one way

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.util.Scanner;
import java.net.*;
public class MyClient {
public static void main(String[] args) {
try{
Socket s=new Socket("localhost",6666);// instesd of IP Address type "localhost"

DataOutputStream dout=new DataOutputStream(s.getOutputStream());
Scanner stri = new Scanner(System.in);
System.out.println("enter text");
```

String name = stri.nextLine();

dout.writeUTF(name);

dout.flush();

dout.close();

s.close();

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

}

}

two way

server-

```
import java.net.*;
import java.io.*;
class MyServer1{
public static void main(String args[])throws Exception{
ServerSocket ss=new ServerSocket(3333);
Socket s=ss.accept();
DataInputStream din=new DataInputStream(s.getInputStream());
DataOutputStream dout=new DataOutputStream(s.getOutputStream());
BufferedReader br=new BufferedReader(new
InputStreamReader(System.in));

System.out.println("Enter text");
String str="",str2="";
while(!str.equals("stop")){
str=din.readUTF();
System.out.println("client says: "+str);
str2=br.readLine();
dout.writeUTF(str2);
dout.flush();
}
din.close();
s.close();
ss.close();
}}
```

Client-

```
import java.net.*;
import java.io.*;
public class MyClient1 {
    public static void main(String args[])throws Exception{
        Socket s=new Socket("localhost",3333);
        DataInputStream din=new DataInputStream(s.getInputStream());
```

```
        DataOutputStream dout=new
DataOutputStream(s.getOutputStream());
        BufferedReader br=new BufferedReader(new
InputStreamReader(System.in));

           System.out.println("Enter text");
        String str="",str2="";
        while(!str.equals("stop")){
        str=br.readLine();
        dout.writeUTF(str);
        dout.flush();
        str2=din.readUTF();
        System.out.println("Server says: "+str2);
        }

        dout.close();
        s.close();
        }
        }
```

Manchester

Server-

```
import java.net.*;
import java.io.*;
class Server_Man{
public static void main(String args[])throws Exception{
ServerSocket ss=new ServerSocket(3333);
Socket s=ss.accept();
DataInputStream din=new DataInputStream(s.getInputStream());
DataOutputStream dout=new DataOutputStream(s.getOutputStream());
BufferedReader br=new BufferedReader(new
InputStreamReader(System.in));

System.out.println("Manchester Encoding");
String str="",str2 ="",dec = "";
int i;
while(!str.equals("stop")){
str=din.readUTF(); // here @1 process starts
System.out.println("Encoded code is"+ str);
int len = str.length();
for (i = 0;i< len;i += 2)
{
      String substr = str.substring(i,i+2);
      if(substr.equals("01"))
      {
            dec = dec + "0";
      }
      if(substr.equals("10"))
      {
            dec = dec + "1";
      }
}
```

```
System.out.println("Client's code decoded by server - " +""+ dec);
str2 = br.readLine();
int length = str2.length();
String enc = "";
for(i = 0;i< length;i++)
{
      if((str2.charAt(i)) == '0')
      {
            enc = enc + "01";
      }
      else{
            enc = enc +"10";
      }

}

dout.writeUTF(enc); // here @2 code goes to client side
}
din.close();
s.close();
ss.close();
}
}
```

Client-

```
import java.net.*;
import java.io.*;
public class Client_Man {
    public static void main(String args[])throws Exception{
        Socket s=new Socket("localhost",3333);
        DataInputStream din=new DataInputStream(s.getInputStream());
        DataOutputStream dout=new
DataOutputStream(s.getOutputStream());
        BufferedReader br=new BufferedReader(new
InputStreamReader(System.in));

           System.out.print("Client enter the string");
        String str="",str2="",dec = "";
        while(!str.equals("stop")){
        str2 = br.readLine(); //reads the string written by client
itself

      int length = str2.length(), i;
      String enc = "";
      for(i = 0;i< length;i++)
      {
            if((str2.charAt(i)) == '0')
            {
                  enc = enc + "01";
            }
            else{
                  enc = enc + "10";
            }
```

```
      }

dout.writeUTF(enc); // here code of client starts @1

str=din.readUTF(); //here @2 code further execute
System.out.println("Encoded code is" +""+ str);
int len = str.length();
for (i = 0;i< len;i += 2)
{
      String substr = str.substring(i,i+2);
      if(substr.equals("01"))
      {
            dec = dec + "0";
      }
      if(substr.equals("10"))
      {
            dec = dec + "1";
      }
}

System.out.println("Server's code decoded by client - "+""+ dec);

        }

        dout.close();
        s.close();
        }
        }
```

Differential

Server-

```
import java.net.*;
import java.io.*;
class Server_Diff{
public static void main(String args[])throws Exception{
ServerSocket ss=new ServerSocket(3333);
Socket s=ss.accept();
DataInputStream din=new DataInputStream(s.getInputStream());
DataOutputStream dout=new DataOutputStream(s.getOutputStream());
BufferedReader br=new BufferedReader(new
InputStreamReader(System.in));

System.out.println("Differential Manchester Encoding");
String str="",str2 ="",dec = "";
int i;
while(!str.equals("stop")){
str=din.readUTF(); // here @1 process starts
System.out.println("Encoded code is"+ str);
int len = str.length(),count=0;
for (i = 0;i< len;i += 2)
{
      String substr = str.substring(i,i+2);
      if(substr.equals("01"))
```

```
      {
            dec = dec + "0";
      }
      if(substr.equals("10"))
      {
            dec = dec + "1";
      }
}

System.out.println("Client's code decoded by server by Manchester
decoding technique - " +""+ dec);
str2 = br.readLine();
int length = str2.length();
String enc = "";
for(i = 0;i< length;i++)
{
      if(str2.charAt(i) == '0' && i == 0)
      {
            enc = enc + "01";

      }
      else if(str2.charAt(i) == '0' && i != 0)
      {
            enc = enc + enc.charAt(count-2)+enc.charAt(count - 1);

      }
      else if(str2.charAt(i) == '1' && i == 0)
      {
            enc = enc + "10";
      }
      else if(str2.charAt(i) == '1' && i != 0)
      {
                  enc = enc + enc.charAt(count-1)+ enc.charAt(count -
2);
      }
      count += 2;


}

dout.writeUTF(enc); // here @2 code goes to client side
}
din.close();
s.close();
ss.close();
}
}
```

Client-

```
import java.net.*;
import java.io.*;
public class Client_Diff {
    public static void main(String args[])throws Exception{
        Socket s=new Socket("localhost",3333);
```

```
        DataInputStream din=new DataInputStream(s.getInputStream());
        DataOutputStream dout=new
DataOutputStream(s.getOutputStream());
        BufferedReader br=new BufferedReader(new
InputStreamReader(System.in));

           System.out.print("Client enter the string");
        String str="",str2="",dec = "";
        while(!str.equals("stop")){
        str2 = br.readLine(); //reads the string written by client
itself

      int length = str2.length(), i,count = 0;
      String enc = "";
      for(i = 0;i< length;i++)
      {

            if(str2.charAt(i) == '0' && i == 0)
            {
                  enc = enc + "01";

            }
            else if(str2.charAt(i) == '0' && i != 0)
            {
                  enc = enc + enc.charAt(count-2)+enc.charAt(count -
1);

            }
            else if(str2.charAt(i) == '1' && i == 0)
            {
                  enc = enc + "10";
            }
            else if(str2.charAt(i) == '1' && i != 0)
            {
                        enc = enc + enc.charAt(count-1)+
enc.charAt(count - 2);
            }
            count += 2;



}

dout.writeUTF(enc); // here code of client starts @1

str=din.readUTF(); //here @2 code further execute
System.out.println("Encoded code is" +""+ str);
int len = str.length();
for (i = 0;i< len;i += 2)
{
      String substr = str.substring(i,i+2);
      if(substr.equals("01"))
      {
            dec = dec + "0";
      }
      if(substr.equals("10"))
```

```
	{
		dec = dec + "1";
	}
}

System.out.println("Server's code decoded by client by Manchester
decoding technique - "+""+ dec);

	}

	dout.close();
	s.close();
	}
	}
```

Bit stuffing-

```
import java.io.*;

public class BitStuff {

public static void main(String[] args) throws IOException {

 // TODO code application logic here

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

 BufferedReader br = new BufferedReader (new InputStreamReader(System.in));

 String data = br.readLine();

 String res = new String();

 int counter = 0;

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

 { System.out.println(data.charAt(i));

