

1: Write a program for error detecting code using CRC-CCITT (16-bits).

```
import java.io.*;
import java.lang.*;
import java.util.*;
class Main
{
    public static String string_val(String sts,int poly_length)
    {
        for(int i=1;i<poly_length;i++)
        {
            sts=sts+"0";
        }
        return sts;
    }

    public static String generate(char[] divisor,char[] dividend,int
len,String org)
    {
        for(int i=0;i<len;i++)
        {
            if(dividend[i]=='1')
            {
                for(int j=0;j<divisor.length;j++)
                {
                    if(dividend[i+j]==divisor[j])
                    {
                        dividend[i+j]='0';
                    }
                    else
                    {
                        dividend[i+j]='1';
                    }
                }
            }
        }
        String st=String.valueOf(dividend);

        String fin=org+st.substring(len);

        return fin;
    }

    public static void main(String[] args)
    {
        String str,rec;
        String d="10001000000100001";
```

```

Scanner sc=new Scanner(System.in);
System.out.println("Enter the string");
str=sc.next();

String org=str;

//  System.out.println("ORIGINAL STRING IS " + org);

int len=str.length();

str=string_val(str,d.length());

char[] divisor=d.toCharArray();
char[] dividend=str.toCharArray();

String fin=generate(divisor,dividend,len,org);

System.out.println("DIVISIOR= " + String.valueOf(divisor));
System.out.println("DIVIDEND= " + String.valueOf(dividend));

System.out.println("TRANSMITTED MESSAGE IS  " + fin);

System.out.println("Enter the received message");
rec=sc.next();
org=rec;

len=rec.length();

rec=string_val(rec,d.length());

dividend=rec.toCharArray();

String rin=generate(divisor,dividend,len,org);

System.out.println("MESSAGE DUE TO ERRORS IS");
System.out.println(rin);
if(fin.equals(rin))
{
    System.out.println("NO ERRORS");
}
else
{
    System.out.println("ERRORS REPORTED");
}

//  System.out.println

```

```
}  
}
```

```
Enter the string  
11111  
DIVISIOR= 100010000000100001  
DIVIDEND= 000001110001111011110  
TRANSMITTED MESSAGE IS 111111110001111011110  
Enter the received message  
1111  
MESSAGE DUE TO ERRORS IS  
11111111000111101111  
ERRORS REPORTED
```

```
...Program finished with exit code 0  
Press ENTER to exit console.□
```

```
Enter the string
11111
DIVISIOR= 10001000000100001
DIVIDEND= 000001110001111011110
TRANSMITTED MESSAGE IS 111111110001111011110
Enter the received message
11111
MESSAGE DUE TO ERRORS IS
111111110001111011110
NO ERRORS
```

```
...Program finished with exit code 0
Press ENTER to exit console.
```

2: Write a program for distance vector algorithm to find suitable path for transmission.

```
#include<stdio.h>
struct node
{
    unsigned dist[20];
    unsigned from[20];
}rt[10];
int main()
{
    int dmat[20][20];
    int n,i,j,k,count=0;
    printf("\nEnter the number of nodes : ");
    scanf("%d",&n);
    printf("\nEnter the cost matrix :\n");
    for(i=0;i<n;i++)
        for(j=0;j<n;j++)
        {
            scanf("%d",&dmat[i][j]);
            dmat[i][i]=0;
            rt[i].dist[j]=dmat[i][j];
            rt[i].from[j]=j;
        }
    do
    {
        count=0;
        for(i=0;i<n;i++)
            for(j=0;j<n;j++)
                for(k=0;k<n;k++)
                    if(rt[i].dist[j]>dmat[i][k]+rt[k].dist[j])
                    {
                        rt[i].dist[j]=rt[i].dist[k]+rt[k].dist[j];
                        rt[i].from[j]=k;
                        count++;
                    }
    }while(count!=0);
    for(i=0;i<n;i++)
    {
        printf("\n\nState value for router %d is \n",i+1);
        for(j=0;j<n;j++)
        {
            printf(" \t\nnode %d via %d Distance%d",j+1,rt[i].from[j]+1,rt[i].dist[j]);
        }
    }
    printf("\n\n");
}
```

OUTPUT:

```
Enter the number of nodes : 4
```

```
Enter the cost matrix :
```

```
0 3 5 99
3 0 99 1
5 99 0 2
99 1 2 0
```

```
State value for router 1 is
```

```
node 1 via 1 Distance0
node 2 via 2 Distance3
node 3 via 3 Distance5
node 4 via 2 Distance4
```

```
State value for router 2 is
```

```
node 1 via 1 Distance3
node 2 via 2 Distance0
node 3 via 4 Distance3
node 4 via 4 Distance1
```

```
State value for router 3 is
```

```
node 1 via 1 Distance5
node 2 via 4 Distance3
node 3 via 3 Distance0
node 4 via 4 Distance2
```

```
State value for router 4 is
```

```
node 1 via 2 Distance4
node 2 via 2 Distance1
node 3 via 3 Distance2
node 4 via 4 Distance0
```

3: Implement Dijkstra's algorithm to compute the shortest path for a given topology.

```
import java.io.*;
import java.lang.*;
import java.util.*;
public class djksitra
{
    public int graph[][];
    int src;
    int n;
    public djksitra(int n,int src)
    {
        this.n=n;
        graph=new int[n][n];
        this.src=src;
    }

    public void input()
    {
        Scanner sc=new Scanner(System.in);
        System.out.println("Enter the Distance matrix");
        for(int i=0;i<n;i++)
        {
            for(int j=0;j<n;j++)
            {
                graph[i][j]=sc.nextInt();
            }
        }
    }

    public int mindistance(int dist[],Boolean val[])
    {
        int min=Integer.MAX_VALUE;
        int min_index=-1;

        for(int i=0;i<dist.length;i++)
        {
            if(dist[i]<min && val[i]==false)
            {
```

```

        min=dist[i];
        min_index=i;
    }

}
return min_index;
}

```

```

public int[] dj()
{
    int dist[]=new int[n];
    Boolean val[]=new Boolean[n];

    Arrays.fill(dist,Integer.MAX_VALUE);
    Arrays.fill(val,false);

    dist[src]=0;

    for(int i=0;i<n-1;i++)
    {
        int u=mindistance(dist,val);

        val[u]=true;

        for(int j=0;j<n;j++)
        {
            if(val[j]==false && graph[u][j]!=0 && dist[u] +
graph[u][j]<dist[j])
            {
                dist[j]=dist[u] + graph[u][j];
            }
        }
    }
    return dist;
}
}

```



## MAIN.JAVA

```
import java.io.*;
import java.util.*;
class Main
{
    public static void main(String[] args)
    {
        int n;
        int src;
        Scanner sc=new Scanner(System.in);

        System.out.println("Enter the Number of nodes");
        n=sc.nextInt();

        System.out.println("Enter the source");
        src=sc.nextInt();

        djksitra ob=new djksitra(n,src);

        ob.input();

        int[] ans=ob.dj();

        System.out.println("SHORTEST DISTANCE FROM THE
SOURCE " + src);
        for(int i=0;i<ans.length;i++)
        {
            System.out.println(src + " -----> " + i + " " +
ans[i]);
        }
    }
}
```

```
Enter the Number of nodes
9
Enter the source
0
Enter the Distance matrix
0 4 0 0 0 0 0 8 0
4 0 8 0 0 0 0 11 0
0 8 0 7 0 4 0 0 2
0 0 7 0 9 14 0 0 0
0 0 0 9 0 10 0 0 0
0 0 4 14 10 0 2 0 0
0 0 0 0 0 2 0 1 6
8 11 0 0 0 0 1 0 7
0 0 2 0 0 0 6 7 0
SHORTEST DISTANCE FROM THE SOURCE 0
0 -----> 0 0
0 -----> 1 4
0 -----> 2 12
0 -----> 3 19
0 -----> 4 21
0 -----> 5 11
0 -----> 6 9
0 -----> 7 8
0 -----> 8 14
```

```
...Program finished with exit code 0
Press ENTER to exit console.[]
```

