

EX.NO.2

HILL CIPHER – ENCRYPTION AND DECRYPTION

CODE:

```
import java.util.Scanner;

class HillCipher {
    String plainText = new String();
    String cipherText = new String();
    String key = new String();
    int det;
    int[][] key_mat = new int[3][3];
    int[][] plain_mat = new int[3][1];
    int[][] cipher_mat = new int[3][1];
    int[][] key_inv_mat = new int[3][3];
    int[][] adj = new int[3][3];

    void printPlainMat() {
        System.out.println("\nThe plain text vector is ");
        for (int i = 0; i < 3; i++) {
            plain_mat[i][0] = plainText.charAt(i) - 97;
            System.out.println(plain_mat[i][0]);
        }
    }

    void printCipherMat() {
        System.out.println("\nThe cipher text vector is ");
        for (int i = 0; i < 3; i++) {
            cipher_mat[i][0] = cipherText.charAt(i) - 97;
            System.out.println(cipher_mat[i][0]);
        }
    }

    void printKeyMat() {
        System.out.println("\nThe key matrix is\n ");
        int k = 0;
        for (int i = 0; i < 3; i++) {
            for (int j = 0; j < 3; j++) {
                key_mat[i][j] = (int) key.charAt(k) - 97;
            }
        }
    }
}
```

```

        System.out.print(key_mat[i][j] + " ");
        k++;
    }
    System.out.println();
}
}

```

```

void printKeyInverseMat(int detInv) {
    System.out.println("\nThe inverse of the key matrix is\n ");
    int k = 0;
    for (int i = 0; i < 3; i++) {
        for (int j = 0; j < 3; j++) {
            key_inv_mat[i][j] = (adj[i][j] * detInv) % 26;
            System.out.print(key_inv_mat[i][j] + " ");
            k++;
        }
        System.out.println();
    }
}

```

```

void encrypt() {
    System.out.println("\nThe cipher text vector is ");
    for (int i = 0; i < 3; i++) {
        for (int j = 0; j < 1; j++) {
            cipher_mat[i][j] = 0;

            for (int k = 0; k < 3; k++) {
                cipher_mat[i][j] += key_mat[i][k] * plain_mat[k][j];
            }

            cipher_mat[i][j] = cipher_mat[i][j] % 26;
            System.out.print(cipher_mat[i][j] + " ");
        }
        System.out.println();
    }
    // cipherText="";
    for (int i = 0; i < 3; i++) cipherText += (char) (cipher_mat[i][0] + 97);
    System.out.println("\nThe ciphertext is: " + cipherText);
}

```

```

void decrypt() {
    System.out.println("\nThe plain text vector is ");
    for (int i = 0; i < 3; i++) {
        for (int j = 0; j < 1; j++) {
            plain_mat[i][j] = 0;

            for (int k = 0; k < 3; k++) {
                plain_mat[i][j] += key_inv_mat[i][k] * cipher_mat[k][j];
            }

            plain_mat[i][j] = plain_mat[i][j] % 26;
            System.out.print(plain_mat[i][j] + " ");
        }
        System.out.println();
    }
    plainText = "";
    for (int i = 0; i < 3; i++) plainText += (char) (plain_mat[i][0] + 97);
    System.out.println("\nThe plaintext is: " + plainText);
}

```

```

public boolean isInvertible() {
    det = findDet(key_mat, 3);
    // System.out.println("det is "+det);
    if (det == 0 || det % 2 == 0 || det % 13 == 0) {
        return false;
    } else {
        return true;
    }
}

```

```

int modInverse(int a, int m) {
    a = a % m;
    for (int x = 1; x < m; x++) if ((a * x) % m == 1) return x;
    return 1;
}

```

```

void findAdjoint() {
    int sign = 1;
    int[][] temp = new int[3][3];

```

```

for (int i = 0; i < 3; i++) {
    for (int j = 0; j < 3; j++) {
        getCofactor(key_mat, temp, i, j, 3);
        sign = ((i + j) % 2 == 0) ? 1 : -1;
        adj[j][i] = (sign) * (findDet(temp, 2));
        while (adj[j][i] < 0) {
            adj[j][i] += 26;
        }
    }
}
}

```

```

void getCofactor(int mat[][3], int temp[][3], int p, int q, int n) {
    int i = 0, j = 0;
    for (int row = 0; row < n; row++) {
        for (int col = 0; col < n; col++) {
            if (row != p && col != q) {
                temp[i][j++] = mat[row][col];
                if (j == n - 1) {
                    j = 0;
                    i++;
                }
            }
        }
    }
}

```

```

int findDet(int mat[][3], int n) {
    int det = 0;
    if (n == 1) return mat[0][0];
    int temp[][3] = new int[3][3];
    int sign = 1;
    for (int f = 0; f < n; f++) {
        getCofactor(mat, temp, 0, f, n);
        det += sign * mat[0][f] * findDet(temp, n - 1);
        sign = -sign;
    }
    while (det < 0) {
        det += 26;
    }
}

