//1. Server.java

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

public class Server {
   public static void main(String[] args) {
      final int PORT = 1234;

      try {
         // Create server socket
         ServerSocket serverSocket = new ServerSocket(PORT);
         System.out.println("Server waiting for client on port " + PORT);

         // Accept client connection
         Socket socket = serverSocket.accept();
         System.out.println("Client connected");

         // Create input stream to receive messages from client
         BufferedReader in = new BufferedReader(new
InputStreamReader(socket.getInputStream()));

         // Create output stream to send messages to client
         PrintWriter out = new PrintWriter(socket.getOutputStream(), true);

         // Read messages from client and send back a response
         String message;
         while ((message = in.readLine()) != null) {
            System.out.println("Client: " + message);
            out.println("Server received: " + message);
         }

         // Close streams and socket
         in.close();
         out.close();
         socket.close();
         serverSocket.close();
      } catch (IOException e) {
         e.printStackTrace();
      }
   }
}
```

//Client.java

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

public class Client {
   public static void main(String[] args) {
      final String SERVER_ADDRESS = "localhost";
      final int PORT = 1234;

      try {
         // Create socket to connect to server
         Socket socket = new Socket(SERVER_ADDRESS, PORT);

         // Create input stream to receive messages from server
         BufferedReader in = new BufferedReader(new
InputStreamReader(socket.getInputStream()));

         // Create output stream to send messages to server
         PrintWriter out = new PrintWriter(socket.getOutputStream(), true);

         // Create input stream to read user input
         BufferedReader userInput = new BufferedReader(new
InputStreamReader(System.in));

         String message;
         while (true) {
            System.out.print("Enter message to send to server: ");
            message = userInput.readLine();
            out.println(message);

            // Exit loop if user types "exit"
            if (message.equalsIgnoreCase("exit")) {
               break;
            }

            // Read response from server
            String response = in.readLine();
            System.out.println("Server: " + response);
         }

         // Close streams and socket
         in.close();
         out.close();
```

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

Output:

Server Site:

Server waiting for client on port 1234

Client Site:

Enter message to send to server**: This is client for testing message**

Server: Server received: This is client for testing message

Enter message to send to server:

// 2. Program to write and execute a program OnetimePad using Java.

```
import java.io.*;
import java.math.*;
import java.lang.*;
import java.util.*;
class Onetimepad
{
public static void main(String arg[])throws IOException
{
BufferedReader br=new BufferedReader(new InputStreamReader(System.in));
Random rg=new Random();
String temp=" ";
System.out.print("enter plain text:");
temp=br.readLine();
temp=temp.toUpperCase();
int length=temp.length();
char plaintext[]=new char[length];
char key[]=new char[length];
char ciphertext[]=new char[length];
char deciphertext[]=new char[length];
for(int i=0;i<length;i++)
{
plaintext[i]=temp.charAt(i);
key[i]=((char)(65+rg.nextInt(26)));
ciphertext[i]=(char)((int)plaintext[i]^(int)key[i]);
deciphertext[i]=((char)((int)ciphertext[i]^(int)key[i]));
}
System.out.print("entered plaintext:");
for(int i=0;i<length;i++)
System.out.print(" "+plaintext[i]);
System.out.println(" ");
System.out.print("enter key:");
for(int i=0;i<length;i++)
System.out.print(" "+key[i]);
System.out.println(" ");
System.out.print("ciphertext:");
for(int i=0;i<length;i++)
System.out.print("  "+ciphertext[i]);
System.out.println(" ");
System.out.print("deciphertext:");
for(int i=0;i<length;i++)
System.out.print(" "+deciphertext[i]);  }}
```

Output:

C:\dccn>javac Onetimepad.java

C:\dccn>java Onetimepad

enter plain text:kmmips

entered plaintext: K M M I P S

enter key: N L D G W C

ciphertext: ♣ ☺ ♫ ►

deciphertext: K M M I P S

//3.Program to write and Execute a program Find My PC IP Address using Java.

```
import java.net.InetAddress;
import java.net.UnknownHostException;

public class ipadd {
   public static void main(String[] args) {
      try {
         // Get the local host information
         InetAddress localHost = InetAddress.getLocalHost();

         // Get the hostname
         String hostname = localHost.getHostName();

         // Get the IP address
         String ipAddress = localHost.getHostAddress();

         // Print the hostname and IP address
         System.out.println("Hostname: " + hostname);
         System.out.println("IP Address: " + ipAddress);
      } catch (UnknownHostException e) {
         System.err.println("Unable to determine the hostname and IP address.");
         e.printStackTrace();
      }
   }
}
```

Output:

```
Hostname: DESKTOP-61JE93H
IP Address: 192.168.1.4
```

//4. A Java program to implement the Leaky Bucket algorithm to smooth out bursty traffic and control network congestion.

