
                start = mid + 1;
            }
        }
        if(flag==0){
            System.out.println("Data not in List");
        }
    }
    public static void main(String[] args){
        MyClass obj = new MyClass();
        int[] myArray = {15,20,30,35,40,50};
        obj.binarySearch(myArray, data: 40);
        obj.binarySearch(myArray, data: 10);
    }
}
```

Output:-

MyClass

```
"C:\Program Files\Amazon Corretto\jdk11.0.15_9\bin\java.exe"
Data found at Index : 4
Data not in List

Process finished with exit code 0
```

8) Write a program to implement Quick Sort.

Code:-

MyClass.java

```
2 usages
public class MyClass {
    //method to partition array
    1 usage
    int partition(int[] arr,int lb, int ub){

        int pivot = arr[lb];
        int start = lb;
        int end = ub;
        int temp;

        while(start<end){
            while(arr[start]<=pivot){
                start++;
            }

            while(arr[end]>pivot){
                end--;
            }
```

```
18
19              if(start<end){
20                  temp = arr[start];
21                  arr[start] = arr[end];
22                  arr[end] = temp;
23              }
24          }
25
26          arr[lb] = arr[end];
27          arr[end] = pivot;
28          return end;
29      }
30
31      //method to quickSort the array
        3 usages
32      void quickSort(int arr[],int lb,int ub){
33          if(lb<ub){
34              int pos = partition(arr,lb,ub);
35              quickSort(arr,lb, ub: pos-1);
36              quickSort(arr, lb: pos+1,ub);
37          }
38      }
39
40      //main method to create object of class
41      public static void main(String[] args){
42          MyClass obj = new MyClass();
43          int[] arr = {35,82,11,74,23,21,89};
44          obj.quickSort(arr, lb: 0, ub: arr.length-1);  //calling quickSort method
45          System.out.print("Sorted Array is : ");
46          for(int i=0;i<arr.length;i++){
47              System.out.print(arr[i]+" ");
48          }
49      }
50  }
51
```

Output:-

```
MyClass
"C:\Program Files\Amazon Corretto\jdk11.0.15_9\bin\java.exe"
Sorted Array is : 11 21 23 35 74 82 89
Process finished with exit code 0
```

Thank You