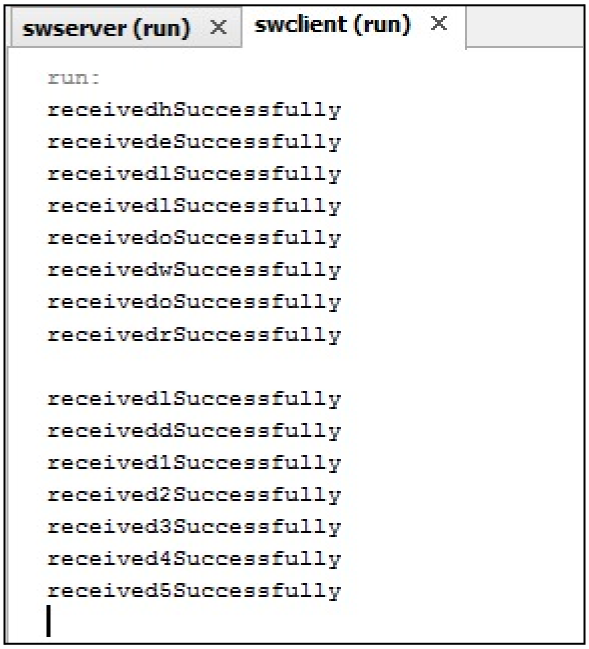
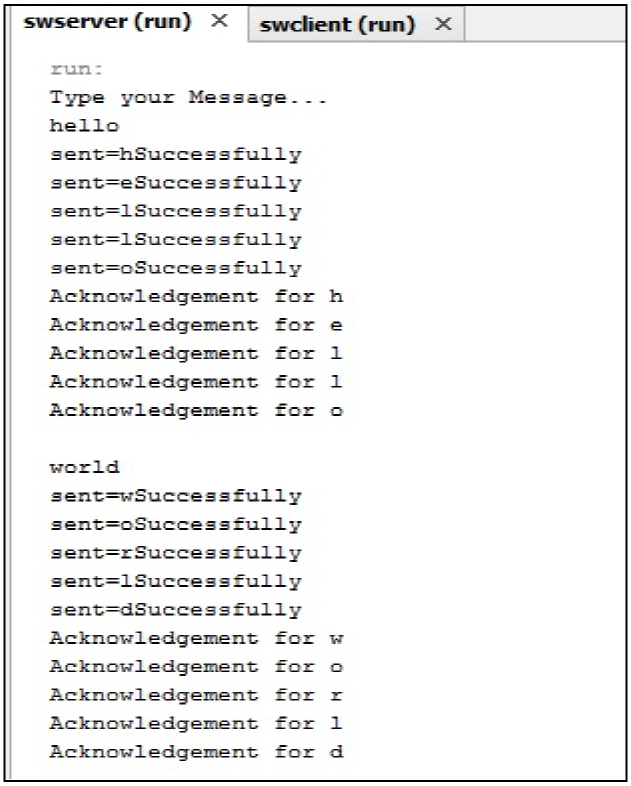
```
2b) //server
package swserver;
import java.io.*;
import java.net.*;
import java. util.Scanner;
public class Swserver {

    public static void main(String[] args)throws IOException{
        int SWS=8;  // Sender window size
        int LAR=0;//Sequence number of the last acknowledgement received
        int LFS=0;//Sequence number of the last frame sent
        ServerSocket ss=new ServerSocket(500);
        Socket s=ss.accept();
        System.out.println("Type your Message...");
        Scanner msgfromReceiver=new Scanner(s.getInputStream());
        PrintStream p=new PrintStream(s.getOutputStream());
        while(true){
            int i=0,j=0;
            Scanner msgtoReceiver=new Scanner(System.in);
            String t=msgtoReceiver.nextLine();
            if(t.trim().toLowerCase().equals("quit")){
                p.println(t);
               System.exit(0);
            }
        char c[]=new char[100];
        c=t.toCharArray();
        int sent=1;
        while(i<t.length()){
            while(i<t.length()&&i<SWS*sent){
                if (LFS-LAR<=SWS){
                    p.println(c[i]);
                    LFS++;
                    System.out.println("sent="+c[i++]+"Successfully");
                }
            }
            while(j<t.length()&&j<SWS*sent){
                String t1=msgfromReceiver.nextLine();
                System.out.println(t1);
                j++;
                LAR++;
            }
            sent++;
            System.out.println();
