

```

import java.util.Scanner;

// Class to Represent a Node
class Node
{
    int info;
    Node next;
}

// Class to Represent the Singly Linked List
public class SinglyLinkedList
{

    public static Node head=null;

    // Create a Singly Linked List
    public static void create()
    {
        Scanner sc=new Scanner (System.in);
        Node p=new Node();
        System.out.print("Input info ");
        p.info=sc.nextInt();
        p.next=null;
        head=p;

        //Adding Remaining Nodes (if any)
        System.out.println("Do you want more nodes(y/n)");
        char ch=sc.next().charAt(0);
        while(ch!='n')
        {
            Node q=new Node();
            System.out.print("Input info ");
            p.info=sc.nextInt();
            q.next=null;
            p.next=q;
            p=q;
            System.out.println("Do you want more nodes(y/n)");
            ch=sc.next().charAt(0);
        }

        // Display the Content of the Singly Linked List
        public static void display()
        {
            Node temp = head;

            while(temp!=null)

```

```

        {
            System.out.print(temp.info+"--->");
            temp=temp.next;
        }
        System.out.print("null \n");
    }

    //Count the no. of nodes in the Singly Linked List
    public static int count()
    {
        int count=0;
        Node temp=head;
        while(temp!=null)
        {
            count = count + 1;
            temp=temp.next;
        }
        return count;
    }

    //Search a Particular Element of the Singly Linked List
    public static int search(int x)
    {
        int pos=0;
        Node temp=head;
        while(temp!=null)
        {
            pos++;
            if(temp.info==x)
                return pos;
            temp=temp.next;
        }
        return -1;
    }

    // Insert at the Beginning of the Singly Linked List
    public static void insert_beg()
    {
        Scanner sc=new Scanner (System.in);
        Node p=new Node();
        System.out.print("Input info ");
        p.info=sc.nextInt();
        p.next=head;
        head=p;
    }

```

```
// Insert at the End of the Singly Linked List
```

```
public static void insert_end()  
{  
    Scanner sc=new Scanner (System.in);  
    Node p=new Node();  
    System.out.print("Input info ");  
    p.info=sc.nextInt();  
    if(head==null)  
    {  
        head=p;  
    }  
    else  
    {  
        Node temp=head;  
        while(temp.next!=null)  
        {  
            temp=temp.next;  
        }  
        temp.next=p;  
    }  
}
```

```
// Insert at a Particular Position of the Singly Linked  
List
```

```
public static void insert_pos(int pos)  
{  
    Scanner sc=new Scanner (System.in);  
    int c=count();  
    if(pos>=1&&pos<=c+1)  
    {  
        if(pos==1)  
        {  
            insert_beg();  
        }  
        else if(pos==c+1)  
        {  
            insert_end();  
        }  
        else  
        {  
            Node p=new Node();  
            System.out.print("Input info ");  
            p.info=sc.nextInt();  
            Node temp=head;  
            int cnt=1;  
            while(cnt<pos-1)
```

```

        {
            cnt++;
            temp=temp.next;
        }
        p.next=temp.next;
        temp.next=p;
    }
}
else
{
    System.out.println("Invalid Position");
}
}

// Delete at the Beginning of the Singly Linked List
public static void delete_beg()
{
    if(head == null)
    {
        System.out.println("Underflow");
        return;
    }

    head=head.next;
}

// Delete at the End of the Singly Linked List
public static Node delete_end()
{
    Node temp=head;
    if(head==null)
    {
        System.out.println("Underflow");
        return null;
    }
    if(head.next==null)
    {
        head=null;

        return temp;
    }
    else
    {

```

```

        while(temp.next.next!=null)
        {
            temp=temp.next;
        }
        temp.next=null;
    }
    return temp;
}

// Delete at a Particular Position of the Singly Linked
List
public static void delete_pos()
{
    if(head==null)
    {
        System.out.println("Underflow");
        return;
    }

    Scanner sc=new Scanner (System.in);
    System.out.println("Enter the position");
    int pos=sc.nextInt();
    int c=count();
    if(pos<=c)
    {
        if(pos==1)
        {
            delete_beg();
        }
        else if(pos==c)
        {
            delete_end();
        }
        else
        {
            Node temp=head;
            int cnt=1;
            while(cnt<pos-1)
            {
                cnt++;
                temp=temp.next;
            }
            temp.next=temp.next.next;
        }
    }
}