Source Code:

```
import java.util.Random;
import java.util.Scanner;

public class LeakyBucket {
   public static void main(String[] args) throws InterruptedException {
      Scanner sc = new Scanner(scScannerSystem.in);
      Random rand = new Random();

      System.out.print("Enter Bucket Capacity (BufferSize): ");
      int bucketSize = sc.nextInt();

      System.out.print("Enter Constant Output Leak Rate: ");
      int outputRate = sc.nextInt();

      System.out.print("Enter total time steps to simulate: ");
      int timeSteps = sc.nextInt();

      int currentBufferContent = 0;

      System.out.println("\nTime\tIncoming\tAccepted\tDropped\t\tBuffer Status");
      System.out.println("-----------------------------------------------------------------------");

      for (int t = 1; t <= timeSteps; t++) {
         // Generate erratic burst traffic between 0 and twice the output rate
         int incomingPacket = rand.nextInt(outputRate * 2 + 5);
         int acceptedPacket = 0;
         int droppedPacket = 0;

         if (incomingPacket + currentBufferContent <= bucketSize) {
            acceptedPacket = incomingPacket;
            currentBufferContent += incomingPacket;
         } else {
            acceptedPacket = bucketSize - currentBufferContent;
            droppedPacket = incomingPacket - acceptedPacket;
            currentBufferContent = bucketSize;
         }
```

```
            System.out.print(t + "\t" + incomingPacket + "\t\t" + acceptedPacket + "\t\t" +
droppedPacket + "\t\t" + currentBufferContent + " (Before Leak)");

            // Apply steady leakage flow
            if (currentBufferContent >= outputRate) {
               currentBufferContent -= outputRate;
            } else {
               currentBufferContent = 0; // Bucket completely drains out
            }

            System.out.println(" -> " + currentBufferContent + " (After Leak)");
            Thread.sleep(600); // Small delay to observe terminal print outs
         }
         sc.close();
      }
}
```

OUT PUT:

```
Enter Bucket Capacity (BufferSize): 15
Enter Constant Output Leak Rate: 4
Enter total time steps to simulate: 5

Time   Incoming     Accepted     Dropped          Buffer Status
---------------------------------------------------------------------
1      3            3            0            3 (Before Leak) -> 0 (After
Leak)
2      11           11           0            11 (Before Leak) -> 7 (After
Leak)
3      12           8            4            15 (Before Leak) -> 11 (After
Leak)
4      2            2            0            13 (Before Leak) -> 9 (After
Leak)
5      9            6            3            15 (Before Leak) -> 11 (After
Leak)
```

//5. A Java program to simulate the Stop-and-Wait Sliding Window protocol for flow control.

Source Code:

```
import java.io.*;
import java.util.Random;
public class StopAndWait {
    public static void main(String[] args) throws Exception {
        BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
        Random rand = new Random();
        System.out.print("Enter the total number of frames to transmit: ");
        int totalFrames = Integer.parseInt(br.readLine());
        int currentFrame = 1;
            while (currentFrame <= totalFrames) {
           System.out.println("\n[Sender] Sending Frame " + currentFrame + "...");
           Thread.sleep(800);
            int outcome = rand.nextInt(3); // Outcomes: 0 = Clear, 1 = Frame Drop, 2 = ACK Drop
             if (outcome == 1) {
               System.out.println("[Network] Warning: Frame " + currentFrame + " dropped!");
               System.out.println("[Sender] Timeout! Retransmitting frame...");
           }
           else if (outcome == 2) {
               System.out.println("[Receiver] Frame " + currentFrame + " processed.");
               System.out.println("[Receiver] Dispatched ACK " + (currentFrame + 1));
               System.out.println("[Network] Warning: ACK " + (currentFrame + 1) + " dropped!");
               System.out.println("[Sender] Timeout! Retransmitting frame...");
           }
```

```
            else {
                System.out.println("[Receiver] Frame " + currentFrame + " processed.");
                System.out.println("[Receiver] Dispatched ACK " + (currentFrame + 1));
                System.out.println("[Sender] ACK " + (currentFrame + 1) + " verified.");
                currentFrame++;
            }
        }
        System.out.println("\nAll transmissions completed!");
    }
}
```

Output:

Enter the total number of frames to transmit: 2

[Sender] Sending Frame 1...