        }}}}
```

```
//client
package swclient;
import java.io.*;
import java.net.*;
import java. util.Scanner;

public class Swclient {

    public static void main(String[] args)throws IOException {
        int RWS=8;//Receiver window size
        int LAF=0;//Sequence number of largest acceptable frame
        int LFR=0; //Sequence number of last framereceived
        InetAddress obj=InetAddress.getLocalHost();
        Socket s=new Socket(obj,500);
        Scanner msgfromSender = new Scanner(s.getInputStream());
        PrintStream p=new PrintStream(s.getOutputStream());
        while(true){
            int i=0;
            while(i<RWS)
            {
                String t;
                if(LAF-LFR<=RWS){
                    t=msgfromSender.nextLine();
                    if(t.equals("quit"))System.exit(0);
                    System.out.println("received"+t+"Successfully");
                    LFR++;
                    p.println("Acknowledgement for "+t);
                    LAF++;
                }
                i++;
            }
            System.out.println();
        }
    }
}
```

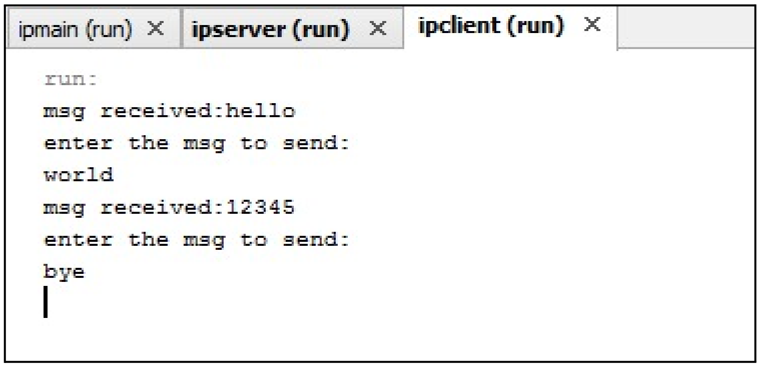


```
run:
receivedhSuccessfully
receivedeSuccessfully
receivedlSuccessfully
receivedlSuccessfully
receivedoSuccessfully
receivedwSuccessfully
receivedoSuccessfully
receivedrSuccessfully

receivedlSuccessfully
receiveddSuccessfully
received1Successfully
received2Successfully
received3Successfully
received4Successfully
received5Successfully
```

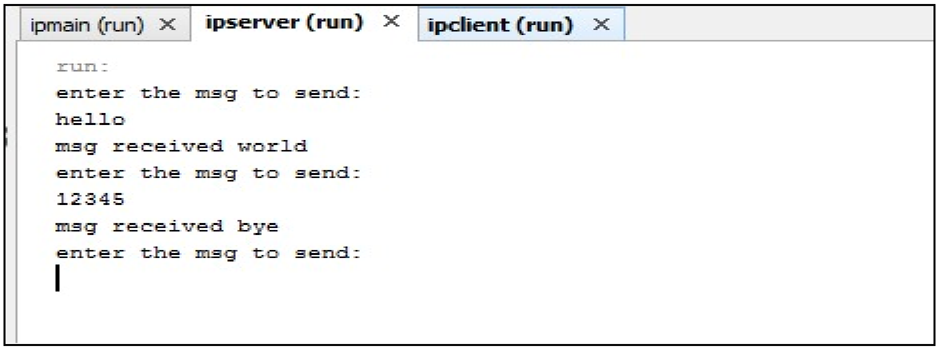


```
run:
Type your Message...
