Name: Raj Koyani

Reg no:-21MIS1017

Subject:- CN LAB-(2)

(a) Client is sending a message to the server. The server encodes the message and returns to the client.(Encoding is done by replacing the character by the ASCII value of the remainder using the formula(ASCII(chr) mod (n th prime))n-value is sent by the client)Write the program to implement the above.

Server code:--

```
import java.util.Scanner;

public class serverg {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter the message: ");
        String message = scanner.nextLine();
        System.out.print("Enter the value of n: ");
        int nValue = scanner.nextInt();

        String encodedMessage = encodeMessage(message, nValue);
        System.out.println("Encoded message: " + encodedMessage);
    }

    public static boolean isPrime(int n) {
        if (n < 2) {
            return false;
        }
        for (int i = 2; i <= Math.sqrt(n); i++) {
            if (n % i == 0) {
                return false;
            }
        }
        return true;
    }

    public static int getNthPrime(int n) {
        int count = 0;
        int num = 2;
        while (true) {
```

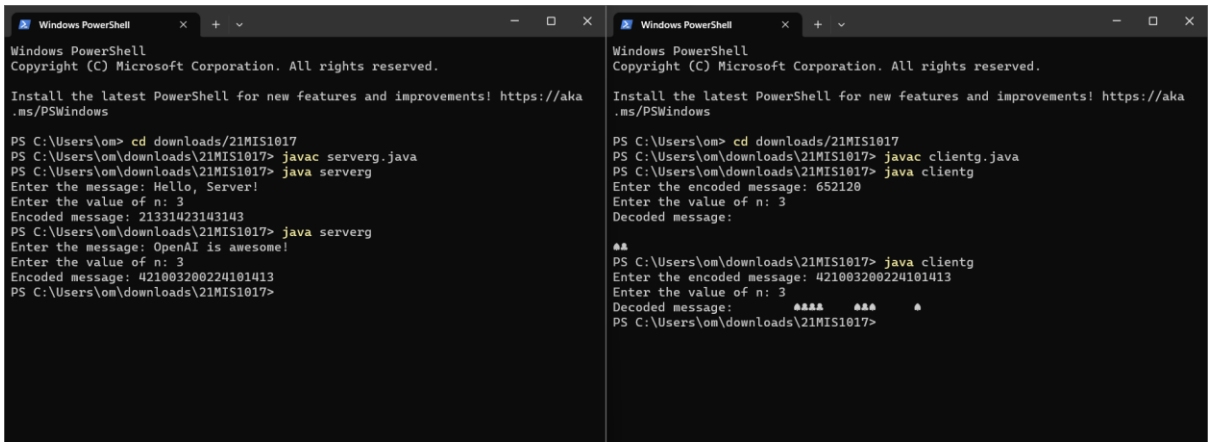
```
            if (isPrime(num)) {
                count++;
                if (count == n) {
                    return num;
                }
            }
            num++;
        }
    }

    public static String encodeMessage(String message, int n) {
        int prime = getNthPrime(n);
        StringBuilder encodedMessage = new StringBuilder();
        for (char c : message.toCharArray()) {
            int asciiValue = (int) c;
            int encodedChar = asciiValue % prime;
            encodedMessage.append(encodedChar);
        }
        return encodedMessage.toString();
    }
}
```

Client:-

```
import java.util.Scanner;

public class clientg {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter the encoded message: ");
        String encodedMessage = scanner.nextLine();
        System.out.print("Enter the value of n: ");
        int nValue = scanner.nextInt();

        String decodedMessage = decodeMessage(encodedMessage, nValue);
        System.out.println("Decoded message: " + decodedMessage);
    }

    public static boolean isPrime(int n) {
        if (n < 2) {
            return false;
        }
        for (int i = 2; i <= Math.sqrt(n); i++) {
            if (n % i == 0) {
                return false;
            }
        }
```

```
        return true;
    }

    public static int getNthPrime(int n) {
        int count = 0;
        int num = 2;
        while (true) {
            if (isPrime(num)) {
                count++;
                if (count == n) {
                    return num;
                }
            }
            num++;
        }
    }

    public static String decodeMessage(String encodedMessage, int n) {
        int prime = getNthPrime(n);
        StringBuilder decodedMessage = new StringBuilder();
        for (int i = 0; i < encodedMessage.length(); i++) {
            int encodedChar =
Character.getNumericValue(encodedMessage.charAt(i));
            int decodedChar = encodedChar + prime;
            decodedMessage.append((char) decodedChar);
        }
        return decodedMessage.toString();
    }
}
```

Output:-



(b) Implement a TCP/IP socket-based ATM system. Server – to maintain the customer details (Name , Cardno, Pin, Balance). Client – when a customer wants to withdraw an amount, validate his login with pin and balance.

