```
1# Implement three nodes point – to – point network with duplex links between them.
# Set the queue size, vary the bandwidth and find the number of packets dropped.

set ns [new Simulator]
set na [open Lab1.nam w]
$ns namtrace-all $na
set nt [open Lab1.tr w]
$ns trace-all $nt
set n0 [$ns node]
set n1 [$ns node]
set n2 [$ns node]
$ns duplex-link $n0 $n1 10Mb 10ms DropTail
$ns queue-limit $n0 $n1 1
$ns duplex-link $n1 $n2 1Mb 10ms DropTail
$ns queue-limit $n1 $n2 1
set TCP [new Agent/TCP]
$ns attach-agent $n0 $TCP
set CBR [new Application/Traffic/CBR]
$CBR attach-agent $TCP
set SINK [new Agent/TCPSink]
$ns attach-agent $n2 $SINK
$ns connect $TCP $SINK
proc End {} {
global ns na nt
$ns flush-trace
close $na
close $nt
exec nam Lab1.nam &
exit 0
}
$ns at 0.0 "$CBR start"
$ns at 50.0 "End"
$ns run
```

```
//AWK CODE//


BEGIN{Count=0;}

{

if($1=="d")

Count++;

}

END{

printf("\n\n\tNumber of Packets Droppped is %d\n\n\n",Count);

}
```

```
#PART - A : Program - 2
#Implement transmission of ping messages/trace route over a network topology consisting
of 6 nodes and find the number of packets dropped due to congestion.
set ns [new Simulator]
$ns color 1 Red
$ns color 2 Green
set na [open Lab2.nam w]
$ns namtrace-all $na
set nt [open Lab2.tr w]
$ns trace-all $nt
set n0 [$ns node]
set n1 [$ns node]
set n2 [$ns node]
set n3 [$ns node]
set n4 [$ns node]
set n5 [$ns node]
$ns duplex-link $n0 $n2 1000Mb 1ms DropTail
$ns duplex-link $n1 $n2 10Mb 1ms DropTail
$ns duplex-link $n2 $n3 1Mb 1ms DropTail
$ns duplex-link $n3 $n4 1Mb 1ms DropTail
$ns duplex-link $n3 $n5 2Mb 1ms DropTail
$ns queue-limit $n2 $n3 3
$ns queue-limit $n3 $n2 3
set Ping1 [new Agent/Ping]
$ns attach-agent $n0 $Ping1
set Ping2 [new Agent/Ping]
$ns attach-agent $n1 $Ping2
set Ping3 [new Agent/Ping]
$ns attach-agent $n4 $Ping3
set Ping4 [new Agent/Ping]
$ns attach-agent $n5 $Ping4
Agent/Ping instproc recv {from rtt} {
    $self instvar node_
    puts "Node[$node_ id] --> Node$from : RTT = $rtt ms"
}
$ns connect $Ping1 $Ping4
$ns connect $Ping2 $Ping3
$Ping1 set class_ 1
$Ping2 set class_ 2
proc End {} {
    global ns na nt
    $ns flush-trace
    close $na
    close $nt
    exec nam Lab2.nam &
    exit 0
}
for {set t 0} {$t < 5} {set t [expr $t+0.001]} {
    $ns at $t "$Ping1 send"
    $ns at $t "$Ping2 send"
}
$ns at 5.0 "End"
$ns run
```

//AWK Code//

```
BEGIN{Count=0;}

{

if($1=="d")

Count++;

}

END{printf("\n\n\tNumber of Packets Dropped is %d\n\n\n",Count);}
```

Program 3.#Implement an Ethernet LAN using n nodes and set multiple traffic nodes and plot congestion window for different source / destination.