hello
sent=hSuccessfully
sent=eSuccessfully
sent=lSuccessfully
sent=lSuccessfully
sent=oSuccessfully
Acknowledgement for h
Acknowledgement for e
Acknowledgement for l
Acknowledgement for l
Acknowledgement for o

world
sent=wSuccessfully
sent=oSuccessfully
sent=rSuccessfully
sent=lSuccessfully
sent=dSuccessfully
Acknowledgement for w
Acknowledgement for o
Acknowledgement for r
Acknowledgement for l
Acknowledgement for d
```

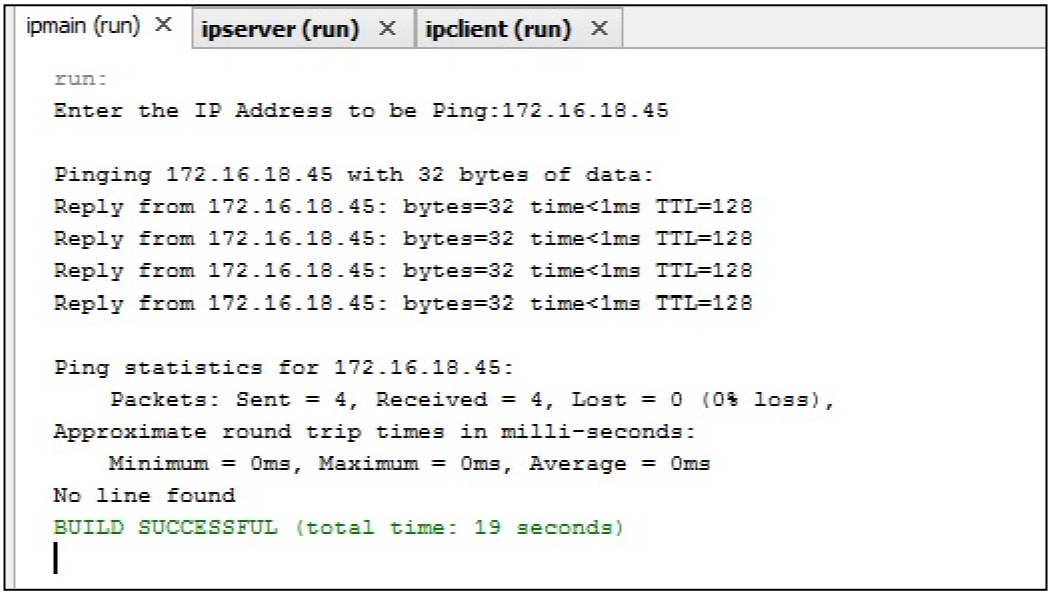
4b)



```
run:
msg received:hello
enter the msg to send:
world
msg received:12345
enter the msg to send:
bye
```



```
run:
enter the msg to send:
hello
msg received world
enter the msg to send:
12345
msg received bye
enter the msg to send:
```



```
run:
Enter the IP Address to be Ping:172.16.18.45

Pinging 172.16.18.45 with 32 bytes of data:
Reply from 172.16.18.45: bytes=32 time<1ms TTL=128
Reply from 172.16.18.45: bytes=32 time<1ms TTL=128
Reply from 172.16.18.45: bytes=32 time<1ms TTL=128
Reply from 172.16.18.45: bytes=32 time<1ms TTL=128

Ping statistics for 172.16.18.45:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms
No line found
BUILD SUCCESSFUL (total time: 19 seconds)
```

```
//main
package ipmain;
import java.io.*;
import java.net.*;
import java.util.Scanner;

public class Ipmain {
    public static void main(String[] args) {
        try{
            String str;
            System.out.print("Enter the IP Address to be Ping:");
            Scanner s1= new Scanner(System.in);
           String ip=s1.nextLine();
           Runtime H=Runtime.getRuntime();
           Process p=H.exec("ping "+ip);
           Scanner s2=new Scanner(p.getInputStream());
           while((str=s2.nextLine())!=null){
                System.out.println(""+str);
            }
        }
        catch (Exception e){
            System.out.println(e.getMessage());
    }
    }
}
```

```
//server
package ipserver;
import java.io.*;
import java.net.*;
import java.util.Scanner;
public class Ipserver {
    public static void main(String a[]) throws IOException{
        ServerSocket ss=new ServerSocket(8000);
        //Opens the socket
        Socket s=ss.accept();
        PrintStream dos=new PrintStream(s.getOutputStream());
        Scanner msgtoSend =new Scanner(System.in);
        Scanner msgfromClient=new Scanner(s.getInputStream());
        while(true){
            System.out.println("enter the msg to send: "); //Readstheinput
            String str=msgtoSend.nextLine();
            dos.println(str); //Checksforendofmessage
            if(str.equals("end")){
                //Closesthesocket
                ss.close();
                break;
            }
            String str1=msgfromClient.nextLine();
            System.out.println("msg received "+str1);
            if(str1.equals("end")){