```

```

    }
    else
    {
        System.out.println("Invalid Position");
    }
}

// Reverse the Singly Linked List
public static void reverse()
{
    Node pvs = null;
    Node cur = head;
    Node nxt = null;

    while (cur != null)
    {
        nxt = cur.next;
        cur.next = pvs;
        pvs = cur;
        cur = nxt;
    }
    head = pvs;
}

// Sort the Singly Linked List
public static void sort()
{
    Node cur = head, index = null;

    if (head == null) {
        return;
    }
    else {
        while (cur != null) {

            index = cur.next;

            while (index != null) {

                if (cur.info > index.info) {
                    swap(cur, index);
                }
            }
        }
    }
}

```

```

        index = index.next;
    }
    cur = cur.next;
}
}

// Swap the information of two nodes
public static void swap(Node cur, Node in)
{
    int temp_r;
    temp_r = cur.info;
    cur.info = in.info;
    in.info = temp_r;
}

public static void main(String[] args) {
    Scanner sc = new Scanner (System.in);

    while(true)
    {
        System.out.println("****MENU****");
        System.out.println("0:Exit");
        System.out.println("1:Creation");
        System.out.println("2:Display");
        System.out.println("3:Count");
        System.out.println("4:Search");
        System.out.println("5:Insert");
        System.out.println("6:Delete");
        System.out.println("7:Reverse");
        System.out.println("8:Sort");
        System.out.println("*****");
        System.out.println("Enter the choice");
        int choice = sc.nextInt();
        switch(choice)
        {
            case 0:
                System.exit(0);
            case 1:
                create();
                break;
            case 2:
                display();
                break;
            case 3:

```

```

        System.out.println("No. of Nodes =" + count());
        break;
    case 4:
        System.out.println("Enter info to be searched");
        int e = sc.nextInt();
        int pos = search(e);
        if (pos == -1)
            System.out.println("Searched info not
present");
        else
            System.out.println("Info present at position
" + pos);
        break;
    case 5:
        System.out.println("****INSERT****");
        System.out.println("1: Begning");
        System.out.println("2: End");
        System.out.println("3: Specific Position");
        System.out.println("Enter the choice");
        int ch = sc.nextInt();
        switch (ch)
        {
            case 1:
                insert_beg();
                break;
            case 2:
                insert_end();
                break;
            case 3:
                System.out.println("Enter the position");
                pos = sc.nextInt();
                insert_pos(pos);
                break;
            default:
                System.out.println("Wrong choice");
                break;
        }
        break;
    case 6:
        System.out.println("****DELETE****");
        System.out.println("1: Begning");
        System.out.println("2: End");
        System.out.println("3: Specific Position");
        System.out.println("Enter the choice");
        ch = sc.nextInt();
        switch (ch)
        {

```



```

        case 1:
            delete_beg();
            break;
        case 2:
            Node t =delete_end();
            System.out.println(t.info);
            break;
        case 3:
            delete_pos();
            break;
        default:
            System.out.println("Wrong choice");
            break;
    }
    break;
case 7:
    reverse();
    break;
case 8:
    sort();
    break;
default:
    System.out.println("Wrong choice");
}
}
}
}

