SERVER

```
import java.io.*;
import java.net.*;
public class StopWaitServer {
   ServerSocket receiver;
   Socket connection = null;
   ObjectOutputStream out;
   ObjectInputStream in;
   String packet, ack, data = "";
   int i = 0, sequence = 0;
   StopWaitServer() {
   }
   public void run() {
      try {
         BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
         receiver = new ServerSocket(2005, 10);
         System.out.println("Waiting for connection...");
         connection = receiver.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 received: " + data);
        out.writeObject("connection ended.");
    } catch (Exception e) {
    } finally {
        try {
            in.close();
            out.close();
            receiver.close();
        } catch (Exception e) {
        }
    }
  }

public static void main(String args[]) {
    StopWaitServer receiver = new StopWaitServer();
    while (true) {
        receiver.run();
```

```
        }
    }
}
```

RECEIVER

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

public class StopWaitReceiver {
    Socket sender;
    ObjectOutputStream out;
    ObjectInputStream in;
    String packet, ack, str, msg;
    int n, i = 0, sequence = 0;
    int timeout = 5000; // 5 seconds timeout (adjust as needed)

    StopWaitReceiver() {
    }

    public void run() {
        try {
            BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
            System.out.println("Waiting for Connection....");
            sender = new Socket("localhost", 2005);
            sender.setSoTimeout(timeout); // Set socket timeout
            sequence = 0;
            out = new ObjectOutputStream(sender.getOutputStream());
            out.flush();
            in = new ObjectInputStream(sender.getInputStream());
            str = (String) in.readObject();
            System.out.println("receiver > " + 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 received\n\n");
      } else {
         System.out.println("Time out resending data....\n\n");
         sequence = (sequence == 0) ? 1 : 0;
      }
   } catch (SocketTimeoutException ste) {
      System.out.println("Connection timed out. Exiting.");
      break;
   }
   catch (SocketException se) {
```

```
                System.out.println("Connection reset by peer. Exiting.");
                // Handle any cleanup or logging if needed
                System.exit(1); // Exit the program
            }
            catch (Exception e) {
                e.printStackTrace();
                break;
            }
        } while (i < n + 1);
        System.out.println("Closing connection.");
    } catch (Exception e) {
        e.printStackTrace();
    } finally {
        try {
            in.close();
            out.close();
            sender.close();
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
  }
  public static void main(String args[]) {
    StopWaitReceiver s = new StopWaitReceiver();
    s.run();
  }
}
```

OUTPUT


```
C:\Users\DHANANJAY\OneDrive\Desktop\RCOEM\V sem\CN>java StopWaitREceiver.java
Waiting for Connection....
receiver > connected.
Enter the data to send....
DC
data sent>0D
waiting for ack.....


receiver >  packet received


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


receiver >  packet received


Closing connection.

C:\Users\DHANANJAY\OneDrive\Desktop\RCOEM\V sem\CN>
```

```
C:\Users\DHANANJAY\OneDrive\Desktop\RCOEM\V sem\CN>java StopWaitREceiver.java
Waiting for Connection....
receiver > connected.
Enter the data to send....
dsgv
Connection reset by peer. Exiting.

C:\Users\DHANANJAY\OneDrive\Desktop\RCOEM\V sem\CN>
```