Name : Harshal Kodgire
Batch : B1
Subject : CNS lab
Topic : Assignment 6

Aim : - To implement Columnar Transposition Cipher

Theory :
The Columnar Transposition Cipher is a form of transposition cipher just like Rail Fence Cipher. Columnar Transposition involves writing the plaintext out in rows, and then reading the ciphertext off in columns one by one.

Code :

```
#include <bits/stdc++.h>
using namespace std;

#define ll long long

int main()
{
    string plainText,key;

    cout<<"\n Enter plain text : ";
    getline(cin,plainText);

    cout<<"\n Enter key : ";
    getline(cin,key);

    // Removing spaces and converting to small from plaintext
    string temp = "";
    for(int i=0;i<plainText.size();i++)
    {
        if(plainText[i]!=' ')
        temp += plainText[i];
    }
    plainText = temp;

    for(int i=0;i<plainText.size();i++)
    {
        if(plainText[i]>=65 && plainText[i]<=90)
```

```
        plainText[i] += 32;
    }

    // Removing spaces and converting to small from key
    string temp2 = "";
    for(int i=0;i<key.size();i++)
    {
        if(key[i]!=' ')
        temp2 += key[i];
    }
    key = temp2;

    for(int i=0;i<key.size();i++)
    {
        if(key[i]>=65 && key[i]<=90)
        key[i] += 32;
    }

    // Encryption
    map<char,vector<char>> mp;
    int keyCounter = 0;

    for(int i=0;i<plainText.size();i++)
    {
        mp[key[keyCounter++]].push_back(plainText[i]);

        if(keyCounter==key.size())
        keyCounter = 0;
    }

    string cipherText;
    for(auto it:mp)
    {
        for(int i=0;i<it.second.size();i++)
        {
            cipherText += it.second[i];
        }
    }

    cout<<"\n Cipher text is : " << cipherText;

    // Decryption
    map<int,int> dmp;
```

```
int common = cipherText.size()/key.size();
int extra = cipherText.size()%key.size();

for(int i=0;i<key.size();i++)
{
    if(i<extra)
    dmp[i] = common + 1;
    else
    dmp[i] = common;
}

map<int,vector<char>> dmp2;

int start = 0;

string sortedKey = key;
sort(sortedKey.begin(),sortedKey.end());

for(int i=0;i<sortedKey.size();i++)
{
    for(int j=0;j<key.size();j++)
    {
        if(sortedKey[i]==key[j])
        {
            for(int k=0;k<dmp[j];k++)
            {
                dmp2[key[j]].push_back(cipherText[start++]);
            }
        }
    }
}

string afterDecryption;

vector<int> counters(key.size(),0);

int i=0;

while(afterDecryption.size()<cipherText.size())
{
    for(int i=0;i<key.size();i++)
    {
```

```
            if(counters[i]<dmp[i])
            afterDecryption += dmp2[key[i]][counters[i]++];
        }
    }

    cout<<"\n\n Text after decryption is : " << afterDecryption<<endl;
    return 0;
}
```

Output :

```
D:\WCE_ENGINEERING\BTECH_SEM1\CNS lab>g++ Assignment_6.cpp

D:\WCE_ENGINEERING\BTECH_SEM1\CNS lab>a.exe

 Enter plain text : geeksforgeeks

 Enter key : hack

 Cipher text is : efeeoegsgskrk

 Text after decryption is : geeksforgeeks

D:\WCE_ENGINEERING\BTECH_SEM1\CNS lab>
```

```
D:\WCE_ENGINEERING\BTECH_SEM1\CNS lab>a.exe

 Enter plain text : Harshal

 Enter key : ram

 Cipher text is : ahrahsl

 Text after decryption is : harshal

D:\WCE_ENGINEERING\BTECH_SEM1\CNS lab>
```