                ss.close();
                break;
            }
        }
    }
}
```

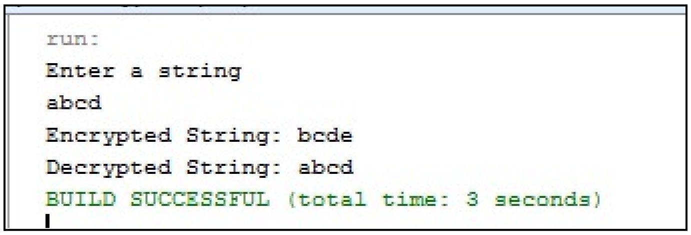
```
//client
package ipclient;
```

```
import java.io.*;
import java.net.*;
import java.lang.*;
import java.util.Scanner;
public class Ipclient {
    public static void main(String[] args) throws IOException {
      //Createsobjectforsocket
       Socket s=new Socket("localHost",8000);
       Scanner msgfromServer=new Scanner(s.getInputStream());
        Scanner msgtoSend=new Scanner(System.in);
        PrintStream dos=new PrintStream(s.getOutputStream());
        while(true){
            //Reads the input from the input deviceString
            String str = msgfromServer.nextLine();
            System.out.println("msg received:"+str);
            //Checksforendofmessage
            if(str.equals("end")){
                //Closesthesocket
                s.close();
                break;
            }
        System.out.println("enter the msg to send:");
        //Reads the message to send
        String str1=msgtoSend.nextLine();
        dos.println(str1);
        //Checksforendofmessage
        if(str1.equals("end")){
            //Closesthesocket
            s.close();
            break;
        }}}}
```

5)

```
package encryption;
import java.io.*;
import java.util.*;
public class Encryption {
    public static void main(String[] args) {
        Scanner sc =new Scanner(System.in);
        System.out.println("Enter a string");
        String temp = sc.next();
        String enc="";
        String dec="";
        for (char c: temp.toCharArray())
        {
```

```
      enc+=(char)(c+1);
    }
    System.out.println("Encrypted String: "+enc);
    for(char c:enc.toCharArray())
    {
      dec+=(char)(c-1);
    }
    System.out.println("Decrypted String: "+dec);
  }
}
```

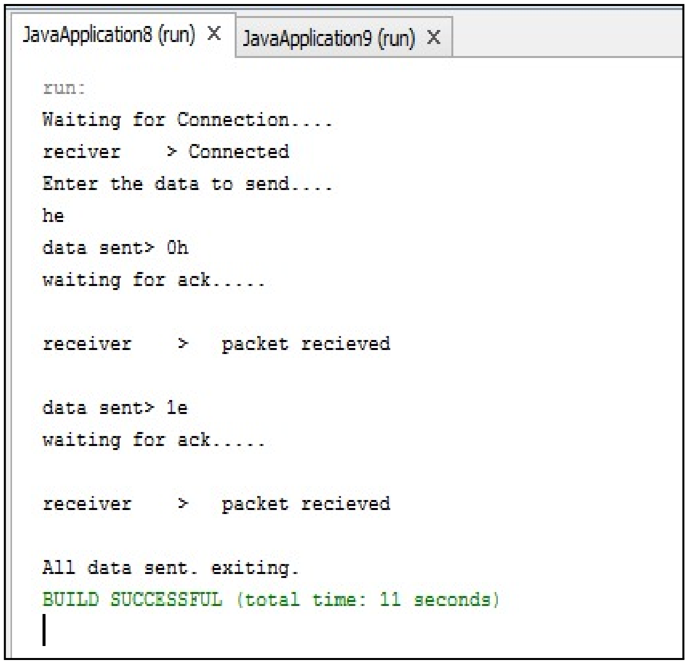


```
run:
Enter a string
abcd
Encrypted String: bcde
Decrypted String: abcd
BUILD SUCCESSFUL (total time: 3 seconds)
```

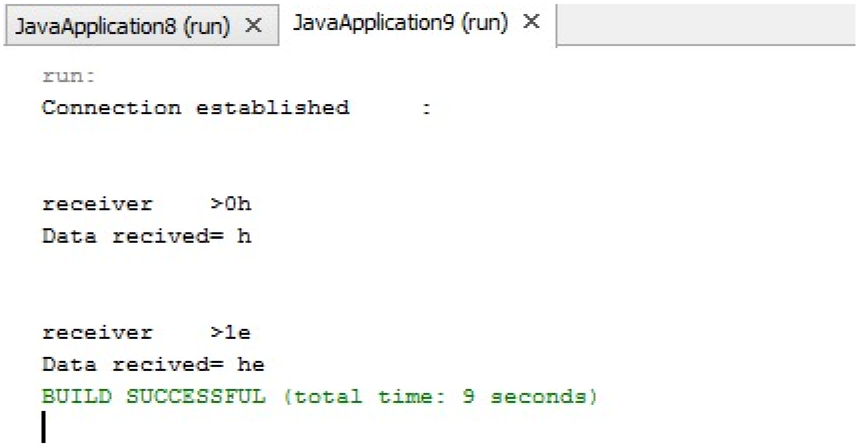
2a)

```
//server
package javaapplication8;
import java.io.*;
import java.net.*;
import java.util.Scanner;
public class JavaApplication8 {
  public static void main(String[] args)throws IOException {
  String packet,ack,str,msg="";
  int n,i=0,sequence=0;
   ServerSocket ss = new ServerSocket(2004);
  Socket s = ss.accept();
  Scanner br=new Scanner(System.in);
  Scanner msgfromReceiver = new Scanner(s.getInputStream());
  PrintStream dos = new PrintStream(s.getOutputStream());
  System.out.println("Waiting for Connection....");
  str= msgfromReceiver.nextLine();
  System.out.println("reciver    > "+str);
  System.out.println("Enter the data to send....");
  packet=br.nextLine();
  n=packet.length();
  while(i<n+1)
