

PRAC 3:

CLIENT SERVER

SERVER:

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

public class MyServer {

    public static void main(String[] args){
        try{
            ServerSocket ss=new ServerSocket(6666);
            Socket s=ss.accept();//establishes connection
            DataInputStream dis=new DataInputStream(s.getInputStream());
            String str=(String)dis.readUTF();
            System.out.println("message= "+str);
            ss.close();
        }catch(Exception e){System.out.println(e);}
    }
}
```

CLIENT:

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

public class MyClient {

    public static void main(String[] args) {
        try{
            Socket s=new Socket("localhost",6666);
            DataOutputStream dout=new DataOutputStream(s.getOutputStream());
            dout.writeUTF("Hello Server");
            dout.flush();
            dout.close();
            s.close();
        }catch(Exception e){System.out.println(e);}
    }
}
```

PRAC 4:

MANCHESTER ENCODING

CLIENT:

```
import java.io.*;

import java.net.*;

import java.util.*;

public class MyClient {

    public static void main(String[] args) {

        try{

            Socket s=new Socket("localhost",6666);

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

            Scanner sc = new Scanner(System.in);

            System.out.println("enter the data want to send ");

            String mystr = sc.nextLine();

            char[] ch = new char[mystr.length()];

            for(int i = 0;i<mystr.length();i++)

            {

                ch[i] = mystr.charAt(i);

            }

            String str = "";

            for(int i = 0;i<mystr.length();i++)

            {

                if(ch[i]=='1')

                {

                    str = str.concat("10");

                }

                if(ch[i]=='0')

                {

                    str = str.concat("01");

                }

            }

            System.out.println("Actual String is :-- "+mystr);

            System.out.println("Encoded string is :-- "+str);

            dout.writeUTF(str);

            dout.flush();
```

```

dout.close();

s.close();

sc.close();

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

}

}

```

SERVER:

```

import java.io.*;

import java.net.*;

public class MyServer {

public static void main(String[] args){

try{

ServerSocket ss=new ServerSocket(6666);

Socket s=ss.accept();//establishes connection

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

String cstr=(String)dis.readUTF();

char[] ch = new char[cstr.length()];

System.out.println("get "+cstr);

for(int i = 0;i<cstr.length();i++)

{

ch[i] = cstr.charAt(i);

}

String mystring="";

for(int i=0;i<cstr.length();i=i+2)

{

if(ch[i]=='1'&&ch[i+1]=='0')

{

mystring = mystring.concat("1");

}

else if(ch[i]=='0'&&ch[i+1]=='1')

{

mystring = mystring.concat("0");

}

}

}

}

```

```
}
```

```
System.out.println("Encoded string is :-- "+cstr);  
System.out.println("Actual String is :-- "+mystring);
```

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

DIFFERENTIAL MANCHESTOR ENCODING

CLIENT:

```
import java.io.*;  
import java.net.*;  
import java.util.*;  
public class MyClientDiff {  
    public static void main(String[] args) {  
        try{  
            Socket s=new Socket("localhost",6666);  
            DataOutputStream dout=new DataOutputStream(s.getOutputStream());  
            Scanner sc = new Scanner(System.in);  
            System.out.println("enter the data want to send ");  
            String mystr = sc.nextLine();  
  
            char[] ch = new char[mystr.length()];  
  
            for(int i = 0;i<mystr.length();i++)  
            {  
                ch[i] = mystr.charAt(i);  
            }  
  
            String str = "";  
            if(ch[0] == '1'){  
                str = str.concat("10");  
            }  
            else{  
                str = str.concat("01");  
            }  
            for(int i = 1;i<mystr.length();i++)  
            {
```

```

        int curr = str.length()-1;

        char cha = str.charAt(curr);

        if(ch[i] == '1')
        {
            if(cha == '0'){
                str = str.concat("01");
            }

            else{
                str = str.concat("10");
            }
        }

        else{
            if(cha == '0'){
                str = str.concat("10");
            }

            else{
                str = str.concat("01");
            }
        }
    }

    System.out.println("Actual String is :- "+mystr);
    System.out.println("Encoded string is :- "+str);

```

```

dout.writeUTF(str);
dout.flush();
dout.close();
s.close();
sc.close();
}catch(Exception e){System.out.println(e);}
}
}

```

SERVER DIFFERENTIAL:

```

import java.io.*;
import java.net.*;

public class MyServerDiff {

    public static void main(String[] args) {

        try {