```

## Queue Using Linked List

```
import java.util.Scanner;

class Node {
    int info;
    Node link;
}

public class QueueLinkedList
{
    static Node front=null;
    static Node rear=null;

    //Inserting an element to the Queue
    public static void enqueue(int x)
    {
        Node p = new Node();
        p.info = x;
        p.link = null;

        if(rear==null)
            front=rear=p;
        else
        {
            rear.link = p;
            rear=p;
        }
    }

    //Removing an element from the Queue
    public static void dequeue()
    {
        if(front==null)
        {
            System.out.println("Queue Underflow ");
            return;
        }
        System.out.println("Deleted value is:"+front.info);
        front=front.link;
        if(front==null)
            rear=null;
    }

    //Displaying the Queue elements
    public static void display()
    {
        if(front==null)
        {
            System.out.println("Queue Underflow ");
            return;
        }
        Node temp=front;
        while(temp!=null)
        {
            System.out.print(temp.info+"-->");
            temp=temp.link;
        }
    }
}
```

```
        System.out.println("null");  
    }
```

```
public static void main(String[] args) {  
    Scanner sc=new Scanner (System.in);  
  
    while(true)  
    {  
        System.out.println("****MENU****");  
        System.out.println("0:Exit");  
        System.out.println("1:Enqueue");  
        System.out.println("2:Dequeue");  
        System.out.println("3:Display");  
  
        System.out.println("*****");  
        System.out.println("Enter the choice");  
        int choice=sc.nextInt();  
        switch(choice)  
        {  
            case 0:  
                System.exit(0);  
            case 1:  
                System.out.println("Enter the element:");  
                int e=sc.nextInt();  
                enqueue(e);  
                break;  
            case 2:  
                dequeue();  
                break;  
            case 3:  
                display();  
                break;  
            default:  
                System.out.println("Wrong choice");  
        }  
    }  
}
```

```
import java.util.Scanner;

public class QueueUsingArray {
    int arr[];
    int front,rear;
    QueueUsingArray(int max)
    {
        arr=new int[max];
        front=rear=-1;
    }

    void enqueue(int ele)
    {
        if(is_full())
        {
            System.out.println("Queue Overflow");
            return;
        }
        else if(front==-1&&rear==-1)
            front=rear=0;
        else
            rear=rear+1;

        arr[rear]=ele;
    }

    void dequeue()
    {
        if(is_empty())
        {
            System.out.println("Queue Underflow");
            return;
        }
        else
        {
            System.out.println("Deleted element:"+arr[front]);
            if(front==rear)
                front=rear=-1;
            else
                front=front+1;
        }
    }

    void display()
    {
```

```

        if(is_empty())
        {
            System.out.println("Queue Underflow");
            return ;
        }
        else
        {
            for(int i=front;i<=rear;i++)
                System.out.print(arr[i]+" ");
            System.out.println();
        }
    }

    boolean is_empty()
    {
        if(front==-1&&rear==-1)
            return true;
        else
            return false;
    }

    boolean is_full()
    {
        if(rear==arr.length-1)
            return true;
        else
            return false;
    }

    public static void main(String[] args) {
        Scanner sc=new Scanner (System.in);
        System.out.println("Enter the size of Queue");
        int s=sc.nextInt();
        QueueUsingArray ob=new QueueUsingArray(s);
        while(true)
        {
            System.out.println("****MENU****");
            System.out.println("0:Exit");
            System.out.println("1:Enqueue");
            System.out.println("2:Dequeue");
            System.out.println("3:Display");
            System.out.println("*****");
            System.out.println("Enter the choice");

```

```
int choice=sc.nextInt();
switch(choice)
{
case 0:
System.exit(0);
case 1:
System.out.println("Enter the element:");
int e=sc.nextInt();
ob.enqueue(e);
break;
case 2:
ob.dequeue();
break;
case 3:
ob.display();
break;
default:
System.out.println("Wrong choice");
}
}
}
```

```

import java.util.Scanner;

public class StackUsingArray {
    int arr[];
    int top;
    StackUsingArray(int max)
    {
        arr=new int[max];
        top=-1;
    }

    void push(int ele)
    {
        if(is_full())
            System.out.println("Stack Overflow");
        else
        {
            top=top+1;
            arr[top]=ele;
        }
    }

    int pop()
    {
        if(is_empty())
        {
            System.out.println("Stack Underflow");
            return -1;
        }
        else
        {
            int temp=arr[top];
            top=top-1;
            return temp;
        }
    }

    int peek()
    {
        if(is_empty())
        {
            System.out.println("Stack Underflow");
            return -1;
        }
        else
        {
            return arr[top];
        }
    }

    boolean is_empty()
    {
        if(top==-1)
            return true;
        else
            return false;
    }

    boolean is_full()
    {

```

```

        if(top==arr.length-1)
            return true;
        else
            return false;
    }

    public static void main(String[] args) {
        Scanner sc=new Scanner (System.in);
        System.out.println("Enter the Stack size");
        int max=sc.nextInt();
        StackUsingArray s= new StackUsingArray(max);
        while(true)
        {
            System.out.println("****MENU****");
            System.out.println("0:Exit");
            System.out.println("1:Push");
            System.out.println("2:Pop");
            System.out.println("3:Peek");

            System.out.println("*****");
            System.out.println("Enter the choice");
            int choice=sc.nextInt();
            switch(choice)
            {
                case 0:
                    System.exit(0);
                case 1:
                    System.out.println("Enter the element:");
                    int e=sc.nextInt();
                    s.push(e);
                    break;
                case 2:
                    int m=s.pop();
                    if(m!=-1)
                        System.out.println("The popped value is: "+m );
                    break;
                case 3:
                    m=s.peek();
                    if(m!=-1)
                        System.out.println("The Top value is: "+m );
                    break;
                default:
                    System.out.println("Wrong choice");
            }
        }
    }
}

```



```

import java.util.Scanner;
class Node {

    int info;
    Node link;
}
public class StackUsingLinkedList {
    static Node top=null;

    public static void push(int x)
    {
        Node p = new Node();
        p.info = x;
        p.link = top;
        top = p;
    }
    public static void pop()
    {
        if(top==null)
        {
            System.out.println("Stack Underflow ");
            return;
        }

        System.out.println("Poped info is:"+top.info);
        top=top.link;
    }

    public static void peek()
    {
        if(top==null)
        {
            System.out.println("Stack Underflow ");
            return;
        }

        System.out.println("Top info is:"+top.info);
    }
    public static void main(String[] args) {
        Scanner sc=new Scanner (System.in);

        while(true)
        {
            System.out.println("****MENU****");
            System.out.println("0:Exit");
            System.out.println("1:Push");
            System.out.println("2:Pop");
            System.out.println("3:Peek");

            System.out.println("*****");
            System.out.println("Enter the choice");
            int choice=sc.nextInt();
            switch(choice)
            {
                case 0:
                    System.exit(0);
                case 1:
                    System.out.println("Enter the element to be pushed");
                    int x=sc.nextInt();

```

```
        push(x);  
        break;  
    case 2:  
        pop();  
        break;  
    case 3:  
        peek();  
        break;  
    default:  
        System.out.println("Wrong choice");  
    }  
}  
  
}
```

```

import java.util.Scanner;
class Node
{
    Node prev;
    int info;
    Node next;
}
public class DoublyLinkedList
{
    public static Node head=null;
    public static Node tail=null;

    public static void create()
    {
        Scanner sc=new Scanner(System.in);
        Node p=new Node();
        System.out.println("Enter the value");
        p.info=sc.nextInt();
        p.next=null;
        p.prev=null;
        head=tail=p;
        System.out.println("Add more? (Y/N)");
        char ch=sc.next().charAt(0);
        while(ch=='y' || ch=='Y')
        {
            Node q=new Node();
            System.out.println("Enter the value");
            q.info=sc.nextInt();
            q.next=null;
            q.prev=tail;
            tail.next=q;
            tail=q;
            System.out.println("Add more? (Y/N)");
            ch=sc.next().charAt(0);
        }
    }

    public static void display()
    {
        Node temp=head;
        System.out.println("Foreward");
        System.out.print("null----->");
        while(temp!=null)
        {
            System.out.print(temp.info+"----->");
            temp=temp.next;
        }
        System.out.println("null");
        temp=tail;
        System.out.println("Backward");
        System.out.print("null<-----");
        while(temp!=null)
        {
            System.out.print(temp.info+"<-----");
            temp=temp.prev;
        }
        System.out.println("null");
    }
}