  {
      if(i<n){
        msg=String.valueOf(sequence);
        msg=msg.concat(packet.substring(i,i+1));
      }
```

```
            else if(i==n){
                msg="end";
                dos.println(msg);
                break;
            }
            dos.println(msg);
            /changing sequence number since data sent/
            sequence=(sequence==0)?1:0;
            System.out.println("data sent> "+msg);
            ack=msgfromReceiver.nextLine();
            System.out.println("waiting for ack.....\n");
            if(ack.equals(String.valueOf(sequence))){
                i++;
                System.out.println("receiver    >  "+" packet recieved\n");
            }
            else{      /* whenever ack lost or wrong ack we
                       change the sequence number*/
                System.out.println("Time out resending data....\n");
                sequence=(sequence==0)?1:0;
            }
    }
    System.out.println("All data sent. exiting.");
     s.close();
    }
}

//client
package javaapplication8;
import java.io.*;
import java.net.*;
import java.util.Scanner;
public class JavaApplication8 {
    public static void main(String[] args)throws IOException {
    String packet,ack,str,msg="";
    int n,i=0,sequence=0;
     ServerSocket ss = new ServerSocket(2004);
    Socket s = ss.accept();
    Scanner br=new Scanner(System.in);
    Scanner msgfromReceiver = new Scanner(s.getInputStream());
    PrintStream dos = new PrintStream(s.getOutputStream());
    System.out.println("Waiting for Connection....");
    str= msgfromReceiver.nextLine();
    System.out.println("reciver    > "+str);
    System.out.println("Enter the data to send....");
    packet=br.nextLine();
    n=packet.length();
    while(i<n+1)
    {
            if(i<n){
                msg=String.valueOf(sequence);
                msg=msg.concat(packet.substring(i,i+1));
            }
            else if(i==n){
                msg="end";
                dos.println(msg);
                break;
            }
            dos.println(msg);
            sequence=(sequence==0)?1:0;
            System.out.println("data sent> "+msg);
            ack=msgfromReceiver.nextLine();
            System.out.println("waiting for ack.....\n");
            if(ack.equals(String.valueOf(sequence))){
                i++;
                System.out.println("receiver    >  "+" packet recieved\n");
            }
            else{      /* whenever ack lost or wrong ack we
                        change the sequence number*/
                System.out.println("Time out resending data....\n");
                sequence=(sequence==0)?1:0;
            }
    }
    System.out.println("All data sent. exiting.");
     s.close();
    }
}
```



```
run:
Waiting for Connection....