```

```
return det % 26;  
}
```

```
public static void main(String[] args) {  
    HillCipher hc = new HillCipher();  
    Scanner sc = new Scanner(System.in);  
    System.out.println("HILL CIPHER");  
    System.out.println("\nENCRYPTION");  
    System.out.println("*****");  
    System.out.print("\nEnter plain text: ");  
    hc.plainText = sc.next();  
    System.out.print("\nEnter the key: ");  
    hc.key = sc.next();  
    hc.printKeyMat();  
    hc.printPlainMat();  
    hc.encrypt();  
  
    System.out.println("\nDECRYPTION");  
    System.out.println("*****");  
    System.out.print("\nEnter cipher text: ");  
    hc.cipherText = sc.next();  
    System.out.print("\nEnter the key: ");  
    hc.key = sc.next();  
    hc.printKeyMat();  
    if (!hc.isInvertible()) System.out.println("Key is not invertible"); else {  
        int detInv = hc.modInverse(hc.det, 26);  
        hc.findAdjoint();  
        hc.printKeyInverseMat(detInv);  
        hc.printCipherMat();  
        hc.decrypt();  
    }  
    sc.close();  
}
```

OUTPUT:

Example 1:

```
C:\Users\WELCOME\Desktop\CNS lab\ex2>java HillCipher  
HILL CIPHER
```

```
ENCRYPTION
```

```
*****
```

```
Enter plain text: mat
```

```
Enter the key: alphabeta
```

```
The key matrix is
```

```
0 11 15
```

```
7 0 1
```

```
4 19 0
```

```
The plain text vector is
```

```
12
```

```
0
```

```
19
```

```
The cipher text vector is
```

```
25
```

```
25
```

```
22
```

```
The ciphertext is: zzw
```

```
DECRYPTION
```

```
*****
```

```
Enter cipher text: zzw
```

```
Enter the key: alphabeta
```

```
The key matrix is
```

```
0 11 15
```

```
7 0 1
```

```
4 19 0
```

The inverse of the key matrix is

3 7 1
24 4 19
5 4 19

The cipher text vector is

25
25
22

The plain text vector is

12
0
19

The plaintext is: mat

Example 2:

HILL CIPHER

ENCRYPTION

Enter plain text: sit

Enter the key: hillmagic

The key matrix is

7 8 11
11 12 0
6 8 2

The plain text vector is

18
8
19

The cipher text vector is

9
8
2

The ciphertext is: jic

DECRYPTION

Enter cipher text: jic

Enter the key: hillmagic

The key matrix is

```
7 8 11
11 12 0
6 8 2
```

Key is not invertible

C:\Users\WELCOME\Desktop\CNS lab\ex2>

VIGENERE CIPHER –ENCRYPTION AND DECRYTION

CODE:

```
import java.util.Scanner;

public class VigenereCipher {

    String plainText = new String();
    String cipherText = new String();
    String key = new String();

    void generateKey() {
        int diff;
        while (key.length() != plainText.length()) {
            diff = plainText.length() - key.length();
            if (diff >= key.length())
                key += key;
            else
                key = key + key.substring(0, diff);
        }
    }

    void encrypt() {
        for (int i = 0; i < plainText.length(); i++) {
```



```

        int x = (plainText.charAt(i) + key.charAt(i)) % 26;
        x += 'A';
        cipherText += (char) (x);
    }
}

```

```

void decrypt() {
    for (int i = 0; i < cipherText.length() && i < key.length(); i++) {
        int x = (cipherText.charAt(i) - key.charAt(i) + 26) % 26;
        x += 'A';
        plainText += (char) (x);
    }
}

```

```

public static void main(String[] args) {
    VigenereCipher vc = new VigenereCipher();
    Scanner sc = new Scanner(System.in);
    System.out.println("VIGENERE CIPHER");
    System.out.println("\nENCRYPTION");
    System.out.println("*****");
    System.out.print("\nEnter plain text: ");
    vc.plainText = sc.next();
    System.out.print("\nEnter the key: ");
    vc.key = sc.next();
    vc.generateKey();
    vc.encrypt();
    System.out.println("\nCipher text is: " + vc.cipherText);

    System.out.println("\nDECRYPTION");
    System.out.println("*****");
    System.out.print("\nEnter cipher text: ");
    vc.cipherText = sc.next();
    System.out.print("\nEnter the key: ");
    vc.key = sc.next();
    vc.generateKey();
    vc.plainText = "";
    vc.decrypt();
    System.out.println("\nPlain text is: " + vc.plainText);
    sc.close();
}
}

```

OUTPUT:

Example 1:

```
C:\Users\WELCOME\Desktop\CNS lab\ex2>java VigenereCipher
VIGENERE CIPHER

ENCRYPTION
*****

Enter plain text: album
Enter the key: block
Cipher text is: bwpww

DECRYPTION
*****

Enter cipher text: bwpww
Enter the key: block
Plain text is: album
C:\Users\WELCOME\Desktop\CNS lab\ex2>_
```

Example 2:

```
C:\Users\WELCOME\Desktop\CNS lab\ex2>java VigenereCipher
VIGENERE CIPHER

ENCRYPTION
*****

Enter plain text: jug
Enter the key: network
The modified key is net
Cipher text is: wyz
```

DECRYPTION

Enter cipher text: wyz

Enter the key: network

The modified key is net

Plain text is: jug

C:\Users\WELCOME\Desktop\CNS lab\ex2>

Example 3:

C:\Users\WELCOME\Desktop\CNS lab\ex2>java VigenereCipher

VIGENERE CIPHER

ENCRYPTION

Enter plain text: administration

Enter the key: rest

The modified key is restrestrestre

Cipher text is: rhebemkmielbfr

DECRYPTION

Enter cipher text: rhebemkmielbfr

Enter the key: rest

The modified key is restrestrestre

Plain text is: administration

C:\Users\WELCOME\Desktop\CNS lab\ex2>_
