SATYAM TRIPATHI
202151141

LAB 7

1. Queue using Linked List

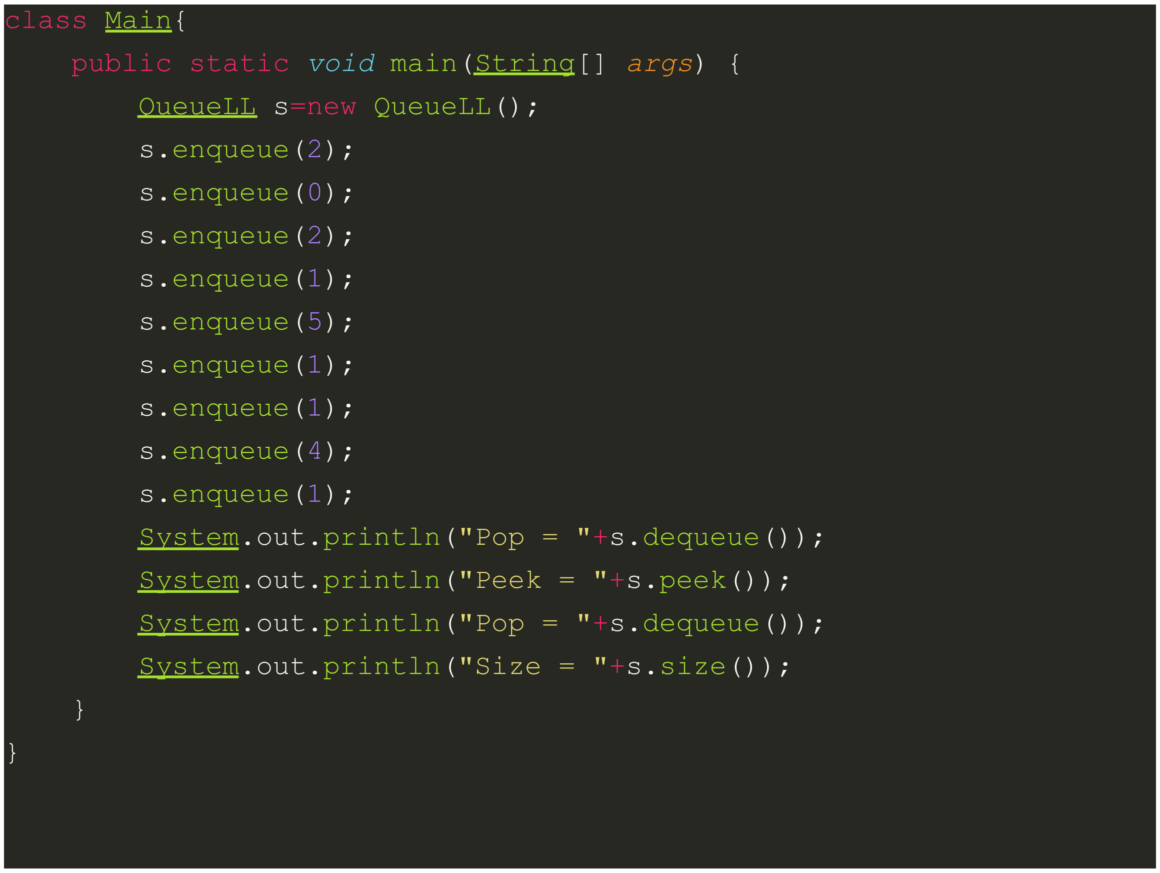
```
class Node{
    Node next;
    int data;
    public Node(int v){
        data=v;
        next=null;
    }
}
class LinkedList{
    Node first,current;
    int size;
    LinkedList(){
        first=null;
        current=null;
        size=0;

    }
    void addRear(int v){
        size++;
        if(first==null){
            first=new Node(v);
            current=first;
        }
        else{
            Node n1=new Node(v);
            current.next=n1;
            current=current.next;
        }
    }
    int deleteFront(){
        if(size==0)
        {
            System.out.println("Underflow");
            System.exit(0);
        }
        int a=first.data;
```

SATYAM TRIPATHI
202151141

```
        if(current==first){
            current=current.next;
        }
        first=first.next;
        size--;
        return a;

    }
    int size(){
        return size;
    }
    int getFront(){
        if(size==0)
        {
            System.out.println("Underflow");
            System.exit(0);
        }
        return first.data;
    }
}
class QueueLL {
    LinkedList s=new LinkedList();
    void enqueue(int v){
        s.addRear(v);
    }
    int dequeue(){
        return s.deleteFront();
    }
    boolean isEmpty(){
        if(s.size==0)
        return true;
        return false;
    }
    int size(){
        return s.size();
    }
    int peek(){
        return s.getFront();
    }
}
```

