1. To write a Java Program to design an interface for Stack ADT and implement Stack ADT using both Array and Linked List.

```
import java.io.*;
interface Mystack
{
   public void pop();
   public void push();
   public void display();
}
class Stack_array implements Mystack
{
   final static int n=5;
   int stack[]=new int[n];
   int top=-1;
   public void push()
   {
      try
      {
         BufferedReader br=new BufferedReader(new InputStreamReader(System.in));
         if(top==(n-1))
         {
            System.out.println(" Stack Overflow");
            return;
         }
         else
         {
            System.out.println("Enter the element");
            int ele=Integer.parseInt(br.readLine());
            stack[++top]=ele;
         }
      }
      catch(IOException e)
      {
         System.out.println("e");
      }
   }
   public void pop()
   {
      if(top<0)
      {
         System.out.println("Stack underflow");
         return;
      }
```

```
        else
        {
            int popper=stack[top];
            top--;
            System.out.println("Popped element:" +popper);
        }
    }

    public void display()
    {
        if(top<0)
        {
            System.out.println("Stack is empty");
            return;
        }
        else
        {
            String str=" ";
            for(int i=0; i<=top; i++)
                str=str+"  "+stack[i]+" <--";
            System.out.println("Elements are:"+str);
        }
    }
}
class Link
{
    public int data;
    public Link nextLink;
    public Link(int d)
    {
        data= d;
        nextLink=null;
    }
    public void printLink()
    {
        System.out.print(" --> "+data);
    }
}
class Stack_List  implements Mystack
{
    private Link first;
    public Stack_List()
    {
        first = null;
    }
    public boolean isEmpty()
```

```
{
   return first == null;
}
public void push()
{
   try
   {
      BufferedReader br=new BufferedReader(new InputStreamReader(System.in));
      System.out.println("Enter the element");
      int ele=Integer.parseInt(br.readLine());
      Link link = new Link(ele);
      link.nextLink = first;
      first = link;
   }
   catch(IOException e)
   {
      System.err.println(e);
   }
}
public Link delete()
{
   Link temp = first;
   try
   {
      first = first.nextLink;
   }
   catch(NullPointerException e)
   {
      throw e;
   }
   return temp;
}
public void pop()
{
   try
   {
      Link deletedLink = delete();
      System.out.println("Popped: "+deletedLink.data);
   }
   catch(NullPointerException e)
   {
      throw e;
   }
}
public void display()
{
      if(first==null)
         System.out.println("Stack is empty");
      else
      {
         Link currentLink = first;
         System.out.print("Elements are: ");
         while(currentLink != null)
         {
            currentLink.printLink();
            currentLink = currentLink.nextLink;
         }
         System.out.println("");
      }
   }
}
class StackADT
{
   public static void main(String arg[])throws IOException
   {
      BufferedReader br=new BufferedReader(new InputStreamReader(System.in));
      System.out.println("Implementation of Stack using Array");
      Stack_array stk=new Stack_array();
      int ch=0;
      do
      {
         System.out.println("1.Push 2.Pop 3.Display 4.Exit 5.Use Linked List");
         System.out.println("Enter your choice:");
         ch=Integer.parseInt(br.readLine());
         switch(ch)
         {
         case 1:
            stk.push();
            break;
         case 2:
            stk.pop();
            break;
         case 3:
            stk.display();
            break;
         case 4:
            System.exit(0);
         }
      }
      while(ch<5);
      System.out.println("Implementation of Stack using Linked List");
      Stack_List stk1=new Stack_List();
```

```
        ch=0;
        do
        {
            System.out.println("1.Push 2.Pop 3.Display 4.Exit");
            System.out.println("Enter your choice:");
            ch=Integer.parseInt(br.readLine());
            switch(ch)
            {
            case 1:
                stk1.push();
                break;
            case 2:
                try
                {
                    stk1.pop();
                }
                catch(NullPointerException e)
                {
                    System.out.println("Stack underflown");
                }
                break;
            case 3:
                stk1.display();
                break;
            default:
                System.exit(0);

            }
        }
        while(ch<5);
    }
}
```

OUTPUT:

```
Implementation of Stack using Array
1.Push 2.Pop 3.Display 4.Exit 5.Use Linked List
Enter your choice:
1
Enter the element
10
1.Push 2.Pop 3.Display 4.Exit 5.Use Linked List
```

```
Enter your choice:
1
Enter the element
15
1.Push 2.Pop 3.Display 4.Exit 5.Use Linked List
Enter your choice:
1
Enter the element
25
1.Push 2.Pop 3.Display 4.Exit 5.Use Linked List
Enter your choice:
3
Elements are:   10 <--  15 <--  25 <--
1.Push 2.Pop 3.Display 4.Exit 5.Use Linked List
Enter your choice:
5
Implementation of Stack using Linked List
1.Push 2.Pop 3.Display 4.Exit
Enter your choice:
1
Enter the element
10
1.Push 2.Pop 3.Display 4.Exit
Enter your choice:
1
Enter the element
15
1.Push 2.Pop 3.Display 4.Exit
Enter your choice:
1
Enter the element
20
1.Push 2.Pop 3.Display 4.Exit
Enter your choice:
3
Elements are:  --> 20 --> 15 --> 10
1.Push 2.Pop 3.Display 4.Exit
Enter your choice:
2
Popped: 20
1.Push 2.Pop 3.Display 4.Exit
Enter your choice:
3
Elements are:  --> 15 --> 10
1.Push 2.Pop 3.Display 4.Exit
Enter your choice:
```

```
4
```