[Receiver] Frame 1 processed.

[Receiver] Dispatched ACK 2

[Sender] ACK 2 verified.

[Sender] Sending Frame 2...

[Network] Warning: Frame 2 dropped!

[Sender] Timeout! Retransmitting frame...

[Sender] Sending Frame 2...

[Receiver] Frame 2 processed.

[Receiver] Dispatched ACK 3

[Sender] ACK 3 verified.

All transmissions completed!

// 6. A Java program to implement the Two-Dimensional Redundancy Parity Check system to compute block check validation bits.

Source Code:

```
import java.util.Scanner;
public class TwoDimensionalParity {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter total structural Rows to allocate: ");
        int rows = sc.nextInt();
        System.out.print("Enter total data bit Columns per Row: ");
        int cols = sc.nextInt();
        // Matrix dimensions include space for parity columns and row bits
        int[][] dataGrid = new int[rows + 1][cols + 1];
        System.out.println("Enter binary bit matrices tokens (Enter 0 or 1 line by line):");
        for (int i = 0; i < rows; i++) {
            int rowOnesCount = 0;
            for (int j = 0; j < cols; j++) {
                dataGrid[i][j] = sc.nextInt();
                if (dataGrid[i][j] == 1) {
                    rowOnesCount++;
                }
            }
            // Assign calculated horizontal row bit check (Even parity logic)
            dataGrid[i][cols] = (rowOnesCount % 2 == 0) ? 0 : 1;
        }
        // Calculate vertical column parity values across block fields
        for (int j = 0; j <= cols; j++) {
```

```
            int colOnesCount = 0;
            for (int i = 0; i < rows; i++) {
                if (dataGrid[i][j] == 1) {
                    colOnesCount++;
                }
            }
            dataGrid[rows][j] = (colOnesCount % 2 == 0) ? 0 : 1;
        }
        // Print final generated transmission array block output format
        System.out.println("\n====== GENERATED 2D EVEN PARITY DATA MATRIX ======");
        for (int i = 0; i <= rows; i++) {
            if (i == rows) {
                System.out.println("-----------------------------------------------");
                System.out.print("V-Parity: ");
            }
            for (int j = 0; j <= cols; j++) {
                if (j == cols && i != rows) {
                    System.out.print("| H-Parity: ");
                }
                System.out.print(dataGrid[i][j] + " ");
            }
            System.out.println();
        }
System.out.println("================================================");
        sc.close();
    }
}
```

Output:

Enter total structural Rows to allocate: 3

Enter total data bit Columns per Row: 4

Enter binary bit matrices tokens (Enter 0 or 1 line by line):

