2/19/2022

3152/B Hardik Togadiya

DCN

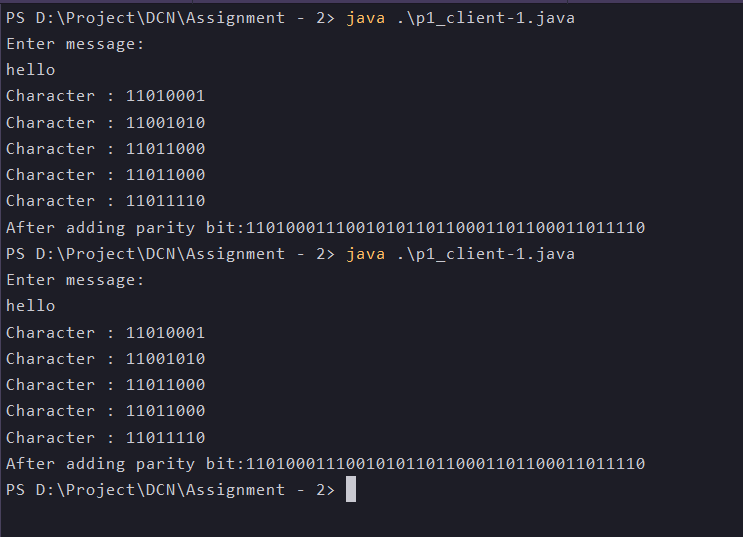
Assignment -2

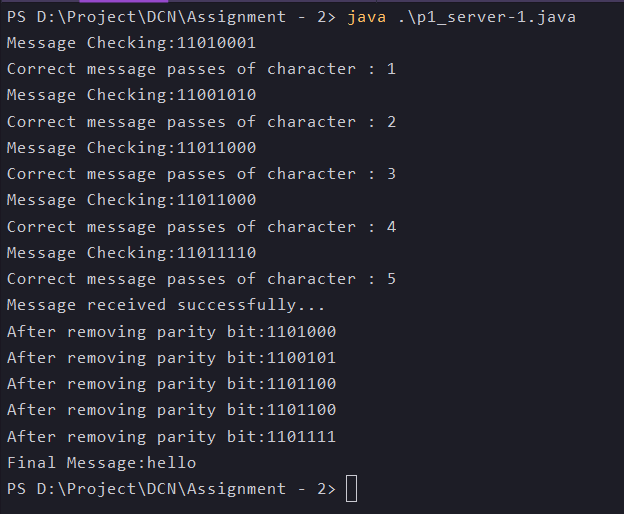
1. Write a java program to implement Single Bit - Even Parity using TCP.  
  
Client :

import java.io.\***;**import java.net.\***;**import java.util.Scanner**;**class p1\_client  
{  
 public static void main(String args[]) throws Exception  
 {  
 try  
 {  
 Socket socket=new Socket("localhost"**,**6363)**;** DataOutputStream outstream=new DataOutputStream(socket.getOutputStream())**;** String msg=""**,**op=""**,**finalmsg=""**;** int counter=0**,**i**,**j**;** Scanner sc=new Scanner(System.in)**;** System.out.println("Enter message:")**;** msg=sc.nextLine()**;** for(i=0**;**i<msg.length()**;**i++)  
 {  
 op=Integer.toBinaryString(msg.charAt(i))**;** counter=0**;** while(op.length()<7)  
 op='0'+op**;** for(j=0**;**j<op.length()**;**j++)  
 {  
 if(op.charAt(j)=='1')  
 counter++**;** }  
 if(counter%2==0)  
 op+='0'**;** else  
 op+='1'**;** finalmsg += op**;** System.out.println("Character : "+op)**;** }  
 System.out.println("After adding parity bit:"+finalmsg)**;** outstream.writeUTF(finalmsg)**;** outstream.flush()**;** outstream.close()**;** socket.close()**;** }  
 catch(Exception e)  
 {  
 System.out.println(e)**;** }  
 }  
}

Server:

import java.io.\***;**import java.net.\***;**class p1\_server  
{  
 public static void main(String args[])throws Exception  
 {  
 try  
 {  
 ServerSocket serversocket = new ServerSocket(6363)**;** Socket socket=serversocket.accept()**;** DataInputStream instream=new DataInputStream(socket.getInputStream())**;** DataOutputStream outstream=new DataOutputStream(socket.getOutputStream())**;** String finalmsg=instream.readUTF()**;** int i**,**j**,**k**,**counter=0**;** String f=""**,**op=""**;** for(i=0**;**i<finalmsg.length()**;**i+=8)  
 {  
 String decode=finalmsg.substring(i**,**i+8)**;** System.out.println("Message Checking:"+decode)**;** for(j=0**;**j<decode.length()**;**j++)  
 {  
 if(decode.charAt(j)=='1')  
 counter++**;** }  
 if(counter%2==0)  
 System.out.println("Correct message passes of character : "+((i/8)+1))**;** else  
 {  
 System.out.println("Please send the message again..")**;** System.exit(0)**;** }  
 }  
 System.out.println("Message received successfully...")**;** for(k=0**;**k<finalmsg.length()**;**k+=8)  
 {  
 op=finalmsg.substring(k**,**k+7)**;** System.out.println("After removing parity bit:"+op)**;** f+=(char)Byte.parseByte(op**,**2)**;** }  
 System.out.println("Final Message:"+f)**;** instream.close()**;** socket.close()**;** serversocket.close()**;** }  
 catch(Exception e)  
 {  
 System.out.println(e)**;** }  
 }  
}

Client:   


Server:   


2. Write a java program to implement Single Bit - Odd Parity using UDP.

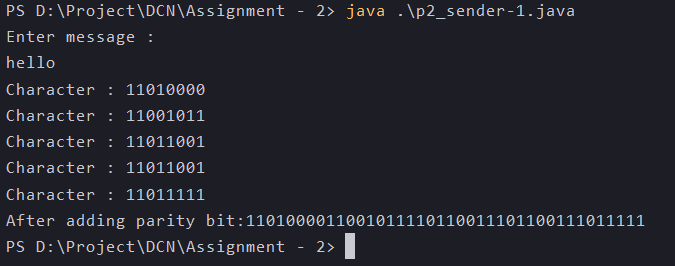
Sender:

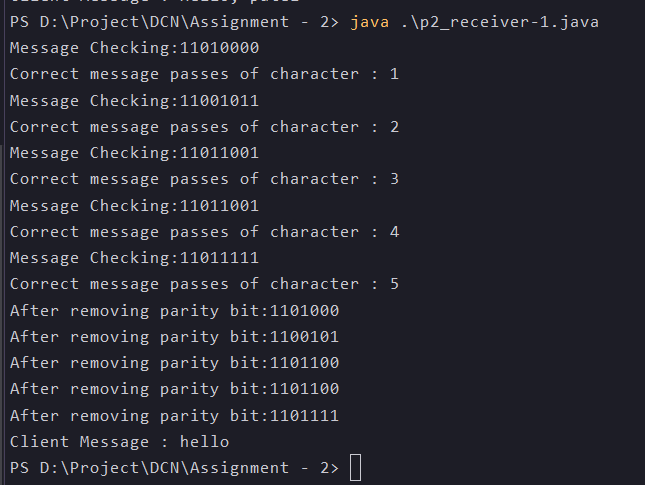
import java.util.Scanner**;**import java.io.\***;**import java.net.\***;**class p2\_sender  
{  
 public static void main(String args[]) throws Exception  
 {  
 try  
 {  
 DatagramSocket ds=new DatagramSocket()**;** InetAddress ip = InetAddress.getByName("localhost")**;** int i**,**j**,**ans**,**counter=0**;** String msg=""**,**op=""**,**finalmsg=""**;** Scanner sc=new Scanner(System.in)**;** System.out.println("Enter message :")**;** msg=sc.nextLine()**;** for(i=0**;**i<msg.length()**;**i++)  
 {  
 op=Integer.toBinaryString(msg.charAt(i))**;** counter=0**;** while(op.length()<7)  
 op='0'+op**;** for(j=0**;**j<op.length()**;**j++)  
 {  
 if(op.charAt(j)=='1')  
 counter++**;** }  
 if(counter%2!=0)  
 op+='0'**;** else  
 op+='1'**;** finalmsg += op**;** System.out.println("Character : "+op)**;** }  
 System.out.println("After adding parity bit:"+finalmsg)**;** DatagramPacket dp=new DatagramPacket(finalmsg.getBytes()**,**finalmsg.length()**,**ip**,**6565)**;** ds.send(dp)**;** ds.close()**;** }  
 catch(Exception e)  
 {  
 System.out.println(e)**;** }  
 }  
}