```
set ns [new Simulator]
$ns color 1 Red
$ns color 2 Blue
set na [open Lab3.nam w]
$ns namtrace-all $na
set nt [open Lab3.tr w]
$ns trace-all $nt
set ng1 [open tcp1.xg w]
set ng2 [open tcp2.xg w]
set n0 [$ns node]
set n1 [$ns node]
set n2 [$ns node]
set n3 [$ns node]
set n4 [$ns node]
set n5 [$ns node]
$ns make-lan "$n0 $n1 $n2" 1Mb 10ms LL Queue/DropTail Mac/802_3
$ns make-lan "$n3 $n4 $n5" 2Mb 10ms LL Queue/DropTail Mac/802_3
$ns duplex-link $n0 $n3 1Mb 10ms DropTail
set tcp1 [new Agent/TCP]
set tcp2 [new Agent/TCP]
set cbr1 [new Application/Traffic/CBR]
set cbr2 [new Application/Traffic/CBR]
$ns attach-agent $n4 $tcp1
$cbr1 attach-agent $tcp1
$ns attach-agent $n1 $tcp2
$cbr2 attach-agent $tcp2
set sink1 [new Agent/TCPSink]
set sink2 [new Agent/TCPSink]
$ns attach-agent $n2 $sink1
$ns attach-agent $n5 $sink2
$ns connect $tcp1 $sink1
$ns connect $tcp2 $sink2
$tcp1 set class_ 1
$tcp2 set class_ 2
proc End {} {
global ns na nt
$ns flush-trace
close $na
close $nt
exec nam Lab3.nam &
exec xgraph tcp1.xg tcp2.xg &
exit 0
}
proc Draw {Agent File} {
global ns
set Cong [$Agent set cwnd_]
set Time [$ns now]
puts $File "$Time $Cong"
$ns at [expr $Time+0.01] "Draw $Agent $File"
}
$ns at 0.0 "$cbr1 start"
$ns at 0.7 "$cbr2 start"
$ns at 0.0 "Draw $tcp1 $ng1"
$ns at 0.0 "Draw $tcp2 $ng2"
$ns at 10.0 "End"
$ns run
```

Program 4: #Implement simple ESS and with transmitting nodes in wire-less LAN by simulation and determine the performance with respect to transmission of packets.

```
set ns [new Simulator]
set na [open Lab4.nam w]
$ns namtrace-all-wireless $na 500 500
set nt [open Lab4.tr w]
$ns trace-all $nt
set topo [new Topography]
$topo load_flatgrid 500 500
$ns node-config -adhocRouting DSDV
$ns node-config -llType LL
$ns node-config -macType Mac/802_11
$ns node-config -ifqType Queue/DropTail
$ns node-config -ifqLen 50
$ns node-config -phyType Phy/WirelessPhy
$ns node-config -channelType Channel/WirelessChannel
$ns node-config -propType Propagation/TwoRayGround
$ns node-config -antType Antenna/OmniAntenna
$ns node-config -topoInstance $topo
$ns node-config -agentTrace ON
$ns node-config -routerTrace ON
create-god 4
set n0 [$ns node]
set n1 [$ns node]
set n2 [$ns node]
set n3 [$ns node]
$n0 set X_ 250.0
$n0 set Y_ 250.0
$n0 set Z_ 0.0
$n1 set X_ 200.0
$n1 set Y_ 250.0
$n1 set Z_ 0.0
$n2 set X_ 250.0
$n2 set Y_ 250.0
$n2 set Z_ 0.0
$n3 set X_ 250.0
$n3 set Y_ 250.0
$n3 set Z_ 0.0
$ns at 0.0 "$n0 setdest 400.0 300.0 50.0"
$ns at 0.0 "$n1 setdest 50.0 100.0 20.0"
$ns at 0.0 "$n2 setdest 75.0 180.0 5.0"
$ns at 0.0 "$n3 setdest 100.0 100.0 25.0"
set tcp1 [new Agent/TCP]
$ns attach-agent $n0 $tcp1
set tcp2 [new Agent/TCP]
$ns attach-agent $n2 $tcp2
set sink1 [new Agent/TCPSink]
$ns attach-agent $n1 $sink1
set sink2 [new Agent/TCPSink]
$ns attach-agent $n3 $sink2
set cbr1 [new Application/Traffic/CBR]
$cbr1 attach-agent $tcp1
set cbr2 [new Application/Traffic/CBR]
$cbr2 attach-agent $tcp2
$ns connect $tcp1 $sink1
$ns connect $tcp2 $sink2
proc End {} {
global ns nt na
$ns flush-trace
close $na
close $nt
exec nam Lab4.nam &
}
$ns at 0.0 "$cbr1 start"
$ns at 0.0 "$cbr2 start"
$ns at 10.0 "End"
$ns run
```

AWK file:

```
BEGIN{Num_of_pkts=0;}
{
if ($1 == "r" && $3 == "_1_" && $4 ==
"AGT" && $7 == "tcp")
{
Num_of_pkts = Num_of_pkts + $8;
}
}
END{
Throughput = Num_of_pkts * 8 / $2
/1000000;
printf("\n\n\tThroughput =
%fbpms\n\n\n",Throughput);}
```

```
# Program - 5
# Implement and study the performance of GSM on NS2/NS3 (Using MAC layer) or equivalent
environment.

set bwDL(gsm) 9600
set bwUL(gsm) 9600
set propDL(gsm) .500
set propUL(gsm) .500
set buf(gsm) 10
set ns [new Simulator]
set nt [open Lab5.tr w]
$ns trace-all $nt
set nodes(c1) [$ns node]
set nodes(ms) [$ns node]
set nodes(bs1) [$ns node]
set nodes(bs2) [$ns node]
set nodes(c2) [$ns node]
proc cell_topo {} {
global ns nodes
$ns duplex-link $nodes(c1) $nodes(bs1) 3Mbps 10ms DropTail
$ns duplex-link $nodes(bs1) $nodes(ms) 1 1 RED
$ns duplex-link $nodes(ms) $nodes(bs2) 1 1 RED
$ns duplex-link $nodes(bs2) $nodes(c2) 3Mbps 50ms DropTail
}
switch gsm {
gsm -
gprs -
umts {cell_topo}
}
$ns bandwidth $nodes(bs1) $nodes(ms) $bwDL(gsm) simplex
$ns bandwidth $nodes(ms) $nodes(bs1) $bwUL(gsm) simplex
$ns bandwidth $nodes(bs2) $nodes(ms) $bwDL(gsm) simplex
$ns bandwidth $nodes(ms) $nodes(bs2) $bwUL(gsm) simplex
$ns delay $nodes(bs1) $nodes(ms) $propDL(gsm) simplex
$ns delay $nodes(ms) $nodes(bs1) $propDL(gsm) simplex
$ns delay $nodes(bs2) $nodes(ms) $propDL(gsm) simplex
$ns delay $nodes(ms) $nodes(bs2) $propDL(gsm) simplex
$ns queue-limit $nodes(bs1) $nodes(ms) $buf(gsm)
$ns queue-limit $nodes(ms) $nodes(bs1) $buf(gsm)
$ns queue-limit $nodes(bs2) $nodes(ms) $buf(gsm)
$ns queue-limit $nodes(ms) $nodes(bs2) $buf(gsm)
$ns insert-delayer $nodes(ms) $nodes(bs1) [new Delayer]
$ns insert-delayer $nodes(bs1) $nodes(ms) [new Delayer]
$ns insert-delayer $nodes(ms) $nodes(bs2) [new Delayer]
$ns insert-delayer $nodes(bs2) $nodes(ms) [new Delayer]
set tcp [new Agent/TCP]
$ns attach-agent $nodes(c1) $tcp
set sink [new Agent/TCPSink]
$ns attach-agent $nodes(c2) $sink
set ftp [new Application/FTP]
$ftp attach-agent $tcp
$ns connect $tcp $sink
proc End {} {
global ns nt
$ns flush-trace
close $nt
exec awk -f Lab5.awk Lab5.tr &
exec xgraph -P -bar -x TIME -y DATA
gsm.xg &
exit 0
}
$ns at 0.0 "$ftp start"
$ns at 10.0 "End"
$ns run
```

```
AWK file:
BEGIN {Total_no_of_pkts=0;}
{
if($1 == "r")
{
Total_no_of_pkts = Total_no_of_pkts + $6;
printf("%f %d\n",$2,Total_no_of_pkts) >>
"gsm.xg"
}
}
END{}
```

```
# Program - 6
# Implement and study the performance of CDMA on NS2/NS3 (Using stack called Call net) or
equivalent environment.

set bwDL(cdma) 384000
set bwUL(cdma) 64000
set propDL(cdma) .150
set propUL(cdma) .150
set buf(cdma) 20
set ns [new Simulator]
set nt [open Lab6.tr w]
$ns trace-all $nt
set nodes(c1) [$ns node]
set nodes(ms) [$ns node]
set nodes(bs1) [$ns node]
set nodes(bs2) [$ns node]
set nodes(c2) [$ns node]
proc cell_topo {} {
global ns nodes
$ns duplex-link $nodes(c1) $nodes(bs1) 3Mbps 10ms DropTail
$ns duplex-link $nodes(bs1) $nodes(ms) 1 1 RED
$ns duplex-link $nodes(ms) $nodes(bs2) 1 1 RED
$ns duplex-link $nodes(bs2) $nodes(c2) 3Mbps 50ms DropTail
}
switch umts {
umts {cell_topo}
}
$ns bandwidth $nodes(bs1) $nodes(ms) $bwDL(cdma) simplex
$ns bandwidth $nodes(ms) $nodes(bs1) $bwUL(cdma) simplex
$ns bandwidth $nodes(bs2) $nodes(ms) $bwDL(cdma) simplex
$ns bandwidth $nodes(ms) $nodes(bs2) $bwUL(cdma) simplex
$ns delay $nodes(bs1) $nodes(ms) $propDL(cdma) simplex
$ns delay $nodes(ms) $nodes(bs1) $propDL(cdma) simplex
$ns delay $nodes(bs2) $nodes(ms) $propDL(cdma) simplex
$ns delay $nodes(ms) $nodes(bs2) $propDL(cdma) simplex
$ns queue-limit $nodes(bs1) $nodes(ms) $buf(cdma)
$ns queue-limit $nodes(ms) $nodes(bs1) $buf(cdma)
$ns queue-limit $nodes(bs2) $nodes(ms) $buf(cdma)
$ns queue-limit $nodes(ms) $nodes(bs2) $buf(cdma)
$ns insert-delayer $nodes(ms) $nodes(bs1) [new Delayer]
$ns insert-delayer $nodes(bs1) $nodes(ms) [new Delayer]
$ns insert-delayer $nodes(ms) $nodes(bs2) [new Delayer]
$ns insert-delayer $nodes(bs2) $nodes(ms) [new Delayer]
set tcp [new Agent/TCP]
$ns attach-agent $nodes(c1) $tcp
set sink [new Agent/TCPSink]
$ns attach-agent $nodes(c2) $sink
set ftp [new Application/FTP]
$ftp attach-agent $tcp
$ns connect $tcp $sink
proc End {} {
global ns nt
$ns flush-trace
close $nt
exec awk -f Lab6.awk Lab6.tr &
exec xgraph -P -bar -x TIME -y DATA cdma.xg
&
exit 0
}
$ns at 0.0 "$ftp start"
$ns at 10.0 "End"
$ns run
```

```
AWK file:
BEGIN {Total_no_of_pkts=0;}
{
if($1 == "r")
{
Total_no_of_pkts = Total_no_of_pkts + $6;
printf("%f %d\n",$2,Total_no_of_pkts) >>
"cdma.xg"
}
}
END{}
```

```
//Part - B : Program - 1 - CRC
//Write a program for error detecting code using CRC - CCITT (16-Bits)
//Filename: P1_crc.java
import java.util.Scanner;
public class P1_crc
{
    public static int a[]=new int [100];
    public static int b[]=new int [100],i,j,len,k,count=0;
    public static int gp[]= {1,0,0,0,1,0,0,0,0,0,0,1,0,0,0,0,1};
    public static void div()
    {
        for(i=0; i<k; i++)
        {
            if(a[i]==gp[0])
            {
                for(j=i; j<17+i; j++)
                    a[j]=a[j]^gp[count++];
            }
            count=0;
        }
    }
    public static void main(String[] args)
    {
        int ch=0;
        Scanner input = new Scanner(System.in);
        System.out.print("\nenter the length of data frame:");
        len = input.nextInt();
        System.out.print("\nenter the message:");
        for(i=0; i<len; i++)
            a[i]=input.nextInt();
        for(i=0; i<16; i++)
            a[len++]=0;
        for(i=0; i<len; i++)
            b[i]=a[i];
        k=len-16;
        div();
        for(i=0; i<len; i++)
            b[i]=b[i]^a[i];
        System.out.print("\nData to be transmitted:");
        for(i=0; i<len; i++)
            System.out.print(b[i]+" ");
        System.out.print("\n\nEnter the recieved data:");
        for(i=0; i<len; i++)
            a[i] = input.nextInt();
        div();
        for(i=0; i<len; i++)
            if(a[i]!=0)
            {
                System.out.println("\n\nERROR in recieved data. . . ");
                System.out.println("\nERROR is in "+(i+1)+"th bit");
                System.out.print("\nRemainder is:");
                for(i=(len-16); i<len; i++)
                    System.out.print(a[i]+" ");
                System.out.println("\n");
                ch=1;
            }
        if(ch==0)
            System.out.println("\nData Recived is ERROR FREE. . .");
    }
}
```

```
/*PART-B: Program - 2
Write a program to find the shortest path using bellman-ford Algorithm.*/

import java.util.Scanner;
public class bellmanford
{
    private int D[];
    private int n;
    public static final int MAX_VALUE=999;

    public bellmanford(int n)
    {
      this.n=n;
      D=new int[n+1];
    }

    public void bellmanfordEvaluation(int source,int A[][])
    {
      for(int node=1;node<=n;node++)
      {
            D[node]=MAX_VALUE;
      }
      D[source]=0;
      for(int node=1;node<=n;node++)
      {
            for(int i=1;i<=n;i++)
            {
                  for(int j=1;j<=n;j++)
                  {
                        if(A[i][j]!=MAX_VALUE)
                        {
                              if(D[j]>D[i]+A[i][j])
                                    D[j]=D[i]+A[i][j];
                        }
                  }
            }
      }
      for(int vertex=1;vertex<=n;vertex++)
            {
                  System.out.println(" Distance of source "+ source +" to "+vertex+"
is "+D[vertex]);
            }
    }

    public static void main(String[]args)
    {
      int n=0;
      int source;
      Scanner Scanner=new Scanner(System.in);
      System.out.print("Enter the number of vertices:");
      n=Scanner.nextInt();
      int A[][]=new int[n+1][n+1];
      System.out.println("Enter the adjacency matrix");
      for(int i=1;i<=n;i++)
      {
            for(int j=1;j<=n;j++)
            {
                  A[i][j]=Scanner.nextInt();
                  if(i==j)
                  {
                        A[i][j]=0;
                        continue;
                  }
                  if(A[i][j]==0)
```

```
                {
                        A[i][j]=MAX_VALUE;
                }
            }
        }
        System.out.println("Enter the source vertex:");
        source=Scanner.nextInt();
        bellmanford b=new bellmanford(n);
        b.bellmanfordEvaluation(source, A);
        Scanner.close();
    }
}
```

```
/*
Program - 3
Using TCP/IP Sockets, Write a client-server program to make the client send the file
name and to make the server send back the contents of the requested file is present.
*/

import java.net.*;
import java.io.*;
class TCPServer
{
    public static void main(String args[]) throws Exception   // establishing the
connection with the server
    {
     ServerSocket sersock = new ServerSocket(4000);
     System.out.println("Server Ready for Connection:");
     Socket sock = sersock.accept(); // binding with port: 4000
     System.out.println("Connection is Successful and Waiting to Serve"); // reading
the file name from client
     InputStream istream = sock.getInputStream();
     BufferedReader fileRead =new BufferedReader(new InputStreamReader(istream));
     String fname = fileRead.readLine(); // reading file contents
     BufferedReader contentRead = new BufferedReader(new FileReader(fname) ); //
keeping output stream ready to send the contents
     OutputStream ostream = sock.getOutputStream();
     PrintWriter pwrite = new PrintWriter(ostream,true);
     String str;
     while((str = contentRead.readLine()) != null)   //reading line-by-line from file
     {
         pwrite.println(str); // sending each line to client
     }
     sock.close();
     sersock.close(); // closing network sockets
     pwrite.close();
     fileRead.close();
     contentRead.close();
    }}
```

//TCP Client//

```
import java.net.*;
import java.io.*;
class TCPClient
{
    public static void main( String args[ ] ) throws Exception
    {
     Socket sock = new Socket( "127.0.0.1", 4000);
     System.out.print("Enter the file name:");
     BufferedReader keyRead = new BufferedReader(new InputStreamReader(System.in));
      String fname = keyRead.readLine(); // sending the file name to server`
      OutputStream ostream = sock.getOutputStream( );
     OutputStream ostream = sock.getOutputStream();
     PrintWriter pwrite = new PrintWriter(ostream, true);
     pwrite.println(fname); // receiving the contents from server. Uses input stream
     InputStream istream = sock.getInputStream();
     BufferedReader socketRead = new BufferedReader(new InputStreamReader(istream));
     String str;
     while((str=socketRead.readLine())!=null) //reading lineby-line
     {
         System.out.println(str);
     }
     pwrite.close();
     socketRead.close();
     keyRead.close();
    }}
```

```
/*PART - B: Program - 4:
Write a program on datagram socket for client/server to display the messages on
client side, typed at the server side.*/

//Client Program
import java.io.*;
import java.net.*;
class UDPC
{
    public static void main(String args[]) throws Exception
    {
    BufferedReader in = new BufferedReader(new InputStreamReader(System.in));
    DatagramSocket clientSocket = new DatagramSocket();
    InetAddress IPAddress = InetAddress.getByName("localhost");
    byte[] sendData = new byte[1024];
    byte[] receiveData = new byte[1024];
    DatagramPacket sendPacket = new DatagramPacket(sendData, sendData.length,
IPAddress, 9876);
    clientSocket.send(sendPacket);
    DatagramPacket receivePacket = new DatagramPacket(receiveData,
receiveData.length);
    clientSocket.receive(receivePacket);
    String modifiedSentence = new String(receivePacket.getData());
    System.out.println("FROM SERVER:" + modifiedSentence);
    clientSocket.close();
    }
}

//Server Program

import java.io.*;
import java.net.*;
class UDPS
{
    public static void main(String args[]) throws Exception
    {
        DatagramSocket serverSocket = new DatagramSocket(9876);
        BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
        byte[] receiveData = new byte[1024];
        byte[] sendData = new byte[1024];
        DatagramPacket receivePacket = new DatagramPacket(receiveData,
receiveData.length);
        serverSocket.receive(receivePacket);
        InetAddress IPAddress = receivePacket.getAddress();
        int port = receivePacket.getPort();
        System.out.println("Enter the Message");
        String data = br.readLine();
        sendData = data.getBytes();
        DatagramPacket sendPacket = new DatagramPacket(sendData, sendData.length,
IPAddress,port);
        serverSocket.send(sendPacket);
        serverSocket.close();
    }
}
```

```
/*PART - B:Program - 5:
Write a program for simple RSA algorithm to encrypt and decrypt the data.*/
import java.util.*;
import java.io.*;
public class RSA
{
    static int gcd(int m,int n){
        while(n!=0){
            int r=m%n;
            m=n;
            n=r;
        }
        return m;
    }
    public static void main(String args[]){
        int p=0,q=0,n=0,e=0,d=0,phi=0;
        int temp[]=new int[100];   // number of messages
        int encrypted[]=new int[100];
        int decrypted[]=new int[100];
        int i=0,j=0,nofelem=0;    // nofelem=number of elements
        Scanner sc=new Scanner(System.in);
        String message ;
        System.out.println("Enter the Message to be encrypted:");
        message= sc.nextLine();
        System.out.println("Enter value of p and q\n");
        p=sc.nextInt();
        q=sc.nextInt();
        n=p*q;
        phi=(p-1)*(q-1);
        for(i=2; i<phi; i++)
            if(gcd(i,phi)==1) break;
        e=i;
        for(i=2; i<phi; i++)
            if((e*i-1)%phi==0)
                break;
        d=i;
        for(i=0; i<message.length(); i++){
            char c = message.charAt(i);
            int a =(int)c;
            temp[i]=c-96;
        }
        nofelem=message.length();
        for(i=0; i<nofelem; i++){
            encrypted[i]=1;
            for(j=0; j<e; j++)
                encrypted[i] =(encrypted[i]*temp[i])%n;
        }
        System.out.println("\n Encrypted message\n");
        for(i=0; i<nofelem; i++)
        {
            System.out.print(encrypted[i]);
            System.out.print((char)(encrypted[i]+96));
        }
        for(i=0; i<nofelem; i++){
            decrypted[i]=1;
            for(j=0; j<d; j++)
                decrypted[i]=(decrypted[i]*encrypted[i])%n;
        }
        System.out.println("\n Decrypted message\n ");
        for(i=0; i<nofelem; i++)
            System.out.print((char)(decrypted[i]+96));
        return;
    }
}
```

```
/*P-6 Write a program for Congestion control using Leaky Bucket Algorithm.*/
import java.util.Random;
import java.io.*;
import java.util.Scanner;

class lb
{
static int t_rand(int n)
{
int rn;
Random r=new Random();
rn=r.nextInt(50);

return rn;
}
public static void main(String args[])
{
int a[]=new int[5];
int buck_rem=0,buck_cap=0,rate=0,i,sent,recieve;
System.out.println("Enter the bucket capacity");
Scanner in=new Scanner(System.in);
buck_cap=in.nextInt();
System.out.println("Enter the rate of transmission");
Scanner input=new Scanner(System.in);
rate=input.nextInt();
for(i=0;i<5;i++)
a[i]=t_rand(6);
System.out.println("CLOCK PACKET_SIZE RECEIVED SENT REMAINING");
for(i=0;i<5;i++)
{
  if(a[i]!=0)
  {
    if((buck_rem+a[i])>buck_cap)
     recieve=-1;
    else
    {
     recieve=a[i];
    buck_rem=buck_rem+a[i];
    }
  }
  else
    recieve=0;
  if(buck_rem!=0)
  {
       if(buck_rem<rate)
       {
          sent=buck_rem;
          buck_rem=0;
       }
       else
       {
       sent=rate;
       buck_rem=buck_rem-rate;
       }
  }
  else
  sent=0;
  if(recieve==-1)
    System.out.println(i+"\t"+a[i]+"\t[dropped]"+sent+"\t"+buck_rem);
  else
    System.out.println(i+"\t"+a[i]+"\t"+recieve+"\t"+sent+"\t"+buck_rem);
  }
}
}
```