1 1 0 0

0 1 1 1

1 0 1 0

====== GENERATED 2D EVEN PARITY DATA MATRIX ======

1 1 0 0 | H-Parity: 0

0 1 1 1 | H-Parity: 1

1 0 1 0 | H-Parity: 0

V-Parity: 0 0 0 1 1

==========

// 7. a) Java Program For Bit Stuffing Using Java

```
import java.util.Scanner;
public class Bitstf {

   // Function to perform bit stuffing
   public static String bitStuffing(String data) {
      int count = 0;
      StringBuilder stuffedData = new StringBuilder();

      for (int i = 0; i<data.length(); i++) {
         if (data.charAt(i) == '1') {
            count++;
         } else {
            count = 0;
         }
stuffedData.append(data.charAt(i));
         // If there are five consecutive 1's, add a 0 after them
         if (count == 5 &&i != data.length() - 1) {
stuffedData.append('0');
            count = 0; // Reset count after stuffing
         }
      }
      return stuffedData.toString();
   }
 // Function to perform bit destuffing
   public static String bitDestuffing(String stuffedData) {
      int count = 0;
      StringBuilder destuffedData = new StringBuilder();

      for (int i = 0; i<stuffedData.length(); i++) {
destuffedData.append(stuffedData.charAt(i));
         if (stuffedData.charAt(i) == '1') {
            count++;
         } else {
            count = 0;
         }
         // If there are five consecutive 1's followed by 0, skip the next bit
         if (count == 5 &&i<stuffedData.length() - 1 &&stuffedData.charAt(i + 1) == '0') {
i++;
            count = 0; // Reset count after destuffing
         }
      }
      return destuffedData.toString();
```

```
    }
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
System.out.println("Enter data to be stuffed (binary format): ");
        String data = scanner.nextLine();

        // Perform bit stuffing
        String stuffedData = bitStuffing(data);
        System.out.println("Stuffed data: " + stuffedData);
        String destuffedData = bitDestuffing(stuffedData);
System.out.println("Destuffed data: " + destuffedData);
Scanner.close();
    }
}
```

Input/Output:

Enter data to be stuffed (binary format):

11111111111000000011111

Stuffed data: 111110111111010000000111111

Destuffed data: 11111111111000000011111

//7.b) Java Program For Character Stuffing Using Java

```
import java.io.*;
import java.lang.*;
import java.util.*;

class chst
{
public static void main(String args[])
        {
                Scanner k=new Scanner(System.in);
                System.out.println("Enter the string\t");
                String s=k.nextLine();
                String str1;
                String str2="";
                int i,j,m;
                m=s.length();
                System.out.println("Original Data "+s);
                str1= "dleghththtx";
                for(i=0;i<=m-1;i++)
                {
                if((s.charAt(i)=='d')&&(s.charAt(i+2)=='e'))
                {
                str1=str1+ "dle";
                }
                str1=str1+s.substring(i,i+1);
                }
                str1=str1+"dlestx";
                int p=str1.length();
                System.out.println("trasmitted data " +str1);
                }
}
```

Input/ Output:

```
C:\dccn>javac chst.java

C:\dccn>java chst

Enter the string

GTMT

Original Data GTMT

trasmitted data dleghththtxGTMTdlestx
```

//8. Design And Develop A Java Program To Implement The CRC Polynomials Is Sender Side And Receiver Side.

```
//CRC
import java.io.*;
class crc
{
 public static void main(String args[]) throws IOException
 {
  BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
  System.out.println("Enter Generator:");
  String gen = br.readLine();
  System.out.println("Enter Data:");
  String data = br.readLine();
  String code = data;
  while(code.length() < (data.length() + gen.length() - 1))
   code = code + "0";
  code = data + div(code,gen);
  System.out.println("The transmitted Code Word is: " + code);
  System.out.println("Please enter the received Code Word: ");
  String rec = br.readLine();
  if(Integer.parseInt(div(rec,gen)) == 0)
   System.out.println("The received code word contains no errors.");
  else
   System.out.println("The received code word contains errors.");
 }
  static String div(String num1,String num2)
 {
  int pointer = num2.length();
  String result = num1.substring(0, pointer);
  String remainder = "";
  for(int i = 0; i < num2.length(); i++)
  {
   if(result.charAt(i) == num2.charAt(i))
    remainder += "0";
   else
    remainder += "1";
  }
  while(pointer < num1.length())
  {
   if(remainder.charAt(0) == '0')
   {
    remainder = remainder.substring(1, remainder.length());
    remainder = remainder + String.valueOf(num1.charAt(pointer));
```

```
   pointer++;
  }
  result = remainder;
  remainder = "";
  for(int i = 0; i < num2.length(); i++)
  {
   if(result.charAt(i) == num2.charAt(i))
    remainder += "0";
   else
    remainder += "1";
  }
 }
 return remainder.substring(1,remainder.length());
}
}
```

Output1:

Enter Generator:

1011

Enter Data:

1001010

The transmitted Code Word is: 1001010111

Please enter the received Code Word:

1001010111

The received code word contains no errors.

Output2:

Enter Generator:

1011

Enter Data:

1001010

The transmitted Code Word is: 1001010111

Please enter the received Code Word:

1001010011

The received code word contains errors.