4:Write a program for congestion control using Leaky bucket algorithm.

```
import java.io.*;
import java.lang.*;
import java.util.*;
public class leakey_bucket
{

    public int no_of_packets;
    public int bucket_size;
    public int o_rate;
    public int packet_size_remaining;
    public int op;
    public int packet_size[];

    leakey_bucket(int size)
    {
        no_of_packets=size;
        bucket_size=0;
        o_rate=0;
        packet_size_remaining=0;
        op=0;
        packet_size=new int[size];
    }
    public void input()
    {
        Scanner sc=new Scanner(System.in);

        System.out.println("Enter the bucket size");
        bucket_size=sc.nextInt();

        System.out.println("Enter the Output Rate");
        o_rate=sc.nextInt();

        System.out.println("Enter the size of the each of the packets");

        for(int i=0;i<no_of_packets;i++)
        {
            packet_size[i]=sc.nextInt();
        }

    }
    public void calc()
    {
```

```

for(int i=0;i<no_of_packets;i++)
{
    if(packet_size[i] + packet_size_remaining > bucket_size)
    {
        if(packet_size[i]>bucket_size)
        {
            System.out.println("INCOMING PACKET SIZE " +
packet_size[i] + " GREATER THAN BUCKET CAPACITY " +
bucket_size);
            System.out.println();
        }
        else
        {
            System.out.println("BUCKET CAPACITY EXCEEDED
PACKET_REJECTED");
        }
    }
    else
    {
        packet_size_remaining+=packet_size[i];
        System.out.println("INCOMING PACKET SIZE " +
packet_size[i]);
        System.out.println("BYTE REMAINING TO BE
TRANSMITTED " + packet_size_remaining);
        System.out.println();
        while(packet_size_remaining>0)
        {
            if(packet_size_remaining>0)
            {
                if(packet_size_remaining<=o_rate)
                {
                    op=packet_size_remaining;
                    packet_size_remaining=0;
                }
                else
                {
                    op=o_rate;
                    packet_size_remaining-=o_rate;
                }
            }
            System.out.println("PACKET SIZE TRANSMITTED "
+ op);
            System.out.println("BYTES REMAINING " +
packet_size_remaining);
            System.out.println();
        }
    }
}

```



## Output:

```
Enter the number of packets
5
Enter the bucket size
85
Enter the Output Rate
30
Enter the size of the each of the packets
83
86
77
15
93
INCOMING PACKET SIZE 83
BYTE REMAINING TO BE TRANSMITTED 83

PACKET SIZE TRANSMITTED 30
BYTES REMAINING 53

PACKET SIZE TRANSMITTED 30
BYTES REMAINING 23

PACKET SIZE TRANSMITTED 23
BYTES REMAINING 0

INCOMING PACKET SIZE 86 GREATER THAN BUCKET CAPACITY 85
INCOMING PACKET SIZE 77
BYTE REMAINING TO BE TRANSMITTED 77

PACKET SIZE TRANSMITTED 30
BYTES REMAINING 47

PACKET SIZE TRANSMITTED 30
BYTES REMAINING 17

PACKET SIZE TRANSMITTED 17
BYTES REMAINING 0

INCOMING PACKET SIZE 15
BYTE REMAINING TO BE TRANSMITTED 15

PACKET SIZE TRANSMITTED 15
```

5:Using TCP/IP sockets, write a client-server program to make client sending the file name and the server to send back the contents of the requested file if present.

```
import java.io.*;
import java.net.*;
import java.lang.*;
import java.util.*;
public class tcpclient
{
    public static void main(String[] args)
    {
        try
        {
            Scanner sc=new Scanner(System.in);

            Socket s=new Socket("127.0.0.1",12000);

            DataOutputStream dos=new
DataOutputStream(s.getOutputStream());

            DataInputStream dis=new
DataInputStream(s.getInputStream());

            dos.writeUTF("CONNECTED TO 127.0.0.1\n");

            System.out.println("Enter the path of the file");

            String path=sc.nextLine();
            dos.writeUTF(path);

            System.out.println(new String(dis.readUTF()));

            dis.close();
            dos.close();
            sc.close();
            s.close();
        }
        catch(Exception e)
```

```

        {
            System.out.println(e);
        }
    }
}

