```java
import java.util.Scanner;

class List {
    class Node {
        int data;
        Node next;

        Node(int d) {
            data = d;
            next = null;
        }
    }

    Node root;

    List() {
        root = null;
    }

    public void insertLeft(int e) {
        Node n = new Node(e);
        if (root == null) {
            root = n;
        } else {
            n.next = root;
            root = n;
        }
    }

    public void insertRight(int e) {
        Node n = new Node(e);
        if (root == null) {
            root = n;
        } else {
            Node t = root;
            while (t.next != null) {
                t = t.next;
                t.next = n;
            }
        }
    }

    public void deleteLeft() {
        if (root == null) {
            System.out.println("Link List Empty");
        } else {
            Node t = root;
            root = root.next;
            System.out.println(t.data + "Remove");
        }
    }

    public void deleteRight() {
        if (root == null) {
            System.out.println("Linked List Empty");
        } else {
            Node t = root;
            Node t2 = t;
            while (t.next != null) {
                t2 = t;
                t = t.next;
            }
```

```
63              t2.next = null;
64              System.out.println(t.data + "Remove");
65          }
66      }
67
68      public void printLink() {
69          if (root == null) {
70              System.out.println("Empty List");
71          } else {
72              Node t = root;
73              while (t != null) {
74                  System.out.println(t.data);
75                  t = t.next;
76              }
77          }
78
79      }
80
81      public boolean SearchKey(int key) {
82          if (root == null) {
83              System.out.println("Empty List");
84          } else {
85              Node t = root;
86              while (t != null) {
87                  if (t.data == key)
88                      return (true);
89                  t = t.next;
90              }
91          }
92          return (false);
93
94      }
95
96      public void deletekey(int key) {
97          if (root == null) {
98              System.out.println("Empty List");
99          } else {
100             Node t = root;
101             Node t2 = root;
102             while (t != null && t.data != key) {
103                 t2 = t;
104                 t = t.next;
105             }
106             if (t == null) {
107                 System.out.println("Not found");
108             } else {
109                 if (t == root) {
110                     root = root.next;
111                 } else if (t.next == null) {
112                     t2.next = null;
113                 } else {
114                     t2.next = t.next;
115                 }
116                 System.out.println(t.data + "Data Remove");
117             }
118         }
119     }
120 }
121
122 public class LinkList {
123     public static void main(String args[]) {
124         int val, ch;
```

```
125         Scanner s = new Scanner(System.in);
126         List q = new List();
127 
128         do {
129             System.out.println(
130                     "\n1.Insert Left \n2.Insert Right \n3.Delete Left \n4.Delete Right
    \n5.Search Key Del \n6.Print Link List \n Enter choice :");
131             ch = s.nextInt();
132             switch (ch) {
133             case 1:
134                 System.out.println("Enter Left Node");
135                 val = s.nextInt();
136                 q.insertLeft(val);
137                 break;
138             case 2:
139                 System.out.println("Enter Right Node");
140                 val = s.nextInt();
141                 q.insertRight(val);
142                 break;
143             case 3:
144                 System.out.println("Enter Delete Left Node");
145                 q.deleteLeft();
146                 break;
147             case 4:
148                 System.out.println("Enter Delete Right Node");
149                 q.deleteRight();
150                 break;
151             case 5:
152                 System.out.println("Enter Search Key Delete Node");
153                 val = s.nextInt();
154                 q.SearchKey(val);
155                 break;
156             case 6:
157                 System.out.println("Print LinkList");
158                 q.printLink();
159                 break;
160             }
161         } while (ch != 0);
162     }
163 }
```

```java
import java.util.Scanner;

class Stack
{
    class Node
    {
        int data;
        Node next;

        Node(int d) {
            data = d;
            next = null;
    }
    }
    Node tos;
    Stack()
    {
        tos=null;
    }
    public void pushStack(int e)
    {
        Node n=new Node(e);
        if(tos==null)
        {
            tos=n;
        }
        else
        {
            n.next=tos;
            tos=n;

        }
    }
    public void popStack()
    {
        if(tos==null)
         {
            System.out.println("Empty Stack");
         }
        else
        {
            Node t=tos;
            tos=tos.next;
            System.out.println(t.data+ "Data poped");
        }
    }

    public void printStack()
    {
        for(Node t=tos;t!=null;t=t.next)
        {
            System.out.println(t.data);
        }
    }
}
public class StackLink {
    public static void main(String args[])
    {
        int val, ch;
        Scanner s = new Scanner(System.in);
        Stack p = new Stack();
        do {
```

```
63              System.out.println(
64                      "\n1.Push Stack \n2.Pop Stack \n3.Print Stack \nEnter choice :");
65              ch = s.nextInt();
66          switch (ch) {
67              case 1:
68                  System.out.println("Enter Push Element");
69                  val = s.nextInt();
70                  p.pushStack(val);
71                  break;
72              case 2:
73                  System.out.println("Enter Pop Element");
74                  p.popStack();
75                  break;
76              case 3:
77                  System.out.println("Print Stack");
78                  p.printStack();
79                  break;
80
81          }
82          }while(ch!=0);
83      }
84
85  }
86
```

```java
import java.util.Scanner;

class Queue {
    class Node {
        int data;
        Node next;

        Node(int d) {
            data = d;
            next = null;
        }
    }

    Node front, rear;

    Queue() {
        rear = front = null;

    }

    public void enQueue(int e) {
        Node n = new Node(e);
        if (front == null) {
            front = rear = n;
        } else {
            rear.next = n;
            rear = n;
        }
    }

    public void deQueue() {
        if (front == null) {
            System.out.println("Queue is Empty");
        } else {
            Node t = front;
            front = front.next;
            System.out.println(t.data + "Data Removed");
        }
    }

    public void printQueue() {
        for (Node t = front; t != null; t = t.next) {
            System.out.println(t.data);
        }
    }
}

public class QueueList {
    public static void main(String args[]) {
        int val, ch;
        Scanner s = new Scanner(System.in);
        Queue q = new Queue();
        do {
            System.out.println("\n1.enQueue \n2.deQueue \n3.Print Queue \nEnter choice :");
            ch = s.nextInt();
        switch (ch) {
            case 1:
                System.out.println("Enter enQueue Element");
                val = s.nextInt();
                q.enQueue(val);
                break;
            case 2:
```

```
63                  System.out.println("Enter deQueue Element");
64                  q.deQueue();
65                  break;
66              case 3:
67                  System.out.println("Print Queue");
68                  q.printQueue();
69                  break;
70
71              }
72
73          } while (ch != 0);
74      }
75  }
76
77
78
```