Practical Assignment No. 5

Implement Mutual Exclusion using Token Ring.

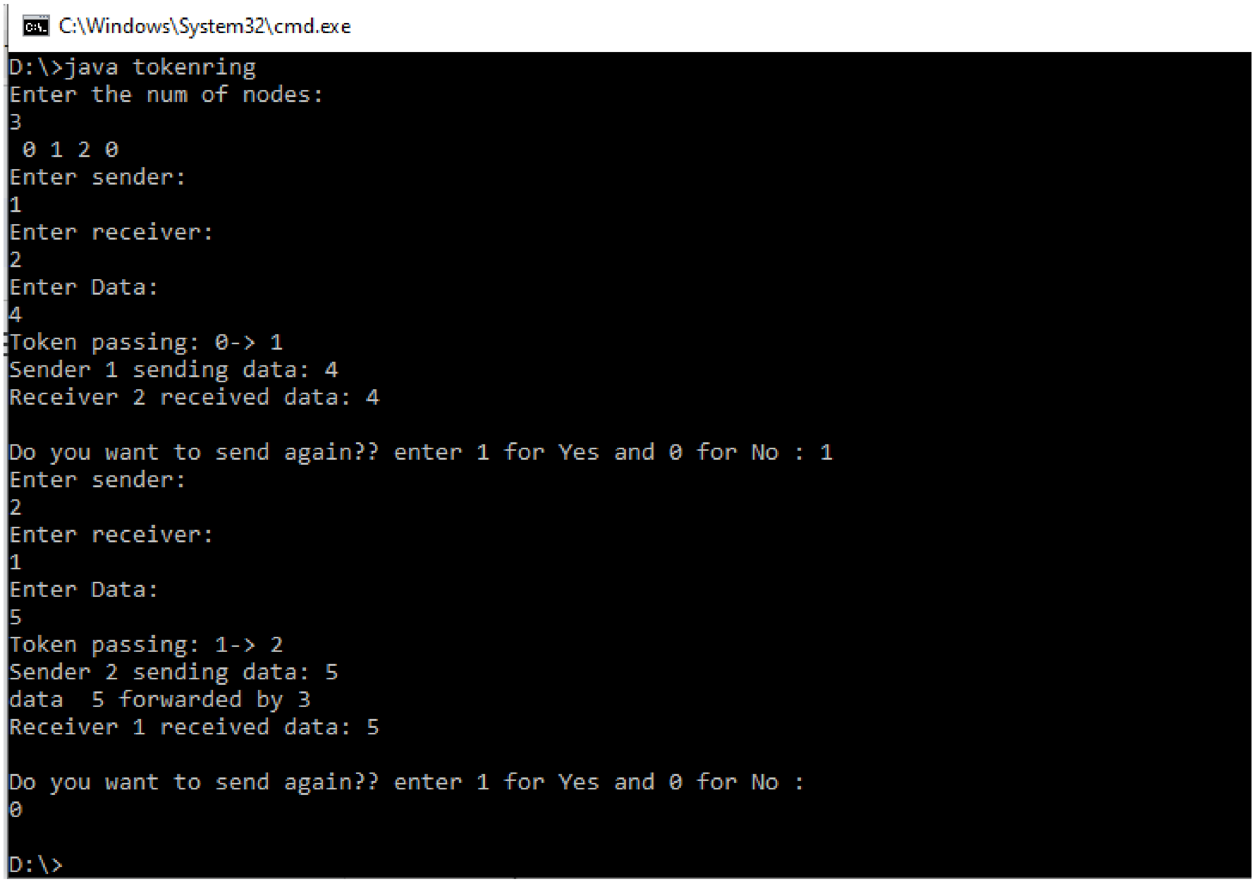
```
import java.io.*;
import java.util.*;

class tokenring {

    public static void main(String args[]) throws Throwable {
        Scanner scan = new Scanner(System.in);
        System.out.println("Enter the num of nodes:");
        int n = scan.nextInt();
        int m = n - 1;
        // Decides the number of nodes forming the ring
        int token = 0;
        int ch = 0, flag = 0;
        for (int i = 0; i < n; i++) {
            System.out.print(" " + i);
        }
        System.out.println(" " + 0);
        do{
            System.out.println("Enter sender:");
            int s = scan.nextInt();
            System.out.println("Enter receiver:");
            int r = scan.nextInt();
            System.out.println("Enter Data:");
            int a;
            a = scan.nextInt();
            System.out.print("Token passing:");
            for (int i = token, j = token; (i % n) != s; i++, j = (j + 1) % n) {
                System.out.print(" " + j + "->");
            }
            System.out.println(" " + s);
            System.out.println("Sender " + s + " sending data: " + a);
            for (int i = s + 1; i != r; i = (i + 1) % n) {
                System.out.println("data  " + a + " forwarded by " + i);
            }
            System.out.println("Receiver " + r + " received data: " + a +"\n");
            token = s;
            do{
                try {
                    if( flag == 1)
```

```
                                    System.out.print("Invalid Input!!...");
                    System.out.print("Do you want to send again?? enter 1 for Yes and 0 for No : ");
                    ch = scan.nextInt();
                    if( ch != 1 && ch != 0 )
                                    flag = 1;
                    else
                                    flag = 0;
                } catch (InputMismatchException e){
                    System.out.println("Invalid Input");
                }
            }while( ch != 1 && ch != 0 );
        }while( ch == 1 );
    }
}
```

Output:



```
D:\>java tokenring
Enter the num of nodes:
3
 0 1 2 0
Enter sender:
1
Enter receiver:
2
Enter Data:
4
Token passing: 0-> 1
Sender 1 sending data: 4
Receiver 2 received data: 4

Do you want to send again?? enter 1 for Yes and 0 for No : 1
Enter sender:
2
Enter receiver:
1
Enter Data:
5
Token passing: 1-> 2
Sender 2 sending data: 5
data  5 forwarded by 3
Receiver 1 received data: 5

Do you want to send again?? enter 1 for Yes and 0 for No :
0

D:\>
```