reciver    > Connected
Enter the data to send....
he
data sent> 0h
waiting for ack.....

receiver    >   packet recieved

data sent> 1e
waiting for ack.....

receiver    >   packet recieved

All data sent. exiting.
BUILD SUCCESSFUL (total time: 11 seconds)
```



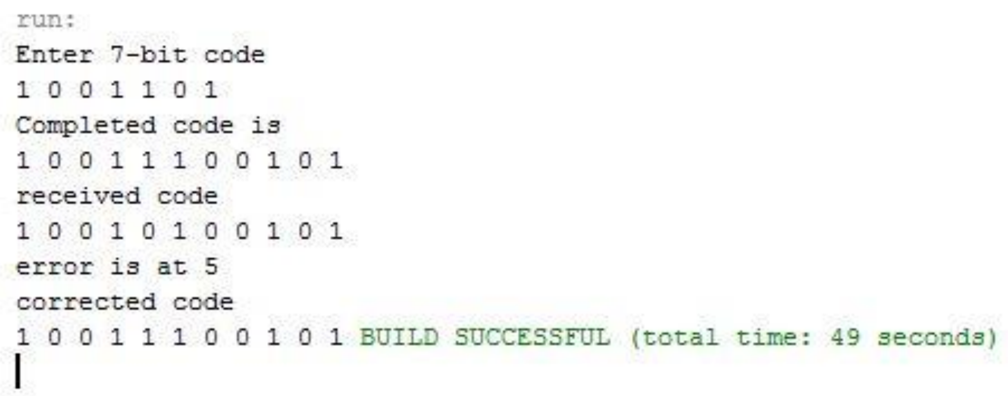
```
run:
Connection established      :


receiver    >0h
Data recived= h


receiver    >1e
Data recived= he
BUILD SUCCESSFUL (total time: 9 seconds)
```

1)

```
package hammingcodej;
import java.io.*;

import java.net.*;

import java.util.Scanner;

public class Hammingcodej {

   public static void main(String[] args){

       Scanner sc=new Scanner(System.in);

       int d[]=new int[7];

       System.out.println("Enter 7-bit code");

       for (int i=6;i>=0;i--)

       {

          d[i]=sc.nextInt();

       }

       int p[]=new int[4];

       p[0]=d[0]^d[1]^d[3]^d[4]^d[6];

       p[1]=d[0]^d[2]^d[3]^d[5]^d[6];

       p[2]=d[1]^d[2]^d[3];

       p[3]=d[4]^d[5]^d[6];

       int c[]=new int[11];

       c[0]=p[0];

       c[1]=p[1];

       c[2]=d[0];

       c[3]=p[2];

       c[4]=d[1];

       c[5]=d[2];

       c[6]=d[3];

       c[7]=p[3];

       c[8]=d[4];

       c[9]=d[5];

       c[10]=d[6];

       System.out.println("Completed code is");
```

```
for(int i=10;i>=0;i--)
{
   System.out.print(c[i]+" ");
}
System.out.println(" ");
int pr[]=new int[4];
int rd[]=new int[7];
System.out.println("received code");
int r[]=new int[11];
for(int i=10;i>=0;i--)
{
   r[i]=sc.nextInt();
}
pr[0]=r[0];
pr[1]=r[1];
rd[0]=r[2];
pr[2]=r[3];
rd[1]=r[4];
rd[2]=r[5];
rd[3]=r[6];
pr[3]=r[7];
rd[4]=r[8];
rd[5]=r[9];
rd[6]=r[10];
int s[]=new int[4];
s[0]=pr[0]^rd[0]^rd[1]^rd[3]^rd[4]^rd[6];
s[1]=pr[1]^rd[0]^rd[2]^rd[3]^rd[5]^rd[6];
s[2]=pr[2]^rd[1]^rd[2]^rd[3];
s[3]=pr[3]^rd[4]^rd[5]^rd[6];
int dec=(s[0]*1)+(s[1]*2)+(s[2]*4)+(s[3]*8);
if(dec==0)
```

```
        {
            System.out.println("no error\n");