```

```

import java.io.*;
import java.net.*;
import java.lang.*;
import java.util.*;
public class tcpserver
{
    public static void main(String[] args)
    {
        try
        {
            ServerSocket s=new ServerSocket(12000);
            System.out.println("SERVER WAITING FOR
CONNECTION ..... ");

            Socket s1=s.accept();

            DataOutputStream dos=new
DataOutputStream(s1.getOutputStream());

            DataInputStream dis=new
DataInputStream(s1.getInputStream());

            System.out.println("STATUS: " + dis.readUTF());

            String path=dis.readUTF();

            System.out.println("REQUEST HAS BEEN RECEIVED
");

            try
            {

```



```

        File mf=new File(path);
        Scanner sc=new Scanner(mf);
        String st=sc.nextLine();
        while(sc.hasNextLine())
        {
            st=st+"\n" + sc.nextLine();
        }
        dos.writeUTF(st);
        dos.close();
        s1.close();
        s.close();
        sc.close();
    }
    catch(FileNotFoundException e)
    {
        dos.writeUTF("FILE NOT FOUND ");
    }
}
catch(IOException e)
{
    System.out.println(e);
}
}
}

```

# TCP CLIENT

```
FILE NOT FOUND
(base) hari@hari-Lenovo-ideapad-330S-15IKB:~$ javac tcpclient.java
(base) hari@hari-Lenovo-ideapad-330S-15IKB:~$ java tcpclient
java.net.ConnectException: Connection refused
(base) hari@hari-Lenovo-ideapad-330S-15IKB:~$ java tcpclient
Enter the path of the file
/home/hari/avl.c
#include<stdio.h>
#include<stdlib.h>
struct node
{
    int key;
    struct node *left;
    struct node *right;
    int height;
};
typedef struct node *NODE;

int max(int a,int b);
int height(NODE n)
{
    if(n==NULL)
        return 0;
    return n->height;
}
int max(int a,int b)
{
    if(a>b)
        return a;
    else
        return b;
}

NODE newnode(int key)
{
    NODE newnode=(NODE)malloc(sizeof(struct node));

    newnode->key=key;
    newnode->left=NULL;
    newnode->right=NULL;
    newnode->height=1;

    return(newnode);
}
NODE rightright(NODE y)
{
    NODE x=y->left;
    NODE t2=x->right;

    x->right=y;
    y->left=t2;
}
```

## TCP SERVER

```
error: Class names, 'tcpserver', are only accepted if annotation processing is explicitly  
1 error  
(base) hari@hari-Lenovo-ideapad-330S-15IKB:~$ javac tcpserver.java  
(base) hari@hari-Lenovo-ideapad-330S-15IKB:~$ java tcpserver  
java.net.BindException: Permission denied  
(base) hari@hari-Lenovo-ideapad-330S-15IKB:~$ javac tcpserver.java  
(base) hari@hari-Lenovo-ideapad-330S-15IKB:~$ java tcpserver  
java.net.BindException: Permission denied  
(base) hari@hari-Lenovo-ideapad-330S-15IKB:~$ javac tcpserver.java  
(base) hari@hari-Lenovo-ideapad-330S-15IKB:~$ java tcpserver  
java.net.BindException: Permission denied  
(base) hari@hari-Lenovo-ideapad-330S-15IKB:~$ javac tcpserver.java  
(base) hari@hari-Lenovo-ideapad-330S-15IKB:~$ java tcpserver  
SERVER WAITING FOR CONNECTION .....  
^C(base) hari@hari-Lenovo-ideapad-330S-15IKB:~$ javac tcpserver.java  
(base) hari@hari-Lenovo-ideapad-330S-15IKB:~$ java tcpserver  
SERVER WAITING FOR CONNECTION .....  
^C(base) hari@hari-Lenovo-ideapad-330S-15IKB:~$ javac tcpserver.java  
(base) hari@hari-Lenovo-ideapad-330S-15IKB:~$ java tcpserver  
SERVER WAITING FOR CONNECTION .....  
SERVER CONNECTED.....  
STATUS CONNECTED TO 192.168.0.166  
  
REQUEST HAS BEEN RECEIVED  
(base) hari@hari-Lenovo-ideapad-330S-15IKB:~$ javac tcpserver.java  
(base) hari@hari-Lenovo-ideapad-330S-15IKB:~$ java tcpserver  
SERVER WAITING FOR CONNECTION .....  
SERVER CONNECTED.....  
STATUS CONNECTED TO 192.168.0.166  
  
REQUEST HAS BEEN RECEIVED  
THE CONTENT OF THE FILE IS #include<stdio.h>  
(base) hari@hari-Lenovo-ideapad-330S-15IKB:~$
```