//9. Design And Develop A Java Program For Implement Of The Data Link Layer Framing Method (Character String).

```
import java.util.Scanner;
public class PrintStringChars5 {
	private static Scanner sc;
	public static void main(String[] args) {
		String str;
		sc= new Scanner(System.in);
		System.out.print("\nPlease Enter any String to Print =  ");
		str = sc.nextLine();
		PrintStringCharacters(str);
	}
	public static void PrintStringCharacters(String str) {

		for(int i = 0; i < str.length(); i++)
		{
			if(str.charAt(i) != ' ') {
				System.out.println("The Character at Position "  + i + " =  " +
str.charAt(i));
			}
		}
	}
}
```

Input/output:

```
Please Enter any String to Print =  GATEMCA
The Character at Position 0 =  G
The Character at Position 1 =  A
The Character at Position 2 =  T
The Character at Position 3 =  E
The Character at Position 4 =  M
The Character at Position 5 =  C
The Character at Position 6 =  A
```

//10. Design And Develop A Java Program For Implementation Of The Parity Check Bit (Even And Odd) .

```
package oddevenparity;
import java.io.*;
import java.util.*;
public class OddEvenParity {
String dataR=new String();
String dataS=new String();
int i,option,len;
int count1,count0;
void getDataSender()throws IOException{
System.out.println("----------At Sender----------");
System.out.println("Enter the Binary data:");
BufferedReader br=new BufferedReader(new InputStreamReader(System.in));
dataS=br.readLine();
len=dataS.length();
for(i=0;i<dataS.length();i++){
if(dataS.charAt(i)!='1' && dataS.charAt(i)!='0'){
System.out.println("Enter only binary values");
return;
}
}
System.out.println("Even parity or Odd Parity?");
System.out.println("Enter 1 for even or 2 for odd");
Scanner s=new Scanner(System.in);
option=s.nextInt();
if(option==1){
System.out.println("Sender has agreed for Even parity");
even();
}
else{
System.out.println("Sender has agreed for odd parity");
odd();
}
System.out.println();
}
void even()throws IOException{
count1=count0=0;
for(i=0;i<dataS.length();i++){
if(dataS.charAt(i)=='1'){
count1++;
}
if(dataS.charAt(i)=='0'){
```

```
count0++;
}
}
if(count1%2==0){
dataS=dataS+'0';
System.out.println("Data after even parity:");
for(i=0;i<dataS.length();i++){
System.out.print(dataS.charAt(i));
}
}
else{
dataS=dataS+'1';
System.out.println("Data after even parity:");
for(i=0;i<dataS.length();i++){
System.out.print(dataS.charAt(i));
}
}
}
void odd()throws IOException{
count1=count0=0;
for(i=0;i<dataS.length();i++){
if(dataS.charAt(i)=='1'){
count1++;
}
if(dataS.charAt(i)=='0'){
count0++;
}
}
if(count1%2!=0){
dataS=dataS+'0';
System.out.println("Data after odd parity:");
for(i=0;i<dataS.length();i++){
System.out.print(dataS.charAt(i));
}
}
else{
dataS=dataS+'1';
System.out.println("Data after odd parity:");
for(i=0;i<dataS.length();i++){
System.out.print(dataS.charAt(i));
}
}
}
void getDataReciver()throws IOException{
```

```
System.out.println("\n\n----------At Receiver----------");
System.out.println("Enter the Binary data:");
BufferedReader br=new BufferedReader(new InputStreamReader(System.in));
dataR=br.readLine();
len+=1;
if(len!=dataR.length()){
System.out.println("Enter the Binary data of length: "+len);
return;
}
for(i=0;i<dataR.length();i++){
if(dataR.charAt(i)!='1' && dataR.charAt(i)!='0'){
System.out.println("Enter only binary values");
return;
}
}
count1=count0=0;
for(i=0;i<dataR.length();i++){
if(dataR.charAt(i)=='1'){
count1++;
}
if(dataR.charAt(i)=='0'){
count0++;
}
}
if(option==1){
if(count1%2==0){
System.out.println("Data received successfully with no error");

}
else{
System.out.println("Error has been detected in received data");
}
}
else{
if(count1%2!=0){
System.out.println("Data received successfully with no error");
//System.out.println("Error has been detected in received data");
}
else{
System.out.println("Error has been detected in received data");
}
}
System.out.println();
}
```

```
public static void main(String[] args) throws IOException {
OddEvenParity p=new OddEvenParity();
p.getDataSender();
p.getDataReciver();
}
}
```

Output:

Output1:

```
----------At Sender----------
Enter the Binary data:
01101110101
Even parity or Odd Parity?
Enter 1 for even or 2 for odd
2
Sender has agreed for odd parity
Data after odd parity:
011011101010


----------At Receiver----------
Enter the Binary data:
011011101010
Data received successfully with no error
```

Output2:

```
----------At Sender----------
Enter the Binary data:
101100111011
Even parity or Odd Parity?
Enter 1 for even or 2 for odd
1
Sender has agreed for Even parity
Data after even parity:
1011001110110


----------At Receiver----------
Enter the Binary data:
1011001110111
Error has been detected in received data
```

//11.Java Program to Implement the RSA Algorithm for String Encryption

```
import java.math.*;
import java.util.*;
import java.math.BigInteger;
import java.util.Scanner;
public class RSA {

    public static void main(String args[]) {
        int p, q, n, phi, e, d = 0, i;
        Scanner scanner = new Scanner(System.in);

        // Input two prime numbers
        System.out.println("Enter two prime numbers:");
        p = scanner.nextInt();
        q = scanner.nextInt();

        // Calculating n and phi
        n = p * q;
        phi = (p - 1) * (q - 1);

        // Generating public key
        for (e = 2; e < phi; e++) {
            // e must be less than phi and GCD of (e, phi) must be 1
            if (gcd(e, phi) == 1) {
                break;
            }
        }
        System.out.println("Public key (e): " + e);

        // Generating private key
        for (i = 0; ; i++) {
            int x = 1 + (i * phi);
            if (x % e == 0) {
                d = x / e;
                break;
            }
        }
        System.out.println("Private key (d): " + d);

        // Input message as a string
        System.out.println("Enter a message to encrypt:");
        scanner.nextLine(); // Consume newline left from previous nextInt()
        String plainTextStr = scanner.nextLine();
```

```
        // Convert string to ASCII numeric representation
        int[] asciiValues = stringToAscii(plainTextStr);

        // Encrypting the data
        BigInteger[] encryptedMsg = new BigInteger[asciiValues.length];
        for (int j = 0; j < asciiValues.length; j++) {
            BigInteger msg = BigInteger.valueOf(asciiValues[j]);
            encryptedMsg[j] = msg.modPow(BigInteger.valueOf(e), BigInteger.valueOf(n));
        }
        System.out.println("Encrypted message:");

        for (BigInteger msg : encryptedMsg) {
            System.out.print(msg + "");
        }
        System.out.println();

        // Decrypting the data
        int[] decryptedValues = new int[encryptedMsg.length];
        for (int j = 0; j < encryptedMsg.length; j++) {
            BigInteger decryptedMsg = encryptedMsg[j].modPow(BigInteger.valueOf(d),
BigInteger.valueOf(n));
            decryptedValues[j] = decryptedMsg.intValue();
        }

        // Convert ASCII values back to string
        String decryptedStr = asciiToString(decryptedValues);
        System.out.println("Original Message After Decryption: " + decryptedStr);
    }

    // Method to compute GCD
    static int gcd(int e, int phi) {
        if (e == 0)
            return phi;
        else
            return gcd(phi % e, e);
    }

    // Convert a string to an array of ASCII values
    static int[] stringToAscii(String str) {
        int[] asciiValues = new int[str.length()];
        for (int i = 0; i < str.length(); i++) {
            asciiValues[i] = (int) str.charAt(i);
        }
        return asciiValues;
```

```
    }

    // Convert an array of ASCII values back to a string
    static String asciiToString(int[] asciiValues) {
        StringBuilder sb = new StringBuilder();
        for (int ascii : asciiValues) {
            sb.append((char) ascii);
        }
        return sb.toString();
    }
}
```

Input/Output:

Enter two prime numbers:

7

13

Public key (e): 5

Private key (d): 29

Enter a message to encrypt:

GATEINSTITUTIONS

Encrypted message:

153928624778832847285028475378 83

Original Message After Decryption: **GATEINSTITUTIONS**

//12. A java program to execute and playfair Cipher Substitution technique using Java.

```
import java.util.*;

public class PlayfairCipher {

   private static final String ALPHABET = "ABCDEFGHIKLMNOPQRSTUVWXYZ"; // 'J' is omitted

   public static void main(String[] args) {

      Scanner scanner = new Scanner(System.in);

      // Input key for Playfair Cipher

      System.out.println("Enter the key for Playfair Cipher:");

      String key = scanner.nextLine().toUpperCase().replace("J", "I");

      // Input plaintext

      System.out.println("Enter the plaintext:");

      String plaintext = scanner.nextLine().toUpperCase().replace("J", "I");

      // Format plaintext

      String formattedPlaintext = formatPlaintext(plaintext);

      char[][] matrix = createMatrix(key);

      System.out.println("Encryption Matrix:");

      printMatrix(matrix);

      // Encrypt the plaintext

      String ciphertext = encrypt(formattedPlaintext, matrix);

      System.out.println("Ciphertext: " + ciphertext);

      scanner.close();

   }  // Format plaintext into digraphs

   private static String formatPlaintext(String text) {

      StringBuilder formatted = new StringBuilder();

      for (int i = 0; i < text.length(); i++) {

         char ch = text.charAt(i);
```

```
        if (ALPHABET.indexOf(ch) != -1) {
            formatted.append(ch);
            if (i < text.length() - 1 && text.charAt(i) == text.charAt(i + 1)) {
                formatted.append('X'); // Insert 'X' between repeated letters
            }
        }
    }
    if (formatted.length() % 2 != 0) {
        formatted.append('X'); // Add 'X' if odd length
    }   return formatted.toString();
}   // Create the 5x5 matrix from the key
private static char[][] createMatrix(String key) {
    Set<Character> seen = new HashSet<>();
    char[][] matrix = new char[5][5];
    int index = 0;
    // Add characters from the key
    for (char ch : key.toCharArray()) {
        if (seen.add(ch) && ALPHABET.indexOf(ch) != -1) {
            matrix[index / 5][index % 5] = ch;
            index++;
        }
    }
    // Add remaining letters of the alphabet
    for (char ch : ALPHABET.toCharArray()) {
        if (seen.add(ch)) {
            matrix[index / 5][index % 5] = ch;
            index++;
```

```
        }
    }
    return matrix;
}
// Print the matrix
private static void printMatrix(char[][] matrix) {
    for (char[] row : matrix) {
        System.out.println(Arrays.toString(row));
    }
}
// Encrypt the plaintext using the matrix
private static String encrypt(String text, char[][] matrix) {
    StringBuilder ciphertext = new StringBuilder();
    for (int i = 0; i < text.length(); i += 2) {
        char a = text.charAt(i);
        char b = text.charAt(i + 1);
        int[] posA = findPosition(a, matrix);
        int[] posB = findPosition(b, matrix);
        if (posA[0] == posB[0]) { // Same row
            ciphertext.append(matrix[posA[0]][(posA[1] + 1) % 5]);
            ciphertext.append(matrix[posB[0]][(posB[1] + 1) % 5]);
        } else if (posA[1] == posB[1]) { // Same column
            ciphertext.append(matrix[(posA[0] + 1) % 5][posA[1]]);
            ciphertext.append(matrix[(posB[0] + 1) % 5][posB[1]]);
        } else { // Rectangle
            ciphertext.append(matrix[posA[0]][posB[1]]);
            ciphertext.append(matrix[posB[0]][posA[1]]);
```

```
            }
        }
        return ciphertext.toString();
    }
    // Find the position of a character in the matrix
    private static int[] findPosition(char ch, char[][] matrix) {
        for (int i = 0; i < 5; i++) {
            for (int j = 0; j < 5; j++) {
                if (matrix[i][j] == ch) {
                    return new int[]{i, j};
                }
            }
        }
        return new int[]{-1, -1}; // Not found
    }
}
```

Input/output:

Enter the key for Playfair Cipher: GATEMCA

Enter the plaintext: TODAY IS SATURDAY

Encryption Matrix:

[G, A, T, E, M]

[C, B, D, F, H]

[I, K, L, N, O]

[P, Q, R, S, U]

[V, W, X, Y, Z]

Ciphertext: MLBTVNRYQEMRXLEW