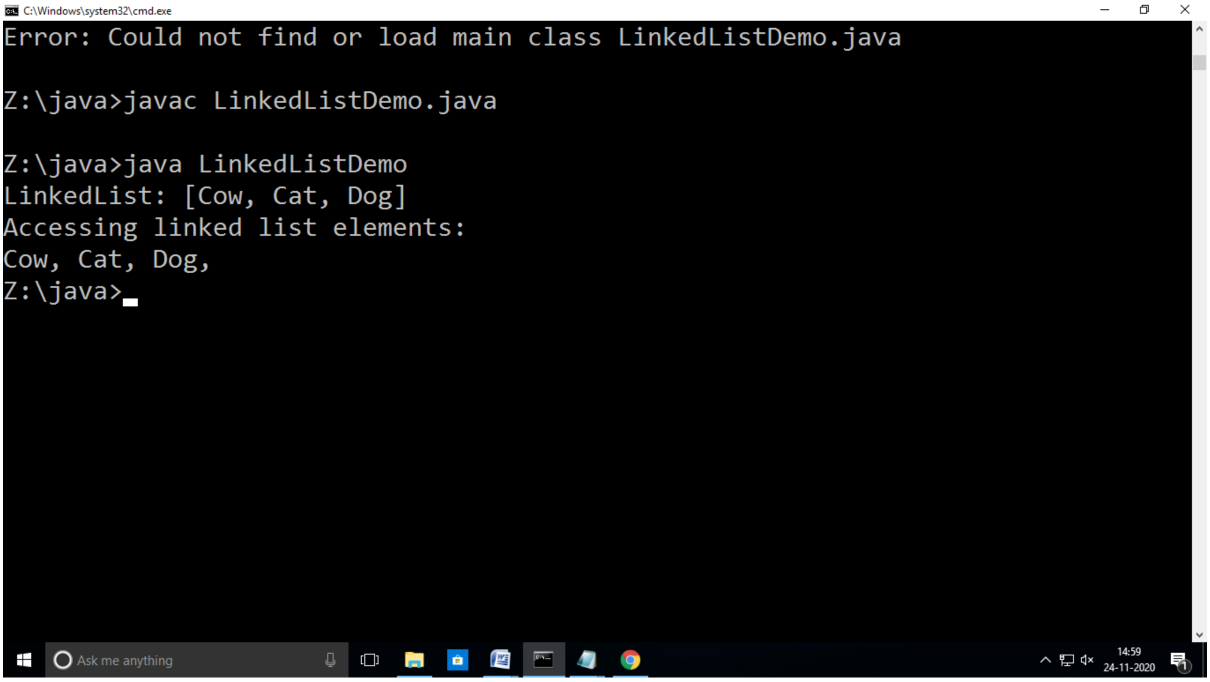
1.Implement the following Data structures in Java

a)Linked Lists b) Stacks c) Queues d) Set e) Map

a.Linked list

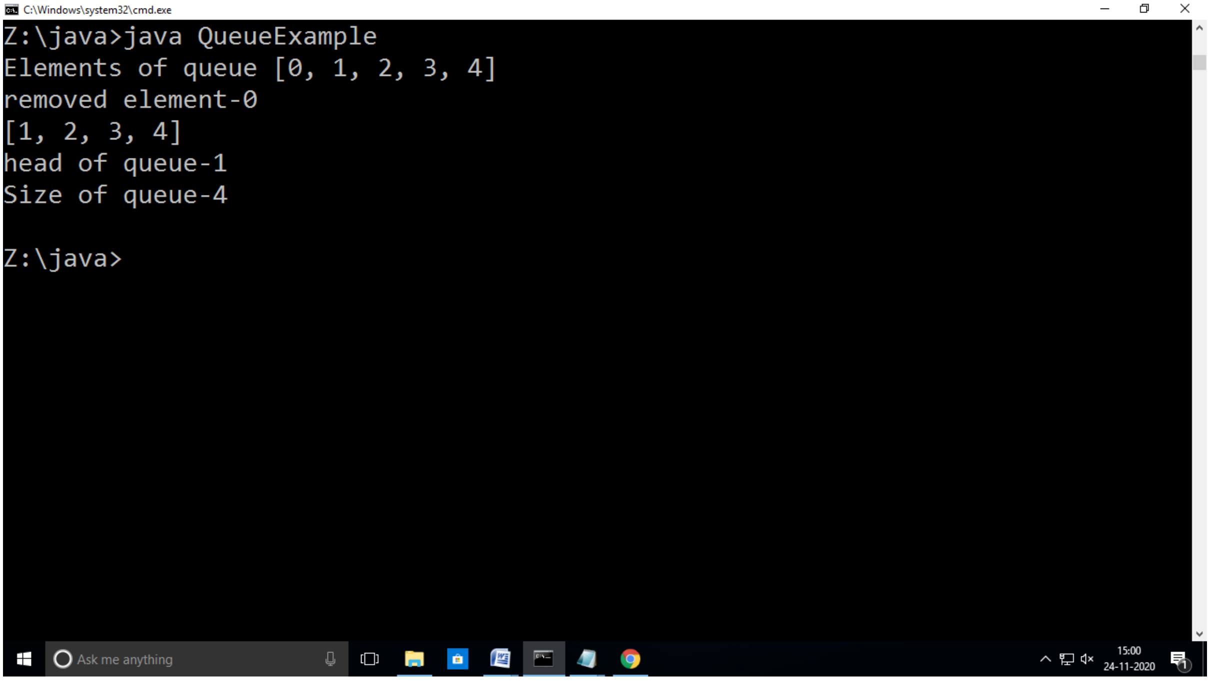
```
import java.util.*;

public class LinkedListDemo {

public static void main(String args[]) {

LinkedList<String> animals = new LinkedList<>();

        animals.add("Cow");

        animals.add("Cat");

        animals.add("Dog");

        System.out.println("LinkedList: " + animals);

// Using forEach loop

        System.out.println("Accessing linked list elements:");

        for(String animal: animals) {

            System.out.print(animal);

            System.out.print(", ");

}

}

}
```



```
Error: Could not find or load main class LinkedListDemo.java

Z:\java>javac LinkedListDemo.java

Z:\java>java LinkedListDemo
LinkedList: [Cow, Cat, Dog]
Accessing linked list elements:
Cow, Cat, Dog,
Z:\java>
```

