CNS LAB

Name: Sadaf Najeem Mulla
PRN: 2019BTECS00038

Assignment 5

Aim - Given the plain text, encrypt it using Rail fence Encryption Algorithm

Rail fence Cipher Encryption Algorithm

● In the rail fence cipher, the plain-text is written downwards and diagonally on successive rails of an imaginary fence.

● When we reach the bottom rail, we traverse upwards moving diagonally, after reaching the top rail, the direction is changed again. Thus the alphabets of the message are written in a zig-zag manner.

● After each alphabet has been written, the individual rows are combined to obtain the cipher-text.

Code:

```
string s;
cout << "Enter plain text" << endl;

getline(cin, s);

string x;
for (int i = 0; i < s.length(); i++)
if (s[i] != ' ')
x += s[i];
s = x;

int k;
```

```
cout << "Enter key" << endl;
cin >> k;


cout << "\nPlain text is: " << s << endl; cout <<
"Key is: " << k << endl;

int n = s.length();


vector<vector<char>> mat(k);
int row = 0;
int flg = 1;
for (int i = 0; i < s.length(); i++)
{
mat[row].push_back(s[i]);
row += flg;
if (row == (k - 1))
{
flg = -1;
}
if (row == 0)
flg = 1;
}

string cip = "";
for (int i = 0; i < k; i++)
{
for (int j = 0; j < mat[i].size(); j++)
cip += mat[i][j];
}

s = cip;
transform(cip.begin(), cip.end(), cip.begin(), ::toupper); cout <<
"\nCipher text is: " << cip;


int tp = 1;



vector<vector<int>> matd(k);
row = 0;
flg = 1;
for (int i = 1; i <= n; i++)
```

```
{
matd[row].push_back(i);
row += flg;
if (row == (k - 1))
{
flg = -1;
}
if (row == 0)
flg = 1;
}

vector<int> dd;
for (int i = 0; i < k; i++)
{
for (int j = 0; j < mat[i].size(); j++)
dd.push_back(matd[i][j]);
}
cout << endl;
map<int, char> m;
for (int i = 0; i < n; i++)
m[dd[i]] = s[i];

string plain = "";
for (int i = 1; i <= n; i++)
plain += m[i];
cout << "\n\nPlain text after decription is: " << plain;
```

TestCases: