CRC .....

```
import java.util.*;
 class crc
   {
     void div(int a[],int k)
      {
      int gp[]={1,0,0,0,1,0,0,0,0,0,0,1,0,0,0,0,1};
      int count=0;
      for(int i=0;i<k;i++)
          {
          if (a[i]==gp[0])
              {
            for(int j=i;j<17+i;j++)
                   {
               a[j]=a[j]^gp[count++];
             }
             count=0;
           }
       }
    }
public static void main(String[] args)
  {
    int a[]=new int[100];
    int b[]=new int[100];
    int len,k;
    crc ob=new crc();
    System.out.println("Enter the length of data frame");
    Scanner sc=new Scanner(System.in);
    len=sc.nextInt();
    int flag=0;
    System.out.println("Enter the message");
    for(int i=0;i<len;i++)
      {
     a[i]=sc.nextInt();
      }
    for(int i=0;i<16;i++)
      {
     a[len++]=0;
      }
     k=len-16;
     for(int i=0;i<len;i++)
       {
      b[i]=a[i];
       }
      ob.div(a,k);
      for(int i=0;i<len;i++)
        {
       a[i]=a[i]^b[i];
      }
      System.out.println("data to be transmitted:");
      for(int i=0;i<len;i++)
      {
        System.out.print(a[i]+ " ");
      }
      System.out.println();
      System.out.println("Enter the reveived data:");
      for(int i=0;i<len;i++)
        {
       a[i]=sc.nextInt();
```

```
    }
    ob.div(a,k);
    for(int i=0;i<len;i++)
    {
      if(a[i]!=0)
           {
           flag=1;
           break;
         }
    }
    if(flag==1)
      {
      System.out.println("Error in reveived data");
    }
      else
        {
          System.out.println("no error");
         }
  }

}
```

Bellmanford ....

```
import java.util.Scanner;
class Edge{
      int src,dest,weight;
      public Edge(int src,int dest,int weight){
            this.src = src;
            this.dest = dest;
            this.weight = weight;
      }
}


public class bellmanford {
      public static void main(String[] args){
            Scanner sc = new Scanner(System.in);
            System.out.println("Enter the number of nodes");
            int n = sc.nextInt();
            int a[][] = new int[n][n];
            System.out.println("Enter the adjacency matrix");
            for(int i=0;i<n;i++){
                  for(int j=0;j<n;j++){
                        System.out.println(i+"->"+j+"\n");
                        a[i][j] = sc.nextInt();
                  }
            }
            Edge edges[] = new Edge[10];
            int x=0;
            for(int i=0;i<n-1;i++){
                  for(int j=i+1;j<n;j++){
                        if(a[i][j]!=0){
                              edges[x] = new Edge(i,j,a[i][j]);
                              x++;
                        }
                  }
            }
            int w[] = new int[n];
            for(int i=0;i<n;i++){
                  w[i]=9999;
            }
            w[0] = 0;
            int y=0;
            while(y!=n-1){
                  for(int i=0;i<x;i++){

      if(w[edges[i].src]+edges[i].weight<w[edges[i].dest]){
                              w[edges[i].dest] =
w[edges[i].src]+edges[i].weight;
                        }
                  }
                  y++;
            }
            for(int i=0;i<n;i++){
                  System.out.print(w[i]+"\t");
            }
      }
}
```

(TCP server ...)

```
import java.io.*;
import java.net.ServerSocket;
import java.net.Socket;

TCP SERVER...

public class TCPServer
{
      public static void main(String args[]) throws Exception
      {
            ServerSocket sersock = new ServerSocket(6000);
            System.out.println("Server ready for connection");
            Socket sock = sersock.accept();

            System.out.println("Connection successful | wating for
filename");
            InputStream istream = sock.getInputStream( );
            BufferedReader br =new BufferedReader(new
InputStreamReader(istream));
            String fname = br.readLine( );
            BufferedReader contentRead = new BufferedReader(new
FileReader(fname) );

            OutputStream ostream = sock.getOutputStream( );
            PrintWriter pwrite = new PrintWriter(ostream, true);
            String str;

            while((str = contentRead.readLine()) != null)
            {
                  pwrite.println(str);
            }
            System.out.println("File Contents sent successfully");
            sock.close(); sersock.close();
            pwrite.close(); br.close(); contentRead.close();
      }
}
```

(TCP CLIENT....)

```
import java.net.*;
import java.io.*;
public 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\n");
            BufferedReader br = new BufferedReader(new
InputStreamReader(System.in));
            String fname = br.readLine();

            OutputStream ostream = sock.getOutputStream( );
            PrintWriter pwrite = new PrintWriter(ostream, true);
            pwrite.println(fname);

            InputStream istream = sock.getInputStream();
            BufferedReader socketRead = new
BufferedReader(newInputStreamReader(istream));
            String str;

            while((str = socketRead.readLine()) != null) // reading line-
by-line
            {
                  System.out.println(str);
            }

            pwrite.close(); socketRead.close(); br.close(); sock.close();
      }
}
```

(UDP server .....)

```
import java.io.*;
import java.net.*;
class UDPServer
{
      public static DatagramSocket serversocket;
      public static DatagramPacket dp;
      public static BufferedReader br;
      public static InetAddress ia;
      public static byte buf[] = new byte[1024];
      public static int cport = 222,sport=555;
      public static void main(String[] args) throws IOException
        {
          serversocket = new DatagramSocket(sport);
          dp = new DatagramPacket(buf,buf.length);
          br = new BufferedReader (new InputStreamReader(System.in));
          ia = InetAddress.getLocalHost();

          System.out.println("Server is Running...");
          while(true)
          {
           serversocket.receive(dp);
           String str2 = new String(dp.getData(), 0, dp.getLength());

            if(str2.equals("exit"))
           {
            System.out.println("Terminated...");
            break;
           }
           System.out.println("Client said : " + str2);

           String str3 = new String(br.readLine());
           buf = str3.getBytes();
           serversocket.send(new DatagramPacket(buf,str3.length(), ia,
cport));
          }
        }
    }
```

(UDP clinet ....)

```
import java.io.*;
import java.net.*;
class UDPClient
{
      public static DatagramSocket clientsocket;
      public static DatagramPacket dp;
      public static BufferedReader br;
      public static InetAddress ia;
      public static byte buf[] = new byte[1024];
      public static int cport = 222, sport = 555;
      public static void main(String[] args) throws IOException
      {
            clientsocket = new DatagramSocket(cport);
            dp = new DatagramPacket(buf, buf.length);
            br = new BufferedReader(new InputStreamReader(System.in));
            ia = InetAddress.getLocalHost();

            System.out.println("Client is Running...");
            System.out.println("Type some text if u want to Quit type
'exit'.");
            while(true)
            {
                  String str1 = new String(br.readLine());
                  buf = str1.getBytes();
                  if(str1.equals("exit"))
                  {
                        System.out.println("Terminated..");
                        clientsocket.send(new
DatagramPacket(buf,str1.length(),ia,sport));
                        break;
                  }
                  clientsocket.send(new DatagramPacket(buf,str1.length(),
ia, sport));

                  clientsocket.receive(dp);
                  String str4 = new String(dp.getData(), 0,
dp.getLength());
                  System.out.println("Server said : " + str4);
            }
      }
}
```

```
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 nummes[]=new int[100];
      int encrypted[]=new int[100];
      int decrypted[]=new int[100];
      int i=0,j=0,nofelem=0;
      Scanner sc=new Scanner(System.in);
      String message ;

      System.out.println("Enter the Message tobe 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;
            nummes[i]=a-96;
      }
      nofelem=message.length();

      for(i=0;i<nofelem;i++)
      {
            encrypted[i]=1;
            for(j=0;j<e;j++)
            encrypted[i] =(encrypted[i]*nummes[i])%n;
      }

      System.out.println("\n Encrypted message\n");

      for(i=0;i<nofelem;i++)
```

```
{
      System.out.print(encrypted[i]);

}


for(i=0;i<nofelem;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(decrypted[i]);

}
for(i=0;i<nofelem;i++){

System.out.print((char)(decrypted[i]+96));
}
return;
}
}
```

leaky bucket ....

```
import java.util.*;
public class leakyass
{
      public static void main(String args[])
      {
            Scanner sc=new Scanner(System.in);
            int max,or,ip,c=0;
            System.out.println("Enter maximum bucket size:");
            max=sc.nextInt();
            System.out.println("Enter output rate:");
            or=sc.nextInt();
            System.out.println("Type 1 to exit");
            while(true)
            {
                  System.out.println("Enter input packets:");
                  ip=sc.nextInt();
                  if(ip==1)
                        System.exit(0);
                  System.out.println("fill "+ip);
                  c=c+ip;
                  if(c>max)
                  {
                        System.out.println("overflow has occured of "+(c-
max));
                        c=max;
                  }
                  c=c-or;
                  if(c<0)
                        c=0;
                  System.out.println("output rate is "+or);
                  System.out.println("bucket has "+c);
            }
      }
}
```