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

 { System.out.println("Enter only Binary values");

 return; }

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

 { counter++;

 res = res + data.charAt(i); }

 else

 { res = res + data.charAt(i);

 counter = 0; }
```

```
 if(counter == 5)
 { res = res + '0';
 counter = 0; }
 }
 System.out.println("The encrypted string is: " +res);
 } }
```

Character stuffing-

```
import java.io.*;
public class CharStuff {
 public static void main(String[] args) throws IOException {
 // TODO code application logic here
 System.out.print("Enter the input: ");
BufferedReader br = new BufferedReader (new InputStreamReader(System.in));
 String data = br.readLine();
 String res = "DLESTX";
 int i;
 for(i=0;i<data.length()-2;i++)
 {
if ((data.charAt(i) == 'd' || data.charAt(i) == 'D') && (data.charAt(i+1) == 'l' ||
data.charAt(i+1) == 'L') && (data.charAt(i+2) == 'e' || data.charAt(i+2) == 'E'))
 {
 res = res + "DLE";
 }
 res = res + data.charAt(i);
 }
 res = res + data.charAt(i) + data.charAt(i+1) + "DLEETX";
 System.out.println("Output is: " +res);
 }
}
```

Crc-

```
import java.util.*;
public class CRC {
 static Scanner sc = new Scanner(System.in);
 public static void main(String[] args) {
 // TODO code application logic here

 char[] msg = new char[20];
 char[] gen = new char[20];
 System.out.print("Enter the message string: ");
 String msgs = sc.nextLine();
 System.out.print("Enter the generator string: ");
 String gens = sc.nextLine();
 for(int i=0;i<msgs.length();i++)
 msg[i] = msgs.charAt(i);
 for(int i=0;i<gens.length();i++)
 gen[i]=gens.charAt(i);
 //Adding Zeroes
 for(int i=msgs.length();i<msgs.length() + gens.length()-1;i++)
 msg[i]='0';
 //Printing Appended Message
 System.out.print("Appended string is: ");
 for(int i=0;i<msgs.length() + gens.length()-1;i++)
 System.out.print(msg[i]);
 System.out.println();
 //Computing CRC
 for(int i=0;i<msgs.length();i++)
 {
 if(msg[i]!='1' && msg[i]!='0')
 {
 System.out.println("Entered message is wrong");
```

```
 return;
 }
if(msg[i]=='1')
 for(int j=0,k=i;j<gens.length();j++,k++)
 { msg[k]=xor(msg[k],gens.charAt(j)); }
 for(int l=0;l<msgs.length()+gens.length()-1;l++)
 System.out.print(msg[l]);
 System.out.println();
 }
 //Solution
 System.out.print("Checksummed message is: " +msgs);
 for(int i=msgs.length();i<msgs.length() + gens.length()-1;i++)
 System.out.print(msg[i]);
 System.out.println();
 }
 public static char xor(char x,char y)
 {
 if((x=='1'&&y=='1')||(x=='0'&&y=='0'))
 return '0';
 else
 return '1'; }
}
```

Stop and wait protocol-

**Sender.java**

```
import java.io.*;
import java.net.*;
public class Sender{
Socket sender;
ObjectOutputStream out;
ObjectInputStream in;
String packet,ack,str, msg;
int n,i=0,sequence=0;
Sender(){}
public void run(){
 try{
 BufferedReader br=new BufferedReader(new InputStreamReader(System.in));
```

```
System.out.println("Waiting for Connection....");
 sender = new Socket("localhost",2005);
sequence=0;
out=new ObjectOutputStream(sender.getOutputStream());
out.flush();
 in=new ObjectInputStream(sender.getInputStream());
 str=(String)in.readObject();
 System.out.println("reciver     > "+str);
System.out.println("Enter the data to send....");
packet=br.readLine();
n=packet.length();
do{
try{

if(i<n){
msg=String.valueOf(sequence);
msg=msg.concat(packet.substring(i,i+1));
}else if(i==n){
 msg="end";out.writeObject(msg);break;
}out.writeObject(msg);
sequence=(sequence==0)?1:0;
out.flush();
System.out.println("data sent>"+msg);
ack=(String)in.readObject();
System.out.println("waiting for ack.....\n\n");
if(ack.equals(String.valueOf(sequence))){
i++;
System.out.println("receiver   >  "+" packet recieved\n\n");
}else{
System.out.println("Time out resending data....\n\n");
sequence=(sequence==0)?1:0;
}}catch(Exception e){}
}while(i<n+1);
System.out.println("All data sent. exiting.");
}catch(Exception e){}
finally{
try{
in.close();
out.close();
sender.close();
}
catch(Exception e){}
}}
public static void main(String args[]){
Sender s=new Sender();
s.run();
}}
```

**Receiver.java**

