

**EXPERIMENT 5**

**Aim:** Write a program to perform Encryption/Decryption using Hill techniques.

**Theory:**

In classical cryptography, the Hill cipher is a polygraphic substitution cipher based on linear algebra. Invented by Lester S. Hill in 1929, it was the first polygraphic cipher in which it was practical (though barely) to operate on more than three symbols at once. The following discussion assumes an elementary knowledge of matrices

**Algorithm:****Encryption:**

We have to encrypt the message 'ACT' (n=3). The key is 'GYBNQKURP' which can be written as the nxn matrix:

$$\begin{bmatrix} 6 & 24 & 1 \\ 13 & 16 & 10 \\ 20 & 17 & 15 \end{bmatrix}$$

The message 'ACT' is written as vector:

$$\begin{bmatrix} 0 \\ 2 \\ 19 \end{bmatrix}$$

The enciphered vector is given as:

$$\begin{bmatrix} 6 & 24 & 1 \\ 13 & 16 & 10 \\ 20 & 17 & 15 \end{bmatrix} \begin{bmatrix} 0 \\ 2 \\ 19 \end{bmatrix} = \begin{bmatrix} 67 \\ 222 \\ 319 \end{bmatrix} \equiv \begin{bmatrix} 15 \\ 14 \\ 7 \end{bmatrix} \pmod{26}$$

which corresponds to ciphertext of 'POH'

**Decryption:**

To decrypt the message, we turn the ciphertext back into a vector, then simply multiply by the inverse matrix of the key matrix (IFKVIVVMI in letters). The inverse of the matrix used in the previous example is:

$$\begin{bmatrix} 6 & 24 & 1 \\ 13 & 16 & 10 \\ 20 & 17 & 15 \end{bmatrix} \begin{bmatrix} 0 \\ 2 \\ 19 \end{bmatrix} = \begin{bmatrix} 67 \\ 222 \\ 319 \end{bmatrix} \equiv \begin{bmatrix} 15 \\ 14 \\ 7 \end{bmatrix} \pmod{26}$$

For the previous Ciphertext 'POH':

$$\begin{bmatrix} 8 & 5 & 10 \\ 21 & 8 & 21 \\ 21 & 12 & 8 \end{bmatrix} \begin{bmatrix} 15 \\ 14 \\ 7 \end{bmatrix} \equiv \begin{bmatrix} 260 \\ 574 \\ 539 \end{bmatrix} \equiv \begin{bmatrix} 0 \\ 2 \\ 19 \end{bmatrix} \pmod{26}$$

which gives us back 'ACT'.