2.Queue

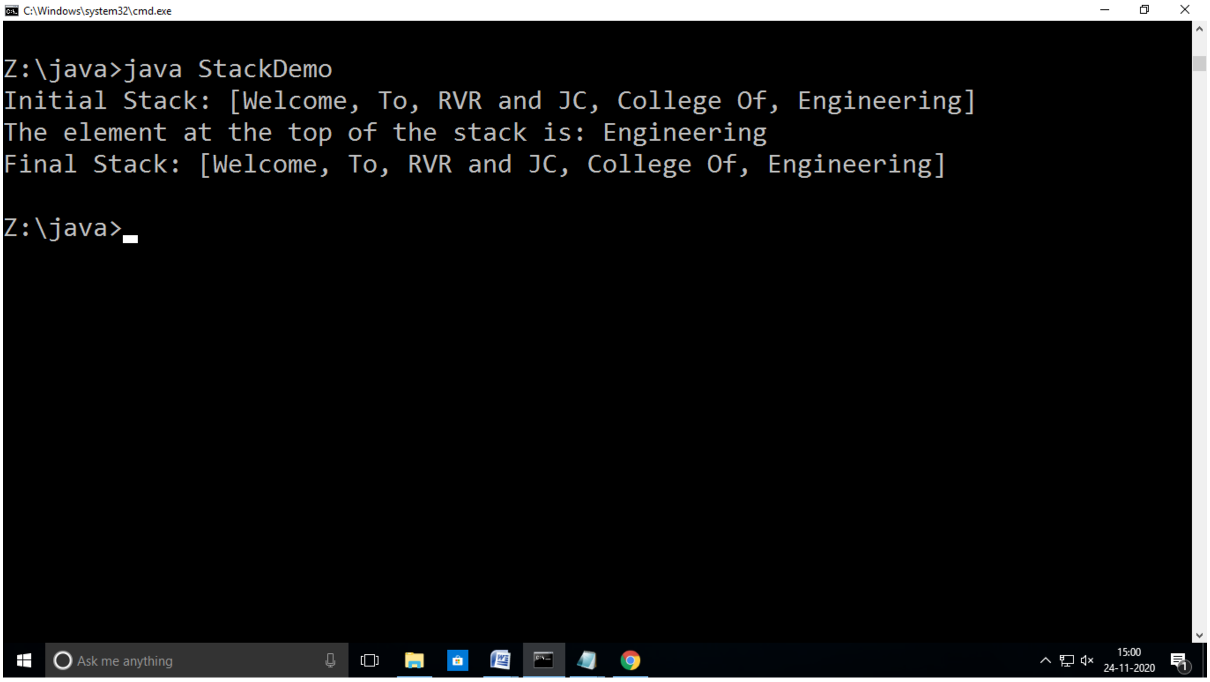
```
import java.util.LinkedList;
import java.util.Queue;

public class QueueExample {

    public static void main(String[] args)
    {
        Queue<Integer> q
            = new LinkedList<>();

        // Adds elements {0, 1, 2, 3, 4} to
        // the queue
        for (int i = 0; i < 5; i++)
            q.add(i);

        // Display contents of the queue.
        System.out.println("Elements of queue "
                           + q);

        // To remove the head of queue.
        int removedele = q.remove();
        System.out.println("removed element-"
                           + removedele);

        System.out.println(q);

        // To view the head of queue
        int head = q.peek();
        System.out.println("head of queue-"
                           + head);

        // Rest all methods of collection
        // interface like size and contains
        // can be used with this
        // implementation.
        int size = q.size();
        System.out.println("Size of queue-"
                           + size);
    }
}
```



```
Z:\java>java QueueExample
Elements of queue [0, 1, 2, 3, 4]
removed element-0
[1, 2, 3, 4]
head of queue-1
Size of queue-4

Z:\java>
```

