**PRACTICAL NO: 2**

**AIM: Implementing following Cipher**

**1)** **Implementing Vernam Cipher**

**CODE:**

import java.util.\*;

public class VernamCipher {

public static void main(String[] args) {

String text,key,output="",dec="";

char t,k;

int x;

Scanner s = new Scanner(System.in);

System.out.println("Enter text to Encrypt/Decrypt:");

text=s.nextLine();

System.out.println("Enter any key of length "+text.length());

key=s.nextLine();

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

{

t=text.charAt(i);

k=key.charAt(i);

x=t^k;

output+=(char)(x+96);

}

System.out.println("Encrypted Text is:"+output);

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

{

t=output.charAt(i);

k=key.charAt(i);

x=t^k;

dec+=(char)(x+96);

}

System.out.println("Decrypted Text is:"+dec);

}

}

**OUTPUT:**

A computer screen with blue text

Description automatically generated

**Q2)Implementing RailFence Cipher .**

**CODE:**

import java.util.Scanner;

public class railFence

{

    public static void main(String args[])

    {

    try

    {

    Scanner sc=new Scanner(System.in);

    int depth;

    String plainText,cipherText,decryptedText;

    System.out.println("Enter plain Text: ");

    plainText=sc.nextLine();

    System.out.println("Enter Depth of Encryption:");

    depth=sc.nextInt();

    cipherText=Encryption(plainText,depth);

    System.out.println("Encrypted Text is:\n"+cipherText);

    decryptedText=Decryption(cipherText,depth);

    System.out.println("Decrypted Text is:\n"+decryptedText);

    }

    catch(Exception e)

    {}

    }

    static String Encryption(String plainText,int depth)throws Exception

    {

        int r=depth,len=plainText.length();

        int c=len/depth;

        char mat[][]=new char[r][c];

        int k=0;

        String cipherText="";

        for(int i=0;i<c;i++)

        {

            for (int j=0;j<r;j++)

            {

                if(k!=len)

                {

                    mat[j][i]=plainText.charAt(k++);

                }

                else

                    mat[j][i]='X';

            }

        }

        for(int i=0;i<r;i++)

        {

            for(int j=0;j<c;j++)

            {

                cipherText+=mat[i][j];

            }

        }

        return cipherText;

    }

    static String Decryption(String cipherText,int depth)

    {

        int r=depth,len=cipherText.length();

        int c=len/depth;

        char mat[][]=new char[r][c];

        int k=0;

        String plainText="";

        for(int i=0;i<r;i++)

        {

            for(int j=0;j<c;j++)

            {

                mat[i][j]=cipherText.charAt(k++);

            }

        }

        for(int i=0;i<c;i++)

        {

            for(int j=0;j<r;j++)

            {

                plainText+=mat[j][i];

            }

        }

        return plainText;

    }

    }

**OUTPUT:**

