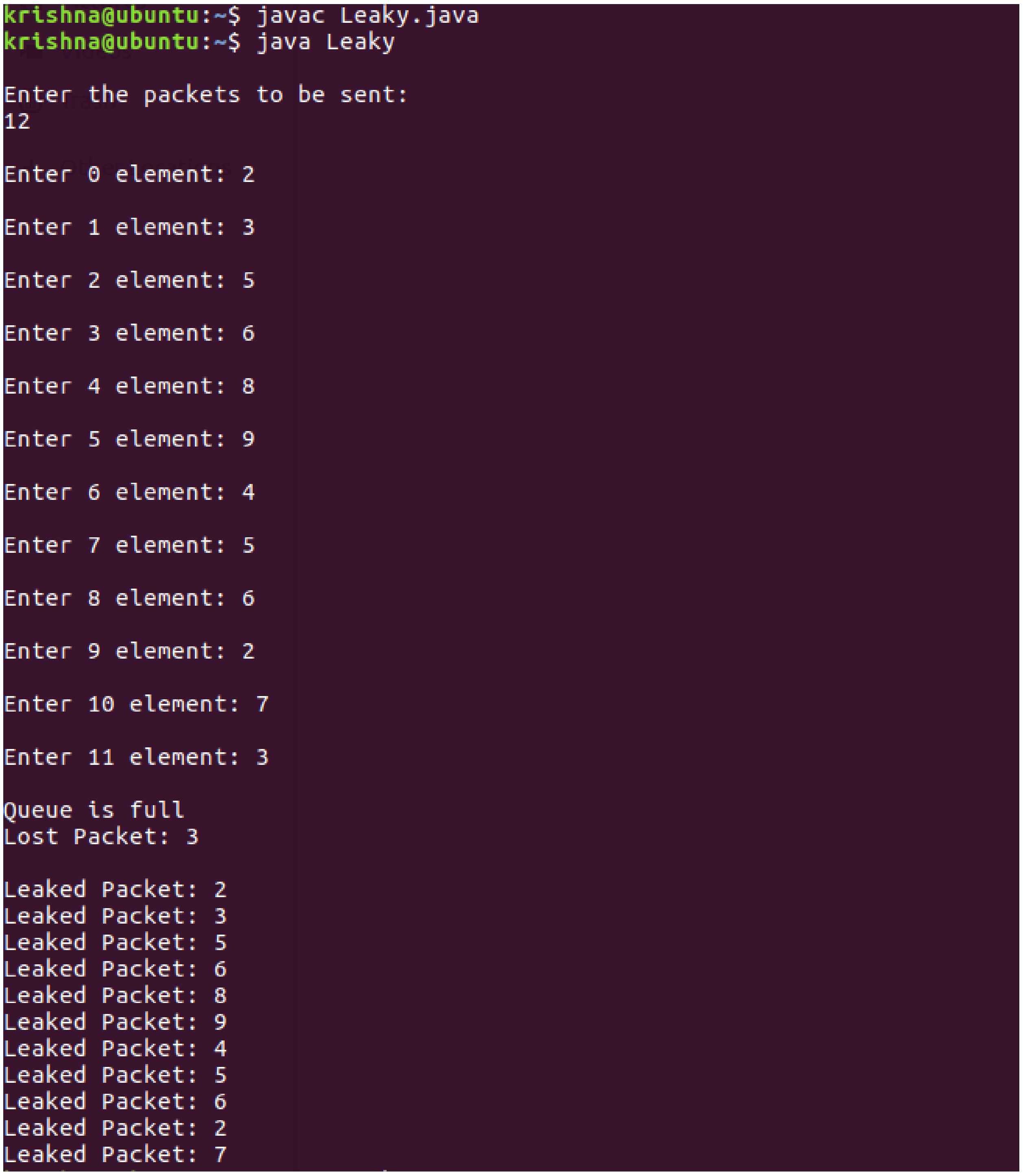
Congestion control using leaky bucket algorithm.

```java
import java.io.*;
import java.util.*;
public class Leaky
{
   public static void main(String args[]) throws Exception
  {
        Queue q=new Queue();
        Scanner src=new Scanner(System.in);
        System.out.println("\nEnter the packets to be sent:");
        int size=src.nextInt();

        q.insert(size);
        q.delete();
  }
}

class Queue
{
        int q[],f=0,r=0,size;
        void insert(int n)
        {
                Scanner in = new Scanner(System.in);
                q=new int[10];
                for(int i=0;i<n;i++)
                {
                        System.out.print("\nEnter " + i + " element: ");
                        int ele=in.nextInt();
                        if(r+1>10)
                        {
                          System.out.println("\nQueue is full \nLost Packet: "+ele);
                          break;
                        }
                        else
                        {
                                 r++;
                                q[i]=ele;
                        }
        }
    }

void delete()
{
        Scanner in = new Scanner(System.in);
        Thread t=new Thread();
        if(r==0)
        System.out.print("\nQueue empty ");
        else
        {
                for(int i=f;i<r;i++)
                {
                        try
                        {
                                t.sleep(1000);
                        }

                        catch(Exception e){}
                        System.out.print("\nLeaked Packet: "+q[i]);
                        f++;
                }
        }
                System.out.println();
}

}
```

Congestion control using leaky bucket algorithm.

Output:



```
krishna@ubuntu:~$ javac Leaky.java
krishna@ubuntu:~$ java Leaky

Enter the packets to be sent:
12

Enter 0 element: 2

Enter 1 element: 3

Enter 2 element: 5

Enter 3 element: 6

Enter 4 element: 8

Enter 5 element: 9

Enter 6 element: 4

Enter 7 element: 5

Enter 8 element: 6

Enter 9 element: 2

Enter 10 element: 7

Enter 11 element: 3

Queue is full
Lost Packet: 3

Leaked Packet: 2
Leaked Packet: 3
Leaked Packet: 5
Leaked Packet: 6
Leaked Packet: 8
Leaked Packet: 9
Leaked Packet: 4
Leaked Packet: 5
Leaked Packet: 6
Leaked Packet: 2
Leaked Packet: 7
```

Congestion control using leaky bucket algorithm.

```
import java.util.*;
public class leaky_b
{
    public static void main(String[] args)
    {
        Scanner my = new Scanner(System.in);
        int no_groups,bucket_size;
        System.out.print("\n Enter the bucket size : \t");
        bucket_size = my.nextInt();
        System.out.print("\n Enter the no of groups : \t");
        no_groups = my.nextInt();
        int no_packets[] = new int[no_groups];
        int in_bw[] = new int[no_groups];
        int out_bw,reqd_bw=0,tot_packets=0;

        for(int i=0;i<no_groups;i++)
       {
           System.out.print("\n Enter the no of packets for group " + (i+1) + "\t");
            no_packets[i] = my.nextInt();
            System.out.print("\n Enter the input bandwidth for the group " + (i+1) + "\t");
            in_bw[i] = my.nextInt();
            if((tot_packets+no_packets[i])<=bucket_size)
            {
                tot_packets += no_packets[i];
            }

                else
```

```
      {
          do
          {
               System.out.println(" Bucket Overflow ");
               System.out.println(" Enter value less than " + (bucket_size-tot_packets));
               no_packets[i] = my.nextInt();
          }
              while((tot_packets+no_packets[i])>bucket_size);
              tot_packets += no_packets[i];
     }
              reqd_bw += (no_packets[i]*in_bw[i]);
 }
             System.out.println("\nThe total required bandwidth is " + reqd_bw);
             System.out.println("Enter the output bandwidth ");
             out_bw = my.nextInt();
             int temp=reqd_bw;
             int rem_pkts = tot_packets;
            if((out_bw<=temp)&&(rem_pkts>0))
           {
               System.out.println("Data Sent \n");
               --rem_pkts;
               System.out.println("Remaining Bandwidth " + (temp -= out_bw));
               if (temp==0)
               {
                   System.out.println("All packets are sent");
               }
          }
```

```
        if(rem_pkts>0)
        System.out.println(" packets discarded due to insufficient bandwidth");
    }
}
```