Server:-

```
import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;
import java.io.PrintWriter;
import java.net.ServerSocket;
import java.net.Socket;
import java.util.HashMap;
import java.util.Map;

public class serveratm {
    private static Map<String, Account> accounts;

    public static void main(String[] args) {
        int port = 1234;
        accounts = initializeAccounts();

        try {
            // Create a server socket
            ServerSocket serverSocket = new ServerSocket(port);
            System.out.println("Server listening on port " + port);

            while (true) {
                Socket clientSocket = serverSocket.accept();
                System.out.println("Client connected: " +
clientSocket.getInetAddress());

                Thread thread = new Thread(() ->
handleClientRequest(clientSocket));
                thread.start();
            }
        } catch (IOException e) {
            e.printStackTrace();
        }
    }

    private static void handleClientRequest(Socket clientSocket) {
        try {
            // Get the input and output streams of the client socket
            BufferedReader in = new BufferedReader(new
InputStreamReader(clientSocket.getInputStream()));
```

```
            PrintWriter out = new PrintWriter(clientSocket.getOutputStream(),
true);
            String cardNumber = in.readLine();
            String pin = in.readLine();

            if (validateLogin(cardNumber, pin)) {

                Account account = accounts.get(cardNumber);
                String customerName = account.getCustomerName();
                double balance = account.getBalance();

                out.println("Welcome, " + customerName);
                out.println("Card Number: " + cardNumber);
                out.println("Balance: " + balance);
            } else {

                out.println("Invalid login credentials");
            }

            clientSocket.close();
        } catch (IOException e) {
            e.printStackTrace();
        }
    }

    private static boolean validateLogin(String cardNumber, String pin) {
        if (accounts.containsKey(cardNumber)) {
            Account account = accounts.get(cardNumber);
            return account.getPin().equals(pin);
        }
        return false;
    }

    private static Map<String, Account> initializeAccounts() {
        Map<String, Account> accounts = new HashMap<>();
        accounts.put("895338223", new Account("Raj", "2003", 50000.0));
        accounts.put("562315545", new Account("shena", "6391", 800.0));
        return accounts;
    }

    private static class Account {
        private String customerName;
        private String pin;
        private double balance;

        public Account(String customerName, String pin, double balance) {
            this.customerName = customerName;
            this.pin = pin;
```

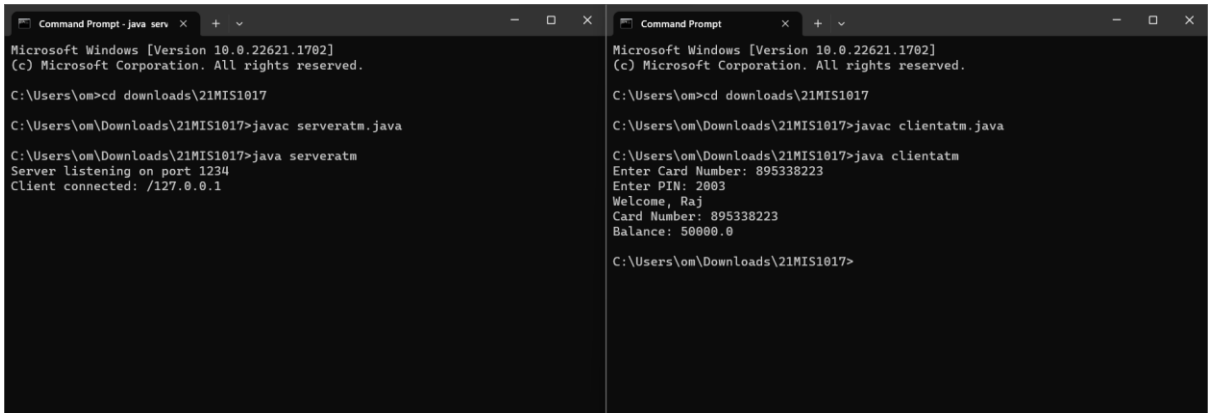
```
            this.balance = balance;
        }

        public String getCustomerName() {
            return customerName;
        }

        public String getPin() {
            return pin;
        }

        public double getBalance() {
            return balance;
        }
    }
}
```

Client:-

```
import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;
import java.io.PrintWriter;
import java.net.Socket;

public class clientatm {
    public static void main(String[] args) {
        String serverAddress = "localhost";
        int serverPort = 1234;

        try {

            Socket socket = new Socket(serverAddress, serverPort);

            BufferedReader in = new BufferedReader(new
InputStreamReader(socket.getInputStream()));
            PrintWriter out = new PrintWriter(socket.getOutputStream(), true);

            BufferedReader userInput = new BufferedReader(new
InputStreamReader(System.in));
            System.out.print("Enter Card Number: ");
            String cardNumber = userInput.readLine();
            System.out.print("Enter PIN: ");
            String pin = userInput.readLine();

            out.println(cardNumber);
```

```
            out.println(pin);

            String response;
            while ((response = in.readLine()) != null) {
                System.out.println(response);
            }

            socket.close();
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
}
```

Output:-



(c) In an IPV4 packet the value of header length is 1000 in binary. Write a code to find how many bytes of options are being carried by this packet.

Code:-

```
public class IPv4Packet21MIS1017 {
    public static void main(String[] args) {
        String headerLengthBinary = "1000"; // Binary header length value

        // Convert binary to decimal
        int headerLengthDecimal = Integer.parseInt(headerLengthBinary, 2);

        // Calculate the length of the IPv4 header in bytes
        int headerLengthBytes = headerLengthDecimal * 4;

        // Subtract minimum header length to get options length in bytes
```

```
        int optionsLengthBytes = headerLengthBytes - 20;

        System.out.println("Options Length: " + optionsLengthBytes + "
bytes");
    }
}
```

Output:-