Code:

```
#include<bits/stdc++.h>
using namespace std;
void getKeyMatrix(string key, int keyMatrix[][3]) {
    int k = 0;
    for (int i = 0; i < 3; i++) {
        for (int j = 0; j < 3; j++) {
            keyMatrix[i][j] = (key[k]) % 65;
            k++;
        }
    }
}

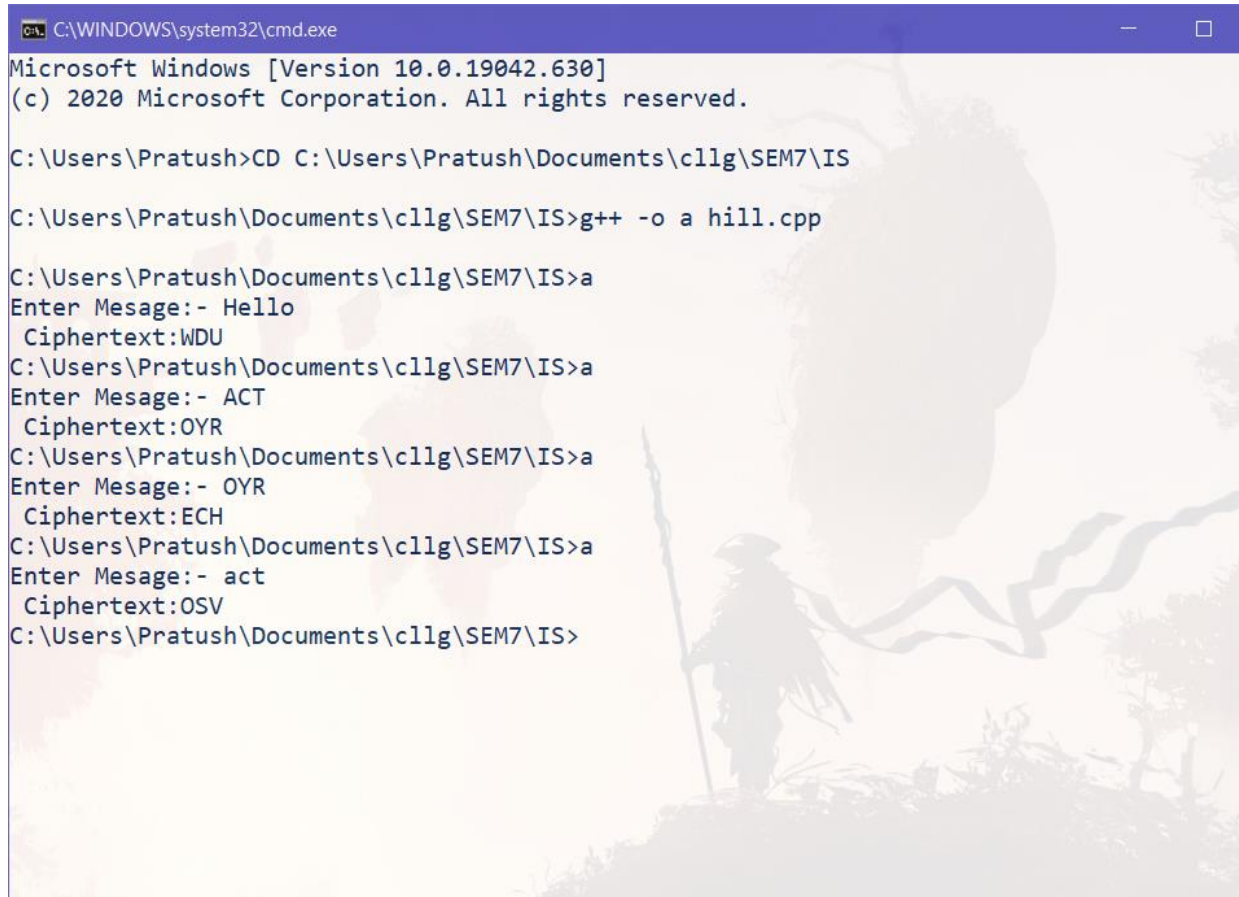
void encrypt(int cipherMatrix[][3], int keyMatrix[][3], int messageVector[][1]) {
    int x, i, j;
    for (i = 0; i < 3; i++) {
        for (j = 0; j < 3; j++) {
            cipherMatrix[i][j] = 0;
            for (x = 0; x < 3; x++) {
                cipherMatrix[i][j] += keyMatrix[i][x] * messageVector[x][j];
            }
            cipherMatrix[i][j] = cipherMatrix[i][j] % 26;
        }
    }
}

void HillCipher(string message, string key) {
    int keyMatrix[3][3];
    getKeyMatrix(key, keyMatrix);
    int messageVector[3][1];
    for (int i = 0; i < 3; i++)
        messageVector[i][0] = (message[i]) % 65;

    int cipherMatrix[3][1];
    encrypt(cipherMatrix, keyMatrix, messageVector);
    string CipherText;
    for (int i = 0; i < 3; i++)
        CipherText += cipherMatrix[i][0] + 65;
    cout << " Ciphertext:" << CipherText;
}

int main() {
    string message;
    string key = "IAMTONSRK";
    cout << "Enter Mesage:- "; getline(cin, message);
    HillCipher(message, key);
    return 0;
}
```

Output:



```
C:\WINDOWS\system32\cmd.exe
Microsoft Windows [Version 10.0.19042.630]
(c) 2020 Microsoft Corporation. All rights reserved.

C:\Users\Pratush>CD C:\Users\Pratush\Documents\c1lg\SEM7\IS

C:\Users\Pratush\Documents\c1lg\SEM7\IS>g++ -o a hill.cpp

C:\Users\Pratush\Documents\c1lg\SEM7\IS>a
Enter Mesage:- Hello
  Ciphertext:WDU
C:\Users\Pratush\Documents\c1lg\SEM7\IS>a
Enter Mesage:- ACT
  Ciphertext:OYR
C:\Users\Pratush\Documents\c1lg\SEM7\IS>a
Enter Mesage:- OYR
  Ciphertext:ECH
C:\Users\Pratush\Documents\c1lg\SEM7\IS>a
Enter Mesage:- act
  Ciphertext:OSV
C:\Users\Pratush\Documents\c1lg\SEM7\IS>
```