Worksheet 8

**Student Name:** Himanshu Dubey **UID:** 18BCS1160

**Branch:** CSE **Section/Group:** IS-1/A

**Semester:** 6th **Date of Performance:** 19/04/21

**Subject Name:** IS LAB **Subject Code:** CSB-372

# AIM:

Implement Columnar Transposition Cipher and upload the Code & screenshot of output screen.

**Logistic Used:**

* CodeBlocks (C++ compiler)
* Windows OS

**Algorithm:**

## Encryption

In a transposition cipher, the order of the alphabets is re-arranged to obtain the cipher-text.

* The message is written out in rows of a fixed length, and then read out again column by column, and the columns are chosen in some scrambled order.
* Width of the rows and the permutation of the columns are usually defined by a keyword.
* For example, the word HACK is of length 4 (so the rows are of length 4), and the permutation is defined by the alphabetical order of the letters in the keyword. In this case, the order would be “3 1 2 4”.
* Any spare spaces are filled with nulls or left blank or placed by a character (Example: \_).
* Finally, the message is read off in columns, in the order specified by the keyword.

**Decryption**

* To decipher it, the recipient has to work out the column lengths by dividing the message length by the key length.
* Then, write the message out in columns again, then re-order the columns by reforming the key word.

**CODE:**

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

string key; map<int,int> keyMap;

void setPermutationOrder()

{

for(int i=0; i < key.length(); i++)

{

keyMap[key[i]] = i;

}

}

string encryptMessage(string msg)

{

int row,col,j; string cipher = "";

col = key.length();

row = msg.length()/col;

if (msg.length() % col) row += 1;

char matrix[row][col];

for (int i=0,k=0; i < row; i++)

{

for (int j=0; j<col; )

{

if(msg[k] == '\0')

{

matrix[i][j] = '\_'; j++;

}

if( isalpha(msg[k]) || msg[k]==' ')

{

matrix[i][j] = msg[k]; j++;

} k++;

}

}

for (map<int,int>::iterator ii = keyMap.begin(); ii!=keyMap.end(); ++ii)

{

j=ii->second;

for (int i=0; i<row; i++)

{

if( isalpha(matrix[i][j]) || matrix[i][j]==' ' || matrix[i][j]=='\_') cipher += matrix[i][j];

}

}

return cipher;

}

string decryptMessage(string cipher)

{

int col = key.length();

int row = cipