INFORMATION SECURITY & CRYPTOGRAPHY
ASSIGNMENT- 3

U20CD005
BANSI MARAKANA

1. Implement encryption and decryption using Hill cipher.
2. Implement encryption and decryption using Vigenere cipher.
Program:

```
#include <bits/stdc++.h>
#include <iostream>
#include <fstream>
#include <math.h>
using namespace std;
typedef long long ll;
int encrypt_msg[3][1], decrypt_msg[3][1], keyMatrix[3][3],
inverseMatrix[3][3], messageVector[3][1], temp[3][3];
int d;
string StringTransform(string msg)
{
    string text = "";
    for (ll i = 0; i < msg.size(); i++)
        if (((msg[i] >= 'a') && (msg[i] <= 'z')) || ((msg[i] >= 'A') &&
(msg[i] <= 'Z')))
            text += toupper(msg[i]);
    while ((text.length() % 3) != 0)
        text += 'Z';
    return text;
}
int mod26(int n)
{
    return n >= 0 ? (n % 26) : 26 - (abs(n) % 26);
}
int modInverse(int d, int m)
{
    for (int r = 1; r < m; r++)
        if (((d % m) * (r % m)) % m == 1)
            return r;
    return 0;
}
void generateKeyMatrix(string key)
{
```

```
    int k = 0, d = 0;
    for (int i = 0; i < 3; i++)
    {
        for (int j = 0; j < 3; j++)
        {
            keyMatrix[i][j] = key[k] - 'A';
            k++;
        }
    }
}
void inverse()
{
    d = 0;
    for (int i = 0; i < 3; i++)
        d = d + (keyMatrix[0][i] * (keyMatrix[1][(i + 1) % 3] *
keyMatrix[2][(i + 2) % 3] - keyMatrix[1][(i + 2) % 3] * keyMatrix[2][(i +
1) % 3]));
    d = mod26(d);
    d = modInverse(d, 26);

    for (int i = 0; i < 3; i++)
    {
        for (int j = 0; j < 3; j++)
        {
            inverseMatrix[i][j] = ((keyMatrix[(j + 1) % 3][(i + 1) % 3] *
keyMatrix[(j + 2) % 3][(i + 2) % 3]) - (keyMatrix[(j + 1) % 3][(i + 2) %
3] * keyMatrix[(j + 2) % 3][(i + 1) % 3])) * d;
            inverseMatrix[i][j] = mod26(inverseMatrix[i][j]);
        }
    }
    cout << "\n\tInverse Matrix is:\n";
    for (int i = 0; i < 3; i++)
    {
        for (int j = 0; j < 3; j++)
            cout << "\t" << inverseMatrix[i][j] << " ";
        cout << "\n";
    }
}
void encrypt_decrypt_hill_cipher(int msg1[][1], int matrix[][3], int
msg2[][1])
```

```
{
    for (int i = 0; i < 3; i++)
    {
        msg1[i][0] = 0;
        for (int j = 0; j < 3; j++)
            msg1[i][0] += (matrix[i][j] * msg2[j][0]);
    }
}
string encrypt_decrypt_Vigenere(string text, string key, int p, int q)
{
    ll n = text.length();
    int k = key.length(), l = 0;
    for (ll i = 0; i < n; i++)
    {
        if (l < k)
        {
            if (islower(text[i]))
                text[i] = mod26(text[i] + p * tolower(key[(i - l + k) %
k]) + q * 'a') + 'a';
            else if (isupper(text[i]))
                text[i] = mod26(text[i] + p * toupper(key[(i - l + k) %
k]) + q * 'A') + 'A';
            else
                l++;
        }
        else
            l = 0;
    }
    return text;
}
int main()
{
    int choice;
    string key;
    cout << "1. Encryption using hill cipher \n2. Decryption using hill
cipher \n3. Encryption using vinegere cipher";
    cout << "\n4. Decryption using vinegere cipher \n5. Exit";
    while (1)
    {
        cout << "\nEnter your choice: ";
```

```
        cin >> choice;
        switch (choice)
        {
        case 1:
        {
            string fname, fname1, key, plain_text = "", cipher_text = "",
text = "";
            cout << "\tEnter file name to read plain text: ";
            cin >> fname;
            cout << "\tEnter file name to write cipher text: ";
            cin >> fname1;
            cout << "\tEnter encryption key: ";
            cin >> key;
            key = StringTransform(key);
            generateKeyMatrix(key);
            ifstream fin;
            ofstream fout;
            fin.open(fname + ".txt");
            if (!fin.is_open())
            {
                cout << "\tFile does not exist!!";
                return 0;
            }
            fout.open(fname1 + ".txt");
            cout << "\tEncrypted text is: " << endl;
            while (getline(fin, text))
                plain_text += text;
            plain_text = StringTransform(plain_text);
            for (int i = 0; i < plain_text.length(); i += 3)
            {
                for (int j = i; j < i + 3; j++)
                    messageVector[j - i][0] = (plain_text[j] - 'A');
                encrypt_decrypt_hill_cipher(encrypt_msg, keyMatrix,
messageVector);
                for (int j = 0; j < 3; j++)
                    cipher_text += mod26(encrypt_msg[j][0]) + 'A';
            }
            cout << "\t" << cipher_text << endl;
            fout << cipher_text;
            fout.close();
```

```
            fin.close();
            break;
        }
        case 2:
        {
            string fname, fname1, key, plain_text = "", cipher_text = "",
text = "";
            cout << "\tEnter file name to read cipher text: ";
            cin >> fname;
            cout << "\tEnter file name to write plain text: ";
            cin >> fname1;
            cout << "\tEnter decryption key: ";
            cin >> key;
            key = StringTransform(key);
            generateKeyMatrix(key);
            inverse();
            ifstream fin;
            ofstream fout;
            fin.open(fname + ".txt");
            if (!fin.is_open())
            {
                cout << "\tFile does not exist!!";
                return 0;
            }
            fout.open(fname1 + ".txt");
            while (getline(fin, text))
                cipher_text += text;
            if (d != 0 && d != 26)
            {
                cout << "\tDecrypted text is: " << endl;
                for (int i = 0; i < cipher_text.length(); i += 3)
                {
                    for (int j = i; j < i + 3; j++)
                        encrypt_msg[j - i][0] = (cipher_text[j] - 'A');
                    encrypt_decrypt_hill_cipher(decrypt_msg,
inverseMatrix, encrypt_msg);
                    for (int j = 0; j < 3; j++)
                        plain_text += mod26(decrypt_msg[j][0]) + 'A';
                }
                cout << "\t" << plain_text << endl;
```

```
                fout << plain_text;
            }
            else
                cout << "\tCannot decrypt this text!!";
            fout.close();
            fin.close();
            break;
        }
        case 3:
        {
            string fname, fname1, key, plain_text, cipher_text;
            cout << "\tEnter file name to read plain text: ";
            cin >> fname;
            cout << "\tEnter file name to write cipher text: ";
            cin >> fname1;
            cout << "\tEnter encryption key: ";
            cin >> key;
            ifstream fin;
            ofstream fout;
            fin.open(fname + ".txt");
            if (!fin.is_open())
            {
                cout << "\tFile does not exist!!";
                return 0;
            }
            fout.open(fname1 + ".txt");
            cout << "\tEncrypted text is: " << endl;
            while (getline(fin, plain_text))
            {
                cipher_text = encrypt_decrypt_Vigenere(plain_text, key, 1,
-2);
                fout << cipher_text << endl;
                cout << "\t" << cipher_text << endl;
            }
            fout.close();
            fin.close();
            break;
        }
        case 4:
        {
```

```
            string fname, fname1, key, plain_text, cipher_text;
            cout << "\tEnter file name to read cipher text: ";
            cin >> fname;
            cout << "\tEnter file name to write plain text: ";
            cin >> fname1;
            cout << "\tEnter decryption key: ";
            cin >> key;
            ifstream fin;
            ofstream fout;
            fin.open(fname + ".txt");
            if (!fin.is_open())
            {
                cout << "\tFile does not exist!!";
                return 0;
            }
            fout.open(fname1 + ".txt");
            cout << "\tDecrypted text is: " << endl;
            while (getline(fin, cipher_text))
            {
                plain_text = encrypt_decrypt_Vigenere(cipher_text, key,
-1, 0);
                fout << plain_text << endl;
                cout << "\t" << plain_text << endl;
            }
            fout.close();
            fin.close();
            break;
        }
        case 5:
            exit(0);
            break;

        default:
            cout << "Please enter valid choice!!";
            break;
        }
    }
}
```

OUTPUT:

Hill Cipher:

p.txt

```
p.txt
Hello World!!
I am learning Cryptography!!
```

c.txt:

```
c.txt
TFJIPIJSGCLOCITFMQCIDCSNOPBCMTWBZ
```

Encryption:

```
PS D:\BANSI MARAKANA\ISC> ./a
1. Encryption using hill cipher
2. Decryption using hill cipher
3. Encryption using vinegere cipher
4. Decryption using vinegere cipher
5. Exit
Enter your choice: 1
        Enter file name to read plain text: p
        Enter file name to write cipher text: c
        Enter encryption key: gybnqkurp
        Encrypted text is:
        TFJIPIJSGCLOCITFMQCIDCSNOPBCMTWBZ
```

Decryption:

```
Enter your choice: 2
        Enter file name to read cipher text: c
        Enter file name to write plain text: p
        Enter decryption key: gybnqkurp

        Inverse Matrix is:
        8       5       10
        21      8       21
        21      12      8
        Decrypted text is:
        HELLOWORLDIAMLEARNINGCRYPTOGRAPHY
```

Vigenere Cipher:

p.txt

```
p.txt
Hello World!!
I am learning Cryptography!!
```

c.txt

```
c.txt
Ieydw Ioiln!!
J az dmmreixg Przpggodaghi!!
```

Encryption:

```
PS D:\BANSI MARAKANA\ISC> ./a
1. Encryption using hill cipher
2. Decryption using hill cipher
3. Encryption using vinegere cipher
4. Decryption using vinegere cipher
5. Exit
Enter your choice: 3
        Enter file name to read plain text: p
        Enter file name to write cipher text: c
        Enter encryption key: BansiMarakana
        Encrypted text is:
        Ieydw Ioiln!!
        J az dmmreixg Przpggodaghi!!
```

Decryption:

```
PS D:\BANSI MARAKANA\ISC> ./a
1. Encryption using hill cipher
2. Decryption using hill cipher
3. Encryption using vinegere cipher
4. Decryption using vinegere cipher
5. Exit
Enter your choice: 4
        Enter file name to read cipher text: c
        Enter file name to write plain text: p
        Enter decryption key: BansiMarakana
        Decrypted text is:
        Hello World!!
        I am learning Cryptography!!
```