6:Using UDP sockets, write a client-server program to make client sending the file name and the server to send back the contents of the requested file if present.

```
import java.util.*;
import java.io.*;
import java.lang.*;
import java.net.*;
public class udpclient
{
    public static void main(String[] args)
    {
        try
        {
            InetAddress
address=InetAddress.getByName("127.0.0.1");
            int port=2000;
            DatagramSocket socket=new DatagramSocket();

            Scanner sc=new Scanner(System.in);
            String name;
            System.out.println("Enter the name of the file");
            name=sc.nextLine();

            byte[] buffer=name.getBytes();
            DatagramPacket request=new
DatagramPacket(buffer,buffer.length,address,port);
            socket.send(request);

            byte[] respo=new byte[512];
            DatagramPacket response=new
DatagramPacket(respo,respo.length);
            socket.receive(response);
            String cont=new
String(respo,0,response.getLength());
            System.out.println(cont);
        }
    }
}
```

```

        catch(Exception e)
        {
            System.out.println(e);
        }
    }
}

```

```

import java.io.*;
import java.lang.*;
import java.net.*;
import java.util.*;
public class udpserver
{
    public static void main(String[] args)
    {
        try{

            byte[] buffer=new byte[512];

            DatagramSocket socket=new
            DatagramSocket(2000);

            DatagramPacket request = new
            DatagramPacket(buffer, buffer.length);
            socket.receive(request);

            String path = new String(buffer, 0,
            request.getLength());

            System.out.println("REQUEST RECEIVED");

            try
            {
                File mf=new File(path);
                Scanner sc=new Scanner(mf);
                String nt=sc.nextLine();
            }
            catch(Exception e)
            {
                System.out.println(e);
            }
        }
    }
}

```

```

while(sc.hasNextLine())
{
    nt=nt+"\n" + sc.nextLine();
}
//System.out.println(nt);
InetAddress clientAddress=request.getAddress();
int clientPort=request.getPort();

byte[] ans=nt.getBytes();
DatagramPacket response = new
DatagramPacket(ans, ans.length, clientAddress,clientPort);
socket.send(response);
}
catch(FileNotFoundException e)
{
    String resp="FILE NOT FOUND";
    InetAddress clientAddress=request.getAddress();
    int clientPort=request.getPort();

    byte[] ans=resp.getBytes();
    DatagramPacket response = new
DatagramPacket(ans, ans.length, clientAddress,clientPort);
    socket.send(response);
}
}catch(Exception e)
{
    System.out.println(e);
}
}
}

```

## UDP CLIENT:

```
(base) hari@hari-Lenovo-ideapad-330S-15IKB:~$ javac udpclient.java
(base) hari@hari-Lenovo-ideapad-330S-15IKB:~$ java udpclient
Enter the name of the file
/home/hari/abc.txt
Hello
How are You
I am doing great, I Hope You are also doing great
1      Haryana
2      Karnatak
3      Tamil Nadu
4
5
(base) hari@hari-Lenovo-ideapad-330S-15IKB:~$
```

## UDP SERVER:

```
(base) hari@hari-Lenovo-ideapad-330S-15IKB:~$ javac udpserver.java
(base) hari@hari-Lenovo-ideapad-330S-15IKB:~$ java udpserver
REQUEST RECEIVED
(base) hari@hari-Lenovo-ideapad-330S-15IKB:~$ █
```