Reciver:

import java.io.\***;**import java.net.\***;**class p2\_receiver  
{  
 public static void main(String args[]) throws Exception  
 {  
 try  
 {  
 DatagramSocket ds=new DatagramSocket(6565)**;** byte[] buf = new byte[1024]**;** //client data mokle e aama store thasse  
 DatagramPacket dp=new DatagramPacket(buf**,**1024)**;**//Or we can write buf.length instead of 1024  
 ds.receive(dp)**;** int i**,**j**,**counter=0**;** String msg=new String(dp.getData()**,**0**,**dp.getLength())**;** for(i=0**;**i<msg.length()**;**i+=8)  
 {  
 String decode=msg.substring(i**,**i+8)**;** System.out.println("Message Checking:"+decode)**;** for(j=0**;**j<decode.length()**;**j++)  
 {  
 if(decode.charAt(j)=='1')  
 counter++**;** }  
 if(counter%2!=0)  
 System.out.println("Correct message passes of character : "+((i/8)+1))**;** else   
 System.exit(0)**;** counter=0**;** }  
   
 int k**;** String f=""**,**op=""**;** for(k=0**;**k<msg.length()**;**k+=8)  
 {  
 op=msg.substring(k**,**k+7)**;** System.out.println("After removing parity bit:"+op)**;** f+=(char)Byte.parseByte(op**,**2)**;** }  
   
 System.out.println("Client Message : " + f)**;** ds.close()**;** }  
 catch(Exception e)  
 {  
 System.out.println(e)**;** }  
 }  
}

Sender:



Reciver:  


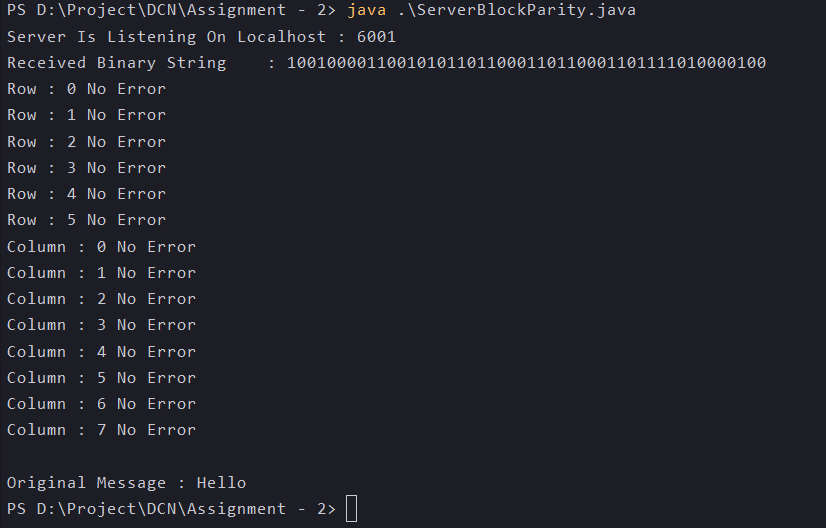
3. Write a java program to implement Block Parity using TCP.

Server:

import java.io.\***;**import java.net.\***;**import java.util.\***;**class ServerBlockParity{  
 public static void main(String args[])throws Exception  
 {  
 try  
 {  
 Scanner sc=new Scanner(System.in)**;** ServerSocket serversocket = new ServerSocket(6001)**;** System.out.println("Server Is Listening On Localhost : 6001")**;** Socket socket = serversocket.accept()**;** DataInputStream istream = new DataInputStream(socket.getInputStream())**;** DataOutputStream ostream = new DataOutputStream(socket.getOutputStream())**;** String rmsg = new String()**;** String omsg = new String()**;** rmsg=istream.readUTF()**;** char cpmsg[][] = new char[rmsg.length()/8][8]**;** int rowerror =0 **,** colerror = 0**;** System.out.println("Received Binary String : "+rmsg)**;** for(int i=0**;**i<rmsg.length()**;**i+=8)  
 {  
 String rtemp = rmsg.substring(i**,**i+8)**;** char p = evenParity(rtemp)**;** if(p == '0')  
 {  
 System.out.println("Row : " + i/8 + " No Error")**;** }  
 else  
 {  
 rowerror++**;** System.out.println("Row : " + i/8 + " Error Detected")**;** }  
   
