Networking and Programming Assignments (Java & C) - Continued

4. Java Program for Stop-and-Wait ARQ

```
Server (StopAndWaitServer.java):
--------------------------------
import java.io.*;
import java.net.*;

public class StopAndWaitServer {
    public static void main(String[] args) throws IOException {
        ServerSocket server = new ServerSocket(6000);
        Socket client = server.accept();
                                BufferedReader    in    =    new    BufferedReader(new
InputStreamReader(client.getInputStream()));
        PrintWriter out = new PrintWriter(client.getOutputStream(), true);
        String data;
        int expected = 0;
        while ((data = in.readLine()) != null) {
            int frame = Integer.parseInt(data);
            if (frame == expected) {
                System.out.println("Received frame: " + frame);
                out.println("ACK " + expected);
                expected = (expected + 1) % 2;
            } else {
                System.out.println("Duplicate frame: " + frame);
                out.println("ACK " + ((expected + 1) % 2));
            }
        }
        client.close();
        server.close();
    }
}

Client (StopAndWaitClient.java):
--------------------------------
import java.io.*;
import java.net.*;

public class StopAndWaitClient {
    public static void main(String[] args) throws IOException {
        Socket socket = new Socket("localhost", 6000);
                                BufferedReader    in    =    new    BufferedReader(new
InputStreamReader(socket.getInputStream()));
        PrintWriter out = new PrintWriter(socket.getOutputStream(), true);
        for (int i = 0; i < 5; i++) {
            System.out.println("Sending frame: " + i % 2);
            out.println(i % 2);
            String ack = in.readLine();
            System.out.println("Received: " + ack);
        }
        socket.close();
```

```
    }
}
```

5. Java Program for Subnetting

```
SubnettingCalculator.java:
--------------------------
import java.util.Scanner;

public class SubnettingCalculator {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter IP address (e.g. 192.168.1.0): ");
        String ip = scanner.nextLine();
        System.out.print("Enter number of subnets: ");
        int numSubnets = scanner.nextInt();
        String[] parts = ip.split("\.");
        int[] ipParts = new int[4];
        for (int i = 0; i < 4; i++) ipParts[i] = Integer.parseInt(parts[i]);
        int bits = (int) Math.ceil(Math.log(numSubnets) / Math.log(2));
        int newPrefix = 24 + bits;
        int numHosts = (int) Math.pow(2, 32 - newPrefix);
        int subnetInc = numHosts;
        System.out.println("Subnet mask: /" + newPrefix);
        System.out.println("Subnet addresses:");
        for (int i = 0; i < numSubnets; i++) {
            int subnet = i * subnetInc;
            int fourthOctet = subnet % 256;
            int thirdOctet = (subnet / 256) % 256;
                System.out.println(ipParts[0] + "." + ipParts[1] + "." + (ipParts[2] +
thirdOctet) + "." + fourthOctet);
        }
    }
}
```

6. Java Echo Client-Server Program

```
Server (EchoServer.java):
-------------------------
import java.io.*;
import java.net.*;

public class EchoServer {
    public static void main(String[] args) throws IOException {
        ServerSocket serverSocket = new ServerSocket(7000);
        Socket socket = serverSocket.accept();
                                BufferedReader    in    =    new    BufferedReader(new
InputStreamReader(socket.getInputStream()));
        PrintWriter out = new PrintWriter(socket.getOutputStream(), true);
        String input;
        while ((input = in.readLine()) != null) {
            System.out.println("Client: " + input);
            out.println("Echo: " + input);
```

```
        }
        socket.close();
        serverSocket.close();
    }
}

Client (EchoClient.java):
-------------------------
import java.io.*;
import java.net.*;

public class EchoClient {
    public static void main(String[] args) throws IOException {
        Socket socket = new Socket("localhost", 7000);
        BufferedReader userInput = new BufferedReader(new InputStreamReader(System.in));
                                  BufferedReader    in    =    new    BufferedReader(new
InputStreamReader(socket.getInputStream()));
        PrintWriter out = new PrintWriter(socket.getOutputStream(), true);
        String line;
        while (!(line = userInput.readLine()).equalsIgnoreCase("exit")) {
            out.println(line);
            System.out.println(in.readLine());
        }
        socket.close();
    }
}
```

7. C Program for CRC Generation

```
crc.c:
------
#include <stdio.h>
#include <string.h>

void xor(char *dividend, char *divisor, char *result) {
    for (int i = 1; i < strlen(divisor); i++)
        result[i - 1] = (dividend[i] == divisor[i]) ? '0' : '1';
    result[strlen(divisor) - 1] = '\0';
}

void crc(char *data, char *key) {
    int dataLen = strlen(data);
    int keyLen = strlen(key);
    char appendedData[100], temp[100], remainder[100];
    strcpy(appendedData, data);
    for (int i = 0; i < keyLen - 1; i++)
        appendedData[dataLen + i] = '0';
    appendedData[dataLen + keyLen - 1] = '\0';
    strncpy(temp, appendedData, keyLen);
    for (int i = 0; i < dataLen; i++) {
        if (temp[0] == '1')
            xor(temp, key, remainder);
        else
```

```
            xor(temp, "0000", remainder);
        temp[keyLen - 1] = appendedData[i + keyLen];
        temp[keyLen] = '\0';
        strcpy(temp, remainder);
    }
    strcpy(remainder, temp);
    printf("CRC code: %s\n", remainder);
    printf("Transmitted data: %s%s\n", data, remainder);
}

int main() {
    char data[50], key[50];
    printf("Enter binary data: ");
    scanf("%s", data);
    printf("Enter generator polynomial: ");
    scanf("%s", key);
    crc(data, key);
    return 0;
}
```