3.stack

```
import java.io.*;
import java.util.*;

class StackDemo {

      // Main Method
    public static void main(String[] args)
    {

        Stack<String> stack = new Stack<String>();

        // Use push() to add elements into the Stack
        stack.push("Welcome");
        stack.push("To");
        stack.push("RVR and JC");
        stack.push("College Of");
        stack.push("Engineering");

        // Displaying the Stack
        System.out.println("Initial Stack: " + stack);

        // Fetching the element at the head of the Stack
        System.out.println("The element at the top of the"
                           + " stack is: " + stack.peek());
        // Displaying the Stack after the Operation
        System.out.println("Final Stack: " + stack);
    }
}
```

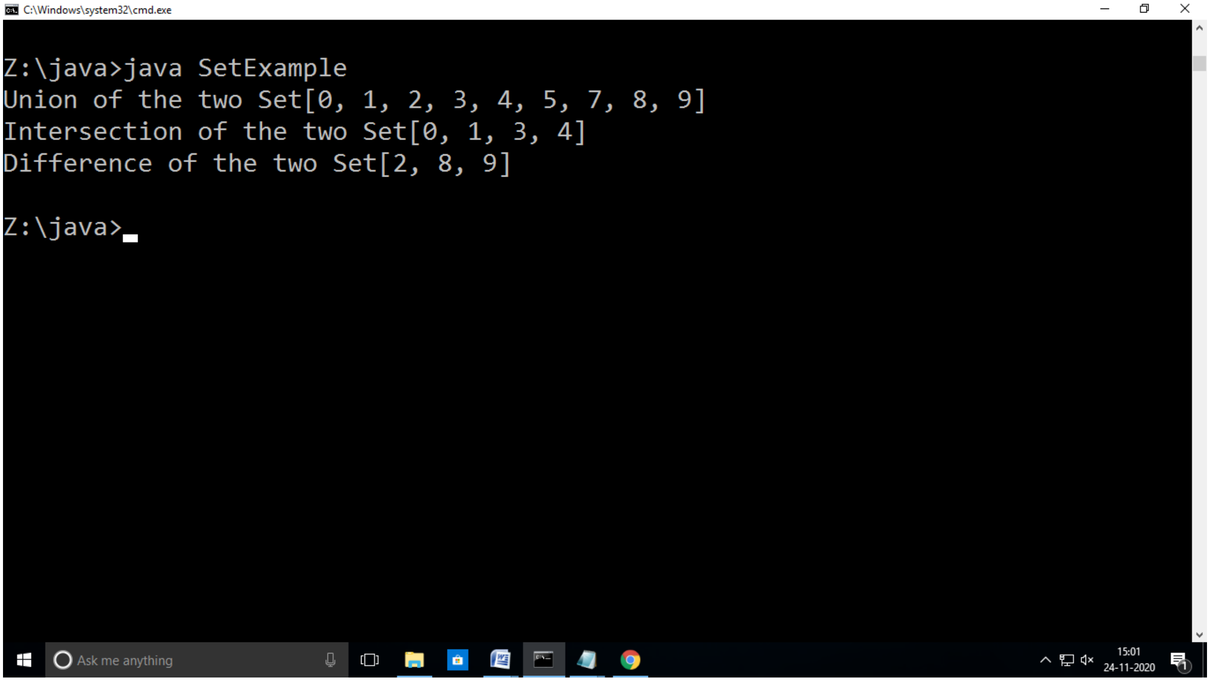


C:\Windows\system32\cmd.exe

```
Z:\java>java StackDemo
Initial Stack: [Welcome, To, RVR and JC, College Of, Engineering]
The element at the top of the stack is: Engineering
Final Stack: [Welcome, To, RVR and JC, College Of, Engineering]

Z:\java>_
```