 cpmsg[i/8] = rtemp.toCharArray()**;** rtemp = rtemp.substring(0**,**rtemp.length()-1)**;** omsg = omsg + (char)Byte.parseByte(rtemp**,**2)**;** }  
 for(int j=0**;**j<8**;**j++)  
 {  
 String ctemp = new String()**;** for(int i=0**;**i<rmsg.length()/8**;**i++)  
 {  
 ctemp = ctemp + cpmsg[i][j]**;** }  
 char p = evenParity(ctemp)**;** if(p == '0')  
 {  
 System.out.println("Column : " + j + " No Error")**;** }  
 else  
 {  
 colerror++**;** System.out.println("Column : " + j + " Error Detected")**;** }  
 }  
 if(rowerror == 0 && colerror == 0)  
 {  
 omsg = omsg.substring(0**,**omsg.length()-1)**;** System.out.println("\nOriginal Message : "+omsg)**;** }  
   
   
   
   
 istream.close()**;** socket.close()**;** serversocket.close()**;** }  
 catch(Exception e)  
 {  
 System.out.println(e)**;** }  
 }  
 public static char evenParity(String msg)  
 {  
 int c=0**;** for(int j=0**;**j<msg.length()**;**j++)  
 {  
 if(msg.charAt(j)=='1')  
 {  
 c++**;** }  
 }  
 //System.out.println("Counter : "+c);  
 if(c%2==0)  
 {  
 return '0'**;** }  
 else  
 {  
 return '1'**;** }  
 }  
  
}

Client:

import java.io.\***;**import java.net.\***;**import java.util.\***;**class ClientBlockParity{  
 public static void main(String args[])throws Exception  
 {  
 try  
 {  
 Scanner sc = new Scanner(System.in)**;** Socket socket = new Socket("localhost"**,**6001)**;** DataInputStream istream = new DataInputStream(socket.getInputStream())**;** DataOutputStream ostream = new DataOutputStream(socket.getOutputStream())**;** System.out.print("Enter Any String : ")**;** String msg=sc.nextLine()**;** char gpmsg[][] = new char[msg.length()+1][8]**;** String sendmsg=new String()**;** for(int i=0**;**i<msg.length()**;**i++)  
 {  
 int ch=msg.charAt(i)**;** // Store Each Char ASCII Value  
 String temp=Integer.toBinaryString(ch)**;** // Convert ASCII Value to Binary Byte  
 while(temp.length()<7)  
 {  
 temp = '0'+ temp**;** }  
 temp = temp + evenParity(temp)**;** // Send String and Check Even Bit Parity or Odd Bit Parity   
   
   
   
 sendmsg = sendmsg + temp**;** for(int j=0**;**j<temp.length()**;**j++)  
 {  
 gpmsg[i][j] = temp.charAt(j)**;** }  
   
 System.out.println("Char : "+msg.charAt(i) + " ASCII :- "+ch+" Binary String With Parity Bit - "+temp)**;** }  
   
 //System.out.println();  
 for(int i = 0**;**i<8**;**i++)  
 {  
 String temp = new String()**;** for(int j=0**;**j<msg.length()**;**j++)  
 {  
 temp = temp + gpmsg[j][i]**;** }  
 gpmsg[msg.length()][i]= evenParity(temp)**;** sendmsg = sendmsg + gpmsg[msg.length()][i]**;** System.out.println("Column : "+i+" Parity Bit : "+gpmsg[msg.length()][i])**;** }  
 System.out.println("Send Binary String : "+sendmsg)**;** ostream.writeUTF(sendmsg)**;** ostream.flush()**;** ostream.close()**;** socket.close()**;** }  
 catch(Exception e)  
 {  
 System.out.println(e)**;** }  
   
 }  
 public static char evenParity(String msg)  
 {  
 int c=0**;** for(int j=0**;**j<msg.length()**;**j++)  
 {  
 if(msg.charAt(j)=='1')  
 c++**;** }  
 if(c%2==0)  
 {  
 return '0'**;** }  
 else  
 {  
 return '1'**;** }  
 }  
  
}

Server:  


Client:  