        }
        else
        {
            System.out.println("error is at " + (11-dec+1));
        }
        if(r[dec-1]==0)
        {
            r[dec-1]=1;
        }
        else
        {
            r[dec-1]=0;
        }
        System.out.print("corrected code\n");
        for(int i=10;i>=0;i--)
        {
            System.out.print(r[i]+" ");
        }
    }
}
```

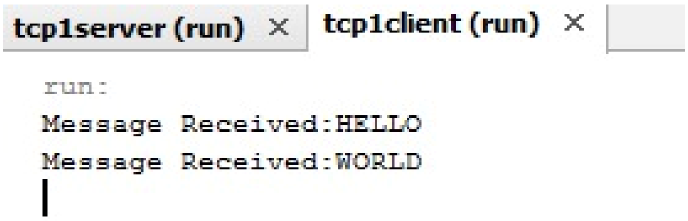


```
run:
Enter 7-bit code
1 0 0 1 1 0 1
Completed code is
1 0 0 1 1 1 0 0 1 0 1
received code
1 0 0 1 0 1 0 0 1 0 1
error is at 5
corrected code
1 0 0 1 1 1 0 0 1 0 1 BUILD SUCCESSFUL (total time: 49 seconds)
```

3)

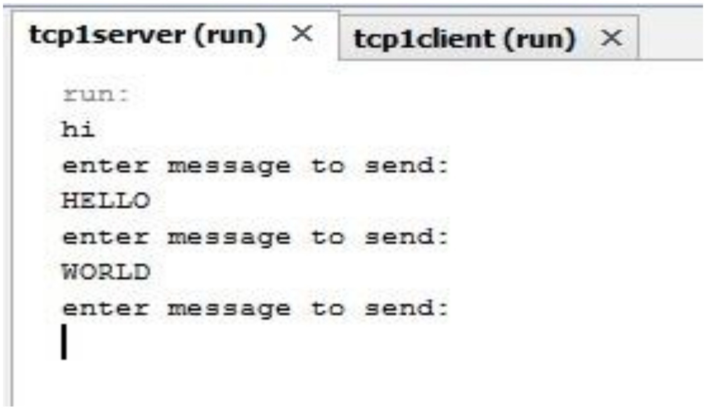
```
package tcp1server;
import java.io.*;

import java.net.*;

import java.util.Scanner;

public class Tcp1server {

   public static void main(String a[]) throws IOException

   {

      ServerSocket ss=new ServerSocket(8000);

      Socket s=ss.accept();

      Scanner in= new Scanner(System.in);

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

      System.out.println("hi");

      while(true)

      {

         System.out.println("enter message to send:");

         //Reads the input from the input device

         String str=in.nextLine();

         dos.println(str);

         //checksforendofmessage

         if(str.equals("end"))

         {

            //Closes the socket

            s.close();

            break;

         }

      }

   }

}
```

```
package tcp1client;
import java.io.*;
import java.net.*;
import java.util.Scanner;
public class Tcp1client {
   public static void main (String[] args) throws IOException{
      Socket s=new Socket("localHost",8000);
      //Used to get input from keyword
       Scanner in=new Scanner(s.getInputStream());
      while(true)
      {
      //Reads the input from keybroad
      String str=in.nextLine();
      System.out.println("Message Received:"+str);
      if(str.equals("end"))
