TCPServer.java

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

public class TCPServer {
   public static void main(String[] args) throws Exception {
      ServerSocket server = new ServerSocket(5000);
      Socket socket = server.accept();
      BufferedReader inClient = new BufferedReader(new InputStreamReader(socket.getInputStream()));
      PrintWriter outClient = new PrintWriter(socket.getOutputStream(), true);
      BufferedReader input = new BufferedReader(new InputStreamReader(System.in));
      String msg;

      while (true) {
         msg = input.readLine();
         outClient.println(msg);
         if (msg.equalsIgnoreCase("q")) break;

         msg = inClient.readLine();
         if (msg.equalsIgnoreCase("q")) break;
         System.out.println("Client: " + msg);
      }
      socket.close();
      server.close();
   }
}
```

TCPClient.java

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

public class TCPClient {
   public static void main(String[] args) throws Exception {
      Socket socket = new Socket("localhost", 5000);
      BufferedReader inServer = new BufferedReader(new InputStreamReader(socket.getInputStream()));
      PrintWriter outServer = new PrintWriter(socket.getOutputStream(), true);
      BufferedReader input = new BufferedReader(new InputStreamReader(System.in));
      String msg;

      while (true) {
         msg = inServer.readLine();
         if (msg.equalsIgnoreCase("q")) break;
         System.out.println("Server: " + msg);

         msg = input.readLine();
         outServer.println(msg);
         if (msg.equalsIgnoreCase("q")) break;
      }
      socket.close();
   }
}
```

EchoServer.java

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

public class EchoServer {
   public static void main(String[] args) throws Exception {
      ServerSocket ss = new ServerSocket(9999);
      Socket s = ss.accept();
      BufferedReader in = new BufferedReader(new InputStreamReader(s.getInputStream()));
      PrintWriter out = new PrintWriter(s.getOutputStream(), true);
      String msg;
      out.println("Echo Server Ready. Type 'bye' to exit.");
      while (!(msg = in.readLine()).equalsIgnoreCase("bye")) {
         System.out.println("Client: " + msg);
         out.println("Echo: " + msg);
      }
      s.close();
      ss.close();
   }
}
```

EchoClient.java

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

public class EchoClient {
   public static void main(String[] args) throws Exception {
      Socket s = new Socket("localhost", 9999);
      BufferedReader in = new BufferedReader(new InputStreamReader(s.getInputStream()));
      PrintWriter out = new PrintWriter(s.getOutputStream(), true);
      BufferedReader user = new BufferedReader(new InputStreamReader(System.in));
      String msg;
      System.out.println(in.readLine());
      while (true) {
         msg = user.readLine();
         out.println(msg);
         if (msg.equalsIgnoreCase("bye")) break;
         System.out.println(in.readLine());
      }
      s.close();
   }
}
```

Ping.java

```
import java.io.*;

public class Ping {
   public static void main(String[] args) throws Exception {
      BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
      System.out.print("IP: ");
      String ip = br.readLine();

      Process p = Runtime.getRuntime().exec("ping " + ip);
      BufferedReader r = new BufferedReader(new InputStreamReader(p.getInputStream()));
      String line;
      while ((line = r.readLine()) != null)
         System.out.println(line);
   }
}
```

UDPServer.java

```
import java.net.*;
public class UDPServer {
    public static void main(String[] args) throws Exception {
        DatagramSocket s = new DatagramSocket(790);
        byte[] b = new byte[1024];
        DatagramPacket p = new DatagramPacket(b, b.length);
        System.out.println("Server started");
        while (true) {
            s.receive(p);
            String msg = new String(b, 0, p.getLength());
            if (msg.equalsIgnoreCase("stop")) break;
            System.out.println("Client: " + msg);

            String reply = new java.util.Scanner(System.in).nextLine();
            s.send(new DatagramPacket(reply.getBytes(), reply.length(),
p.getAddress(), p.getPort()));
        }
        s.close();
    }
}
```

UDPClient.java

```
import java.net.*;

public class UDPClient {
    public static void main(String[] args) throws Exception {
        DatagramSocket s = new DatagramSocket();
        InetAddress ip = InetAddress.getLocalHost();
        java.util.Scanner sc = new java.util.Scanner(System.in);
        byte[] b = new byte[1024];
        System.out.println("Client started");
        while (true) {
            String msg = sc.nextLine();
            s.send(new DatagramPacket(msg.getBytes(), msg.length(), ip, 790));
            if (msg.equalsIgnoreCase("stop")) break;

            DatagramPacket p = new DatagramPacket(b, b.length);
            s.receive(p);
            System.out.println("Server: " + new String(p.getData(), 0,
p.getLength()));
        }
        s.close();
    }
}
```

CRC.java

```
import java.io.*;
public class CRC {
    static int[] divide(int[] div, int[] divisor, int[] rem) {
        int cur = 0;
        while (true) {
            for (int i = 0; i < divisor.length; i++)
                rem[cur + i] ^= divisor[i];
            while (cur < rem.length && rem[cur] == 0) cur++;
            if ((rem.length - cur) < divisor.length) break;
        }
        return rem;
    }
    public static void main(String[] args) throws IOException {
        BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
        System.out.print("Data bits: ");
        int d = Integer.parseInt(br.readLine());
        int[] data = new int[d];
        System.out.println("Enter data:");
        for (int i = 0; i < d; i++) data[i] = Integer.parseInt(br.readLine());

        System.out.print("Divisor bits: ");
        int divLen = Integer.parseInt(br.readLine());
        int[] divisor = new int[divLen];
        System.out.println("Enter divisor:");
        for (int i = 0; i < divLen; i++) divisor[i] = Integer.parseInt(br.readLine());

        int total = d + divLen - 1;
        int[] div = new int[total], rem = new int[total], crc = new int[total];
        for (int i = 0; i < d; i++) div[i] = data[i];
        for (int i = 0; i < total; i++) rem[i] = div[i];
        rem = divide(div, divisor, rem);
        for (int i = 0; i < total; i++) crc[i] = div[i] ^ rem[i];
        System.out.print("CRC Code: ");
        for (int i : crc) System.out.print(i);
        System.out.println();
        System.out.println("Enter received code:");
        for (int i = 0; i < total; i++) crc[i] = Integer.parseInt(br.readLine());
        for (int i = 0; i < total; i++) rem[i] = crc[i];
        rem = divide(crc, divisor, rem);
        boolean error = false;
        for (int i : rem) if (i != 0) error = true;
        System.out.println(error ? "Error" : "No Error");
    }
}
```