```

```

ServerSocket ss = new ServerSocket(6666);

Socket s = ss.accept();// establishes connection

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

String cstr = (String) dis.readUTF();

char[] ch = new char[cstr.length()];


System.out.println("get " + cstr);


for (int i = 0; i < cstr.length(); i++) {

    ch[i] = cstr.charAt(i);

}


String mystring = "";

if (ch[0] == '1' && ch[1] == '0') {

    mystring = mystring.concat("1");

} else {

    mystring = mystring.concat("0");

}

for (int i = 2; i < cstr.length(); i = i + 2) {

    if ((ch[i - 1] == '0' && ch[i] == '1') || (ch[i - 1] == '1' && ch[i] == '0')) {

        mystring = mystring.concat("0");

    } else if ((ch[i - 1] == '0' && ch[i] == '0') || (ch[i - 1] == '1' && ch[i] == '1')) {

        mystring = mystring.concat("1");

    }

}


System.out.println("Encoded string is :- " + cstr);

System.out.println("Actual String is :- " + mystring);


ss.close();

} catch (Exception e) {

    System.out.println(e);

}

}

}

```

Prac 5

CODE (Bit Stuffing):-

```
import java.util.*;

public class bit_stuffing

{

    public static void main(String[] args)

    {

        Scanner sc=new Scanner(System.in);

        System.out.print("Enter the message:-");

        String d1 = sc.nextLine();

        String remaining = new String();

        String output=new String();

        int counter = 0;

        for(int i=0;i<d1.length();i++)

        {

            if (d1.charAt(i)!='1' && d1.charAt(i)!='0')

            {

                System.out.println("Enter valid Binary values");

                return;

            }

            if(d1.charAt(i) == '1')

            {

                counter++;

                remaining = remaining + d1.charAt(i);

            }

            else

            {

                remaining = remaining + d1.charAt(i);

                counter = 0;

            }

            if(counter == 5)

            {

                remaining = remaining + '0';

                counter = 0;

            }

        }

        System.out.println("Flag--> 01111110");

        String new1="|01111110 | "+remaining+" | 01111110|";

        System.out.println("Stuffed data at intermediate site is:");
```

```

for(int k=0;k<=(28+d1.length());k++)
{
    System.out.print("-");
}
System.out.println();
System.out.println(" "+new1);
for(int k=0;k<=(28+d1.length());k++)
{
    System.out.print("-");
}
System.out.println();
counter=0;
for(int i=0;i<remaining.length();i++)
{
    if(remaining.charAt(i) == '1')
    {
        counter++;
        output = output + remaining.charAt(i);
    }
    else
    {
        output = output + remaining.charAt(i);
        counter = 0;
    }
    if(counter == 5)
    {
        if((i+2)!=remaining.length())
        {
            output = output + remaining.charAt(i+2);
        }
        else
        {
            output=output + '1';
        }
        i=i+2;
        counter = 1;
    }
}
System.out.println("Destuffed BIT is: "+output);}

```


CODE (CHARACTER STUFFING):-

```
import java.util.*;

class character_stuffing
{
    public static void main(String r[])
    {
        Scanner sc=new Scanner(System.in);
        System.out.println("Enter number of characters: ");
        int n=sc.nextInt();
        String in[]=new String[n];
        for(int i=0;i<n;i++)
        {
            in[i]=sc.next();
        }
        for(int i= 0;i<n;i++)
        {
            if(in[i].equals("dle"))
            {
                in[i]="dle dle";
            }
        }
        System.out.println("Transmitted message is: ");
        System.out.print(" Hello there! ");
        for(int i=0;i<n;i++)
        {
            System.out.print(in[i]+" ");
        }
        System.out.println(" is Character Stuffing ");
    }
}
```

PRAC 6: CRC

```
// Include headers
#include<stdio.h>
#include<string.h>

// length of the generator polynomial
#define N strlen(gen_poly)

// data to be transmitted and received
char data[28];

// CRC value
```

```

char check_value[28];

// generator polynomial

char gen_poly[10];

// variables

int data_length,i,j;

// function that performs XOR operation

void XOR(){

    // if both bits are the same, the output is 0

    // if the bits are different the output is 1

    for(j = 1;j < N; j++)

        check_value[j] = (( check_value[j] == gen_poly[j])?'0':'1');

}

// Function to check for errors on the receiver side

void receiver(){

    // get the received data

    printf("Enter the received data: ");

    scanf("%s", data);

    printf("\n-----\n");

    printf("Data received: %s", data);

    // Cyclic Redundancy Check

    crc();

    // Check if the remainder is zero to find the error

    for(i=0;(i<N-1) && (check_value[i]!='1');i++);

    if(i<N-1)

        printf("\nError detected\n\n");

    else

        printf("\nNo error detected\n\n");

}

void crc(){

    // initializing check_value

    for(i=0;i<N;i++)

        check_value[i]=data[i];

    do{

        // check if the first bit is 1 and calls XOR function

        if(check_value[0]=='1')

            XOR();

        // Move the bits by 1 position for the next computation

        for(j=0;j<N-1;j++)

            check_value[j]=check_value[j+1];

