DCCN PROGRAMS

1) //TCPServer.java

```
import java.io.*;

import java.net.*;


class TCPServer

{

  public static void main(String argv[]) throws Exception

    {

      String fromclient;

      String toclient;


      ServerSocket Server = new ServerSocket (7000);          //starts the server at port
number 7000


      System.out.println ("TCPServer Waiting for client on port 7000");


      while(true)          //running infinity loop for all the clients

      {

        Socket connected = Server.accept();    //accept the connection  from the client

        System.out.println( " THE CLIENT"+" "+

        connected.getInetAddress() +":"+connected.getPort()+" IS CONNECTED ");//printing
ip address and port number



        BufferedReader inFromUser =
        new BufferedReader(new InputStreamReader(System.in));  //creating the object for
reading from keyboard


        PrintWriter outToClient =
          new PrintWriter(
            connected.getOutputStream(),true); //creating an object for writing to the socket
and make it autoflush



        BufferedReader inFromClient =
          new BufferedReader(new InputStreamReader (connected.getInputStream()));
//reading bytes from socket and converting to character



        while ( true )     // conversation between client and server
        {

         System.out.println("SEND(Type Q or q to Quit):");
         toclient = inFromUser.readLine();       //reading from keyboard

         if ( toclient.equals ("q") || toclient.equals("Q") )
         {
                 outToClient.println(toclient);
                 connected.close();
                 break;
         }
         else
         {
                 outToClient.println(toclient);  //writting to the socket
```

```
               }

             fromclient = inFromClient.readLine();//reading from socket

               if ( fromclient.equals("q") || fromclient.equals("Q") )
               {
                   connected.close();
                   break;
               }

                      else
                      {
                       System.out.println( "RECIEVED:" + fromclient );
                       //printing to the monitor
                      }

                        }

             }
           }
         }
```

2) import java.io.*;

import java.net.*;

```
class TCPClient
{
 public static void main(String argv[]) throws Exception
 {
  String FromServer;
  String ToServer;

  Socket clientSocket = new Socket("localhost", 7000);

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

  PrintWriter outToServer = new PrintWriter(
    clientSocket.getOutputStream(),true);

  BufferedReader inFromServer = new BufferedReader(new InputStreamReader(
    clientSocket.getInputStream()));

  while (true)
  {

   FromServer = inFromServer.readLine();

   if ( FromServer.equals("q") || FromServer.equals("Q"))
      {
      clientSocket.close();
      break;
      }
   else

       {
        System.out.println("RECIEVED:" + FromServer);
        System.out.println("SEND(Type Q or q to Quit):");

        ToServer = inFromUser.readLine();

        if (ToServer.equals("Q") || ToServer.equals("q"))
        {
        outToServer.println (ToServer) ;
        clientSocket.close();
           break;
        }

       else
       {
          outToServer.println(ToServer);
       }
     }
   }
  }
}
```

```
3) // Online Java Compiler
// Use this editor to write, compile and run your Java code online
import java.util.Scanner;
```

```java
public class Main {

    // Function to calculate parity bit
    public static int calculateParity(String data, boolean evenParity) {
        int count = 0;
        for (int i = 0; i < data.length(); i++) {
            if (data.charAt(i) == '1') {
                count++;
            }
        }

        if (evenParity)
            return (count % 2 == 0) ? 0 : 1;  // Even parity
        else
            return (count % 2 == 0) ? 1 : 0;  // Odd parity
    }

    // Function to check for errors in received data
    public static boolean checkError(String receivedData, boolean evenParity) {
        int count = 0;
        for (int i = 0; i < receivedData.length(); i++) {
            if (receivedData.charAt(i) == '1') {
                count++;
            }
        }

        if (evenParity)
            return (count % 2 == 0);  // True if even -> No error
        else
            return (count % 2 != 0);  // True if odd -> No error
    }

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        System.out.print("Enter binary data: ");
        String data = sc.next();

        System.out.print("Choose parity type (1 for Even, 0 for Odd): ");
        boolean evenParity = sc.nextInt() == 1;

        int parityBit = calculateParity(data, evenParity);
        String transmittedData = data + parityBit;

        System.out.println("\nGenerated Parity Bit: " + parityBit);
        System.out.println("Data sent with parity bit: " + transmittedData);

        System.out.print("\nEnter received data (with parity bit): ");
        String receivedData = sc.next();

        if (checkError(receivedData, evenParity))
            System.out.println("✅ No error detected in transmission.");
        else
            System.out.println("❌ Error detected in received data!");
    }
}
```

4) import java.util.Scanner;

```
class Main {

   // Function to add two binary strings and handle carry/overflow
   public static String addBinary(String a, String b, int wordSize) {
      int sum = Integer.parseInt(a, 2) + Integer.parseInt(b, 2);
      String result = Integer.toBinaryString(sum);

      // Handle overflow (carry)
      while (result.length() > wordSize) {
         int overflow = Integer.parseInt(result.substring(0, result.length() - wordSize), 2);
         int remaining = Integer.parseInt(result.substring(result.length() - wordSize), 2);
         result = Integer.toBinaryString(overflow + remaining);
      }

      // Pad with zeros to maintain word size
      while (result.length() < wordSize) {
         result = "0" + result;
      }
      return result;
   }

   // Function to compute 1’s complement of a binary string
   public static String onesComplement(String s) {
      StringBuilder complement = new StringBuilder();
      for (int i = 0; i < s.length(); i++) {
         complement.append(s.charAt(i) == '0' ? '1' : '0');
      }
```

```
        return complement.toString();
    }

    // Function to calculate checksum (sender side)
    public static String calculateChecksum(String[] dataWords, int wordSize) {
        String sum = dataWords[0];
        for (int i = 1; i < dataWords.length; i++) {
            sum = addBinary(sum, dataWords[i], wordSize);
        }
        return onesComplement(sum);
    }

    // Function to verify checksum (receiver side)
    public static boolean verifyChecksum(String[] dataWords, String receivedChecksum, int
wordSize) {
        String sum = dataWords[0];
        for (int i = 1; i < dataWords.length; i++) {
            sum = addBinary(sum, dataWords[i], wordSize);
        }

        // Add checksum
        sum = addBinary(sum, receivedChecksum, wordSize);

        // If all bits in 1’s complement are zero => no error
        String complement = onesComplement(sum);
        return !complement.contains("1");
    }

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
```

```
System.out.print("Enter word size (e.g., 8): ");
int wordSize = sc.nextInt();

System.out.print("Enter number of data words: ");
int n = sc.nextInt();

String[] dataWords = new String[n];
System.out.println("Enter the data words (binary):");
for (int i = 0; i < n; i++) {
    dataWords[i] = sc.next();
}

// Sender side
String checksum = calculateChecksum(dataWords, wordSize);
System.out.println("\nCalculated Checksum: " + checksum);

System.out.println("\n--- Transmission ---");
System.out.println("Data sent with checksum: ");
for (String w : dataWords) System.out.println(w);
System.out.println("Checksum: " + checksum);

// Receiver side
System.out.println("\nEnter received data words (binary):");
String[] receivedWords = new String[n];
for (int i = 0; i < n; i++) {
    receivedWords[i] = sc.next();
}
```

```java
        System.out.print("Enter received checksum: ");
        String receivedChecksum = sc.next();

        if (verifyChecksum(receivedWords, receivedChecksum, wordSize))
            System.out.println("\n✅ No error detected in received data.");
        else
            System.out.println("\n❌ Error detected in received data!");
    }
}
```