```

```

}
public static int count()
{
    Node temp=head;
    int c=0;
    while(temp!=null)
    {
        c++;
        temp=temp.next;
    }
    return c;
}
public static void insert_beg()
{
    Scanner sc=new Scanner(System.in);
    Node p=new Node();
    System.out.println("Enter the value");
    p.info=sc.nextInt();
    if(head==null)
    {
        p.prev=p.next=null;
        head=tail=p;
    }
    p.next=head;
    p.prev=null;
    head.prev=p;
    head=p;
}
public static void insert_end()
{
    Scanner sc=new Scanner(System.in);
    Node p=new Node();
    System.out.println("Enter the value");
    p.info=sc.nextInt();
    if(head==null)
    {
        p.prev=p.next=null;
        head=tail=p;
    }
    p.prev=tail;
    p.next=null;
    tail.next=p;
    tail=p;
}
public static void insert_pos(int pos)
{
    Scanner sc=new Scanner(System.in);
    int count=count();
    if(pos>=1&&pos<=count+1)
    {
        if(pos==1)
            insert_beg();
        else if(pos==count+1)
            insert_end();
        else
        {

```

```

        int cnt=1;
        Node temp=head;
        while(cnt<pos)
        {
            cnt=cnt+1;
            temp=temp.next;
        }
        Node p=new Node();
        System.out.println("Enter the value");
        p.info=sc.nextInt();
        p.prev=temp.prev;
        p.next=temp;
        temp.prev.next=p;
        temp.prev=p;
    }
}
else
{
    System.out.println("Invalid position");
}
}
public static void del_beg()
{
    if(head==null)
    {
        System.out.println("Underflow");
        return;
    }
    else if(head.next==null)
        head=tail=null;
    else
    {
        head=head.next;
        head.prev=null;
    }
}
public static void del_end()
{
    if(head==null)
    {
        System.out.println("Underflow");
        return;
    }
    else if(head.next==null)
        head=tail=null;
    else
    {
        tail=tail.prev;
        tail.next=null;
    }
}
public static void del_pos()
{
    Scanner sc=new Scanner(System.in);
    if(head==null)
    {

```

```

        System.out.println("Underflow");
        return;
    }
    System.out.println("Enter the position");
    int pos=sc.nextInt();
    int count=count();
    if(pos>=1&&pos<=count)
    {
        if(pos==1)
            del_beg();
        else if(pos==count)
            del_end();
        else
        {
            int cnt=1;
            Node temp=head;
            while(cnt<pos)
            {
                cnt=cnt+1;
                temp=temp.next;
            }
            temp.prev.next=temp.next;
            temp.next.prev=temp.prev;
            temp.next=temp.next=null;
            temp=null;
        }
    }
    else
        System.out.println("Invalid position");
}

public static void main(String[] args) {
    Scanner sc=new Scanner (System.in);
    while(true)
    {
        System.out.println("****MENU****");
        System.out.println("0:Exit");
        System.out.println("1:Creation");
        System.out.println("2:Display");
        System.out.println("3:Count");
        System.out.println("4:Insert");
        System.out.println("5:Delete");
        System.out.println("*****");
        System.out.println("Enter the choice");
        int choice=sc.nextInt();
        switch(choice)
        {
            case 0:
                System.exit(0);
            case 1:
                create();
                break;
            case 2:
                display();
                break;
            case 3:
                System.out.println("No. of Nodes =" +count());
        }
    }
}

```

```

        break;
    case 4:
        System.out.println("****INSERT****");
        System.out.println("1: Begning");
        System.out.println("2: End");
        System.out.println("3: Specific Position");
        System.out.println("Enter the choice");
        int ch=sc.nextInt();
        switch(ch)
        {
            case 1:
                insert_beg();
                break;
            case 2:
                insert_end();
                break;
            case 3:
                System.out.println("Enter the position");
                int pos=sc.nextInt();
                insert_pos(pos);
                break;
            default:
                System.out.println("Wrong choice");
                break;
        }
        break;
    case 5:
        System.out.println("****DELETE****");
        System.out.println("1: Begning");
        System.out.println("2: End");
        System.out.println("3: Specific Position");
        System.out.println("Enter the choice");
        ch=sc.nextInt();
        switch(ch)
        {
            case 1:
                del_beg();
                break;
            case 2:
                del_end();
                break;
            case 3:
                del_pos();
                break;
            default:
                System.out.println("Wrong choice");
                break;
        }
        break;
    default:
        System.out.println("Wrong choice");
    }
}
}
}

```