```
import java.io.*;
import java.net.*;
public class Receiver{
ServerSocket reciever;
Socket connection=null;
ObjectOutputStream out;
ObjectInputStream in;
String packet,ack,data="";
int i=0,sequence=0;
Receiver(){}
public void run(){
try{
```

```
BufferedReader br=new BufferedReader(new InputStreamReader(System.in));
reciever = new ServerSocket(2005,10);
System.out.println("waiting for connection...");
connection=reciever.accept();
sequence=0;
System.out.println("Connection established   :");
out=new ObjectOutputStream(connection.getOutputStream());
out.flush();
in=new ObjectInputStream(connection.getInputStream());
out.writeObject("connected    .");

do{
try{
packet=(String)in.readObject();
if(Integer.valueOf(packet.substring(0,1))==sequence){
data+=packet.substring(1);
sequence=(sequence==0)?1:0;
System.out.println("\n\nreceiver         >"+packet);
}
else
{
System.out.println("\n\nreceiver         >"+packet +"   duplicate data");
}if(i<3){
out.writeObject(String.valueOf(sequence));i++;
}else{
out.writeObject(String.valueOf((sequence+1)%2));
i=0;
}}
catch(Exception e){}
}while(!packet.equals("end"));
System.out.println("Data recived="+data);
out.writeObject("connection ended    .");
}catch(Exception e){}
finally{
try{in.close();
out.close();
reciever.close();
}
catch(Exception e){}
}}

public static void main(String args[]){
Receiver s=new Receiver();
while(true){
s.run();
}
}
}
```

Dvr-

```
#include <stdio.h>
int min(int,int);
void table(int [][30],int,int);
int main()
{
    int num, cost[30][30], i,j,k;
    printf("enter number of routers\n");
```

```
    scanf("%d", &num);
    printf("enter delays between routers\n");
    for (i = 0; i < num; i++)
    {
        for (j = 0; j < num; j++)
        {
            scanf("%d", &cost[i][j]);
        }
    }

    printf("A's table initially\n");
    for (i = 0; i < num; i++)
    {
        printf("%d\n", cost[0][i]);
    }

    printf("B's table initially\n");
    for (i = 0; i < num; i++)
    {
        printf("%d\n", cost[1][i]);
    }

    printf("C's table initially\n");
    for (i = 0; i < num; i++)
    {
        printf("%d\n", cost[2][i]);
    }

    printf("D's table initially\n");
    for (i = 0; i < num; i++)
    {
        printf("%d\n", cost[3][i]);
    }

    for(i = 0;i< num;i++)
    {
        table(cost,i,num);
    }



}

int min(int a,int b)
{
    return(a>b)?b:a;
}
```

```
void table(int cost[30][30], int i,int num)
{
    int A1[num],A2[num];
    int j,k,l;
    for (l = 0; l < num; l++)
    {

        if (cost[i][l] != 0 && cost[i][l] != 99)
        {
            printf("neighbour of %d is %d\n", i+1,l + 1);
            for (j = 0; j < num; j++)
            {
                if (cost[l][j] != 99 && cost[i][j] != 99)
                {
                    A1[j] = cost[i][j] + min(cost[l][j],cost[i][j]);
                }
                else if(cost[i][j] == 99 && cost[l][j] != 99 )
                {

                    A1[j] = cost[i][l] + min(cost[l][j],cost[i][j]);
                }
                else if(cost[i][j] != 99 && cost[l][j] == 99)
                {
                    A1[j] = min(cost[l][j],cost[i][j]);
                }
            }

            for (k = 0; k < num; k++)
            {
                A2[k] = A1[k];
            }
            printf("new table is\n");
            for(k = 0;k<num;k++)
            {
                printf("%d\n",min(A1[k],A2[k]));
            }
        }

    }
}
```