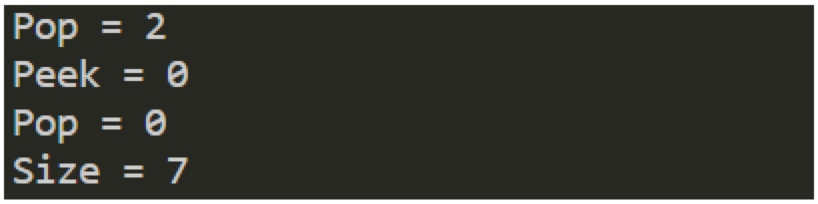
SATYAM TRIPATHI
202151141

2. Driver Class



```java
class Main{
    public static void main(String[] args) {
        QueueLL s=new QueueLL();
        s.enqueue(2);
        s.enqueue(0);
        s.enqueue(2);
        s.enqueue(1);
        s.enqueue(5);
        s.enqueue(1);
        s.enqueue(1);
        s.enqueue(4);
        s.enqueue(1);
        System.out.println("Pop = "+s.dequeue());
        System.out.println("Peek = "+s.peek());
        System.out.println("Pop = "+s.dequeue());
        System.out.println("Size = "+s.size());
    }
}
```

OUTPUT



```
Pop = 2
Peek = 0
Pop = 0
Size = 7
```

3. Implement the following

SATYAM TRIPATHI
202151141

a. Stack using Queue

```
class Node{
    Node next;
    int data;
    public Node(int v){
        data=v;
        next=null;
    }
}
class LinkedList{
    Node first,current;
    int size;
    LinkedList(){
        first=null;
        current=null;
        size=0;

    }
    void addRear(int v){
        size++;
        if(first==null){
            first=new Node(v);
            current=first;
        }
        else{
            Node n1=new Node(v);
            current.next=n1;
            current=current.next;
        }
    }
    int deleteFront(){
        if(size==0)
```

SATYAM TRIPATHI
202151141

```
        {
            System.out.println("Underflow");
            System.exit(0);
        }
        int a=first.data;
        if(current==first){
            current=current.next;
        }
        first=first.next;
        size--;
        return a;

    }
    int size(){
        return size;
    }
    int getFront(){
        if(size==0)
        {
            System.out.println("Underflow");
            System.exit(0);
        }
        return first.data;
    }
}
class QueueLL {
    LinkedList s=new LinkedList();
    void enqueue(int v){
        s.addRear(v);
    }
    int dequeue(){
        return s.deleteFront();
    }
```

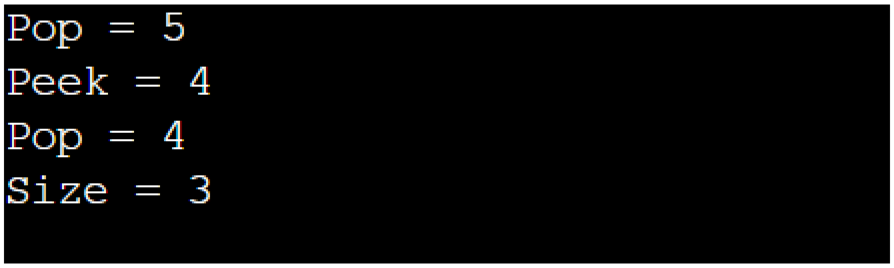
SATYAM TRIPATHI
202151141

```java
    int size(){
        return s.size();
    }
    boolean isEmpty(){
        if(s.size==0)
        return true;
        return false;
    }
    int peek(){
        return s.getFront();
    }
}
class StackUsingQueue {
    QueueLL q1=new QueueLL();
    QueueLL q2=new QueueLL();
    int size=0;
    void push(int v) {
        q1.enqueue(v);
        size++;
    }
    int pop(){
        if(isEmpty()){
            System.out.println("Underflow");

        }
        while(q1.size()!=1){
            q2.enqueue(q1.dequeue());
        }
        int v=q1.dequeue();
        while(!q2.isEmpty()){
            q1.enqueue(q2.dequeue());
        }
        size--;
```

SATYAM TRIPATHI
202151141

```
        return v;

    }
    int peek(){
        if(isEmpty()){
            System.out.println("Underflow");
            System.exit(0);
        }
        while(q1.size()!=1){
            q2.enqueue(q1.dequeue());
        }
        int v=q1.dequeue();
        q2.enqueue(v);
        while(!q2.isEmpty()){
            q1.enqueue(q2.dequeue());
        }
        return v;
    }
    boolean isEmpty(){
        return size==0;
    }
    int size(){
        return size;
    }
}
public class Main{
    public static void main(String[] args) {
        StackUsingQueue s=new StackUsingQueue();
        s.push(1);
        s.push(2);
        s.push(3);
        s.push(4);
        s.push(5);
        System.out.println("Pop = "+s.pop());
        System.out.println("Peek = "+s.peek());
        System.out.println("Pop = "+s.pop());
        System.out.println("Size = "+s.size());

    }
}
```

OUTPUT



```
Pop = 5
Peek = 4
Pop = 4
Size = 3
```

b. Queue using Stack

```
class Node{
    int data;
    Node next;
    public Node(int v){
        data=v;
        next=null;
    }
    public Node(){}
}
```

SATYAM TRIPATHI
202151141

```java
class LinkedList {
    protected Node first;
    protected int size;
    public LinkedList(){
        first=null;
        size=0;
    }
    public boolean isEmpty(){
        if(size==0)
        return true;
        return false;
    }
    public int size(){
        return size;
    }
    public void checkIndex(int index){
        if(index<0 || index>size){
            throw new IndexOutOfBoundsException("index = "+index+" for size = "+size);
        }
    }
    public int get(int index){
        checkIndex(index);
        Node current=first;
        for(int i=0;i<index;i++)
        {
            current=current.next;
        }
        return current.data;

    }

    public void add(int element,int index){
```

SATYAM TRIPATHI
202151141

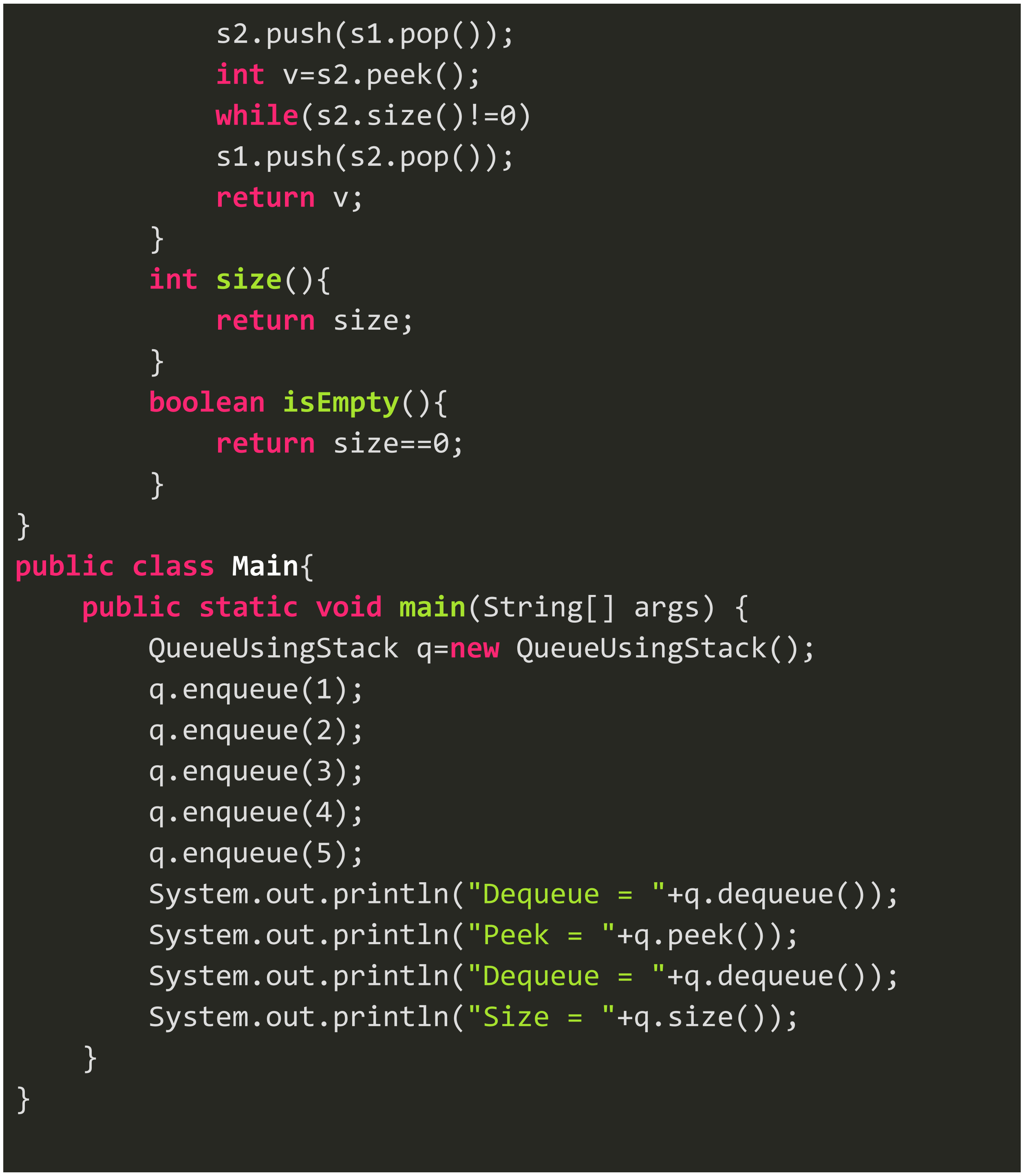
```
        checkIndex(index);

        size++;
        Node temp=first;
        if(index==0)
        {
            first=new Node(element);

            return;
        }
       for(int i=1;i<index;i++)
       {
            temp=temp.next;
       }
       Node n1=new Node(element);
       n1.next=temp.next;
       temp.next=n1;

    }

    public int remove(int index){
        checkIndex(index);
        Node temp=first;
        Node prev=null;
        size--;
        while(temp.next!=null){
            prev=temp;
            temp=temp.next;
        }
        if(temp==first)
        {
            int v=temp.data;
            first=null;
```

SATYAM TRIPATHI
202151141

```
            return v;
        }
        prev.next=null;
        int v=temp.data;
        temp=null;
        return v;
    }
    public void show(){
        Node temp=first;
        while(temp!=null){
            System.out.print(temp.data+" ");
            temp=temp.next;
        }
        System.out.println();
    }



}

class StackLL  {
    LinkedList list=new LinkedList();
    public void push(int v){
        list.add(v,list.size());

    }
    public int pop(){
        if(list.isEmpty()){
            System.out.println("Underflow");
            System.exit(0);
        }

        return list.remove(list.size()-1);
```

```java
    }
    public int peek(){
        return list.get(list.size()-1);
    }
    public int size(){
        return list.size();
    }
    public void show(){
        list.show();
    }

}
 class QueueUsingStack {

        StackLL s1=new StackLL();
        StackLL s2=new StackLL();
        int size=0;
        void enqueue(int v){
            size++;
            s1.push(v);
        }
        int dequeue(){
            while(s1.size()!=0)
            s2.push(s1.pop());
            int v=s2.pop();
            while(s2.size()!=0)
            s1.push(s2.pop());
            size--;
            return v;
        }
        int peek(){
            while(s1.size()!=0)
```

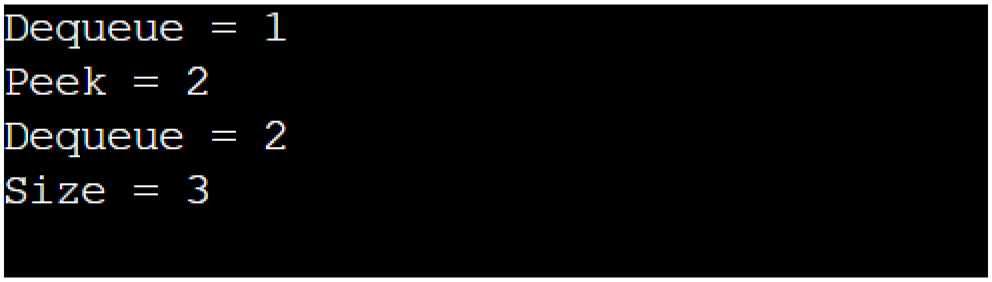
SATYAM TRIPATHI
202151141



```java
            s2.push(s1.pop());
            int v=s2.peek();
            while(s2.size()!=0)
            s1.push(s2.pop());
            return v;
        }
        int size(){
            return size;
        }
        boolean isEmpty(){
            return size==0;
        }
}
public class Main{
    public static void main(String[] args) {
        QueueUsingStack q=new QueueUsingStack();
        q.enqueue(1);
        q.enqueue(2);
        q.enqueue(3);
        q.enqueue(4);
        q.enqueue(5);
        System.out.println("Dequeue = "+q.dequeue());
        System.out.println("Peek = "+q.peek());
        System.out.println("Dequeue = "+q.dequeue());
        System.out.println("Size = "+q.size());
    }
}
```

SATYAM TRIPATHI
202151141

OUTPUT



```
Dequeue = 1
Peek = 2
Dequeue = 2
Size = 3
```