      {
         //Close the socket
         s.close();
         break;
      }
      }
   }
}
```



```
run:
Message Received:HELLO
Message Received:WORLD
```

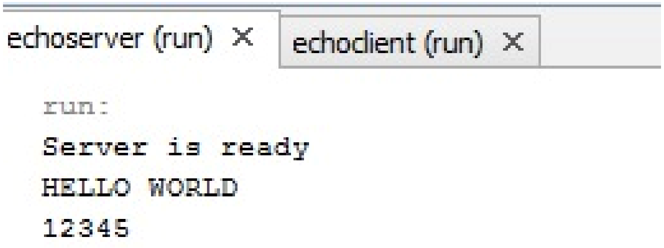
3)



```
run:
hi
enter message to send:
HELLO
enter message to send:
WORLD
enter message to send:
```

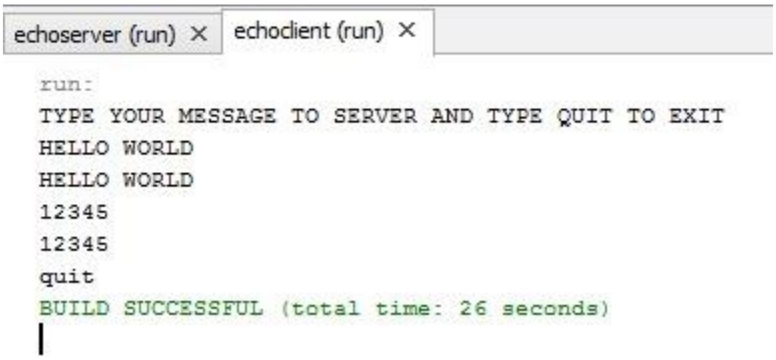
4a)

```
package echoserver;
import java.io.*;

import java.net.*;

import java.util.Scanner;

public class Echoserver{

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

  {

    ServerSocket ss=new ServerSocket(500);

    Socket s= ss.accept();

    System.out.println("Server is ready");

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

    Scanner msgfromClient = new Scanner(s.getInputStream());

    PrintStream p = new PrintStream(s.getOutputStream());

    while(true){

      String t=msgfromClient.nextLine();

      if(t==null)

        break;

      System.out.println(t);

      p.println(t);

    }

  }

}
```



```
run:
Server is ready
HELLO WORLD
12345
```

```
package echoclient;

import java.io.*;

import java.net.*;

import java.util.Scanner;

public class Echoclient{

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

    InetAddress obj=InetAddress.getLocalHost();

    Socket s=new Socket(obj,500);

    Scanner msgfromServer = new Scanner(s.getInputStream());

    PrintStream p=new PrintStream(s.getOutputStream());

    System.out.println("TYPE YOUR MESSAGE TO SERVER AND TYPE QUIT TO EXIT");

    while(true){

      String t=new Scanner(System.in).nextLine();

      if(t.equals("quit")){

        p.close();

        System.exit(0);

      }

      else{

        p.println(t);

        t=msgfromServer.nextLine();

        System.out.println(t);

      }

    }

  } }
```



```
run:
TYPE YOUR MESSAGE TO SERVER AND TYPE QUIT TO EXIT
HELLO WORLD
HELLO WORLD
12345
12345
quit
BUILD SUCCESSFUL (total time: 26 seconds)
```