4.set

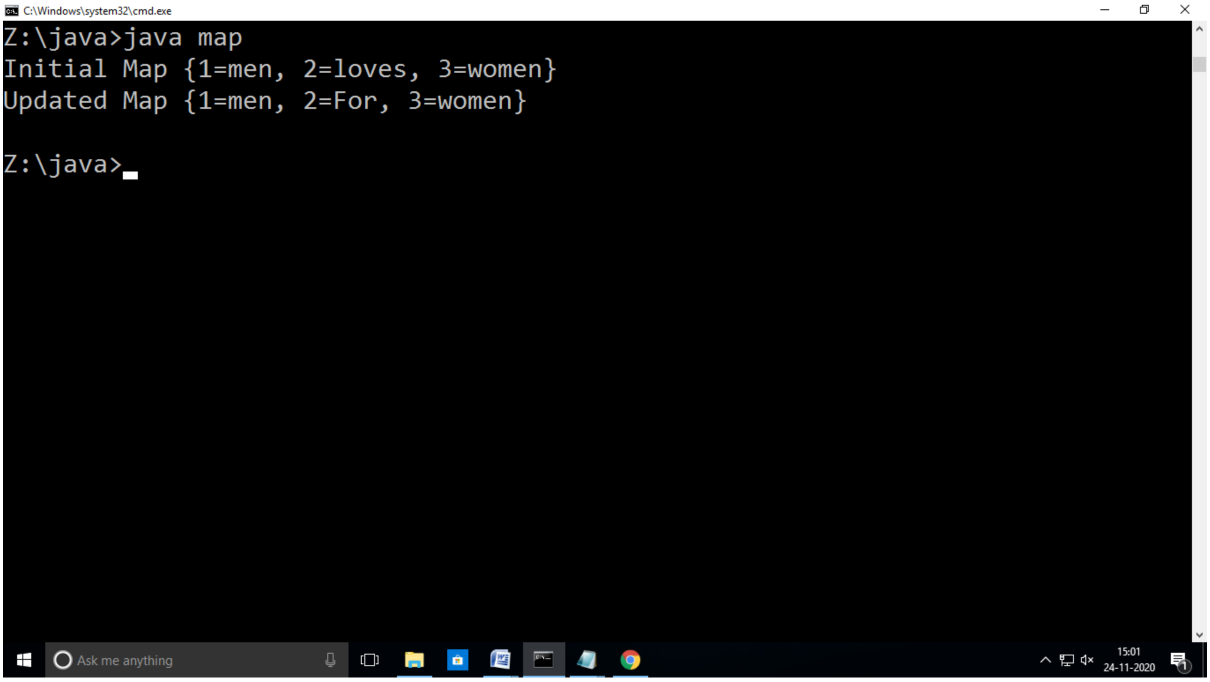
```
import java.util.*;
public class SetExample
{
    public static void main(String args[])
    {
        Set<Integer> a = new HashSet<Integer>();
        a.addAll(Arrays.asList(new Integer[] {1, 3, 2, 4, 8, 9, 0}));
        Set<Integer> b = new HashSet<Integer>();
        b.addAll(Arrays.asList(new Integer[] {1, 3, 7, 5, 4, 0, 7, 5}));

        // To find union
        Set<Integer> union = new HashSet<Integer>(a);
        union.addAll(b);
        System.out.print("Union of the two Set");
        System.out.println(union);

        // To find intersection
        Set<Integer> intersection = new HashSet<Integer>(a);
        intersection.retainAll(b);
        System.out.print("Intersection of the two Set");
        System.out.println(intersection);

        // To find the symmetric difference
        Set<Integer> difference = new HashSet<Integer>(a);
        difference.removeAll(b);
        System.out.print("Difference of the two Set");
        System.out.println(difference);
    }
}
```



```
Z:\java>java SetExample
Union of the two Set[0, 1, 2, 3, 4, 5, 7, 8, 9]
Intersection of the two Set[0, 1, 3, 4]
Difference of the two Set[2, 8, 9]

Z:\java>
```

5.MAP

```
import java.util.*;
class map {
    public static void main(String args[])
    {

        // Initialization of a Map
        // using Generics
        Map<Integer, String> hm1
            = new HashMap<Integer, String>();

        // Inserting the Elements
        hm1.put(new Integer(1), "men");
        hm1.put(new Integer(2), "loves");
        hm1.put(new Integer(3), "women");

        System.out.println("Initial Map " + hm1);

        hm1.put(new Integer(2), "For");

        System.out.println("Updated Map " + hm1);
    }
}
```



```
Z:\java>java map
Initial Map {1=men, 2=loves, 3=women}
Updated Map {1=men, 2=For, 3=women}

Z:\java>_
```