```

```

// appending a bit from data
check_value[j]=data[i++];
}while(i<=data_length+N-1);
// loop until the data ends
}

int main()
{
// get the data to be transmitted
printf("\nEnter data to be transmitted: ");
scanf("%s",data);

printf("\n Enter the Generating polynomial: ");
// get the generator polynomial
scanf("%s",gen_poly);
// find the length of data
data_length=strlen(data);
// appending n-1 zeros to the data
for(i=data_length;i<data_length+N-1;i++)
data[i]='0';
printf("\n-----");
// print the data with padded zeros
printf("\n Data padded with n-1 zeros : %s",data);
printf("\n-----");
// Cyclic Redundancy Check
crc();
// print the computed check value
printf("\nCRC or Check value is : %s",check_value);
// Append data with check_value(CRC)
for(i=data_length;i<data_length+N-1;i++)
data[i]=check_value[i-data_length];
printf("\n-----");
// printing the final data to be sent
printf("\n Final data to be sent : %s",data);
printf("\n-----\n");
// Calling the receiver function to check errors
receiver();

return 0;

```

PRAC 7:

CODE: (Stop and Wait for Protocol):

Sender.java

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

public class Sender extends Thread {

    public static void main(String args[])
    {
        int p = 9000, i, q = 8000;

        String h = "localhost";

        try
        {
            Scanner scanner = new Scanner(System.in);

            System.out.print("Enter number of frames : ");

            int number = scanner.nextInt();

            if (number == 0)
            {
                System.out.println("No frame is sent");
            }
            else
            {
                Socket s2;

                s2 = new Socket(h, q);

                DataOutputStream d1 = new
                DataOutputStream(s2.getOutputStream());

                d1.write(number);

            }

            String str1;

            for (i = 0; i < number; i++)
            {
                System.out.print("Enter message : ");

                String name = scanner.next();

                System.out.println("Frame " + i + " is sent");

                Socket s1;

                s1 = new Socket(h, p + i);

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

                d.writeUTF(name);
```

```

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

Integer sss1 = dd.read();

System.out.println("Ack for : " + sss1 + " is received");

}

}catch (Exception ex){System.out.println("ERROR : " + ex);}}}

```

Receiver.java

```

import java.io.*;
import java.net.*;
import java.util.*;

public class Receiver extends Thread {

    public void run() {

        String h = "Serverhost";

        int q = 5000;

        int i;

        try {

            ServerSocket ss2;

            ss2 = new ServerSocket(8000);

            Socket s1 = ss2.accept();

            DataInputStream dd1 = new DataInputStream(s1.getInputStream());

            Integer i1 = dd1.read();

            for (i = 0; i < i1; i++) {

                ServerSocket ss1;

                ss1 = new ServerSocket(9000 + i);

                Socket s = ss1.accept();

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

                String sss1 = dd.readUTF();

                System.out.println(sss1);

                System.out.println("Frame " + i + " received");

                Thread.sleep(4000);

                DataOutputStream d1 = new

                DataOutputStream(s.getOutputStream());

                d1.write(i);

                System.out.println("ACK sent for " + i);

            }

        } catch (Exception ex) {

            System.out.println("Error" + ex);

        }

    }

    public static void main(String[] args) {

```

```
Receiver r = new Receiver();r.start();}}
```

PAC 8:ROUTING

```
/*
```

Distance Vector Routing in this program

```
*/
```

```
#include<stdio.h>
```

```
struct node
```

```
{
```

```
    unsigned dist[20];
```

```
    unsigned from[20];
```

```
}rt[10];
```

```
int main()
```

```
{
```

```
    int costmat[20][20];
```

```
    int nodes,i,j,k,count=0;
```

```
    printf("\nEnter the number of nodes : ");
```

```
    scanf("%d",&nodes);//Enter the nodes
```

```
    printf("\nEnter the cost matrix :\n");
```

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

```
    {
```

```
        for(j=0;j<nodes;j++)
```

```
        {
```

```
            scanf("%d",&costmat[i][j]);
```

```
            costmat[i][i]=0;
```

```
            rt[i].dist[j]=costmat[i][j];//initialise the distance
```

equal to cost matrix

```
            rt[i].from[j]=j;
```

```
        }
```

```
    }
```

```
    do
```

```
    {
```

```
        count=0;
```

```
        for(i=0;i<nodes;i++)//We choose arbitrary vertex k and we
```

calculate the direct distance from the node i to k using the cost

matrix

```
        //and add the distance from k to node j
```

```
        for(j=0;j<nodes;j++)
```

```
        for(k=0;k<nodes;k++)
```

```
        if(rt[i].dist[j]>costmat[i][k]+rt[k].dist[j])
```

```

{//We calculate the minimum distance

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<nodes;i++)

{

printf("\n\n For router %d\n",i+1);

for(j=0;j<nodes;j++)

{

printf("\t\nnode %d via %d Distance %d

",j+1,rt[i].from[j]+1,rt[i].dist[j]);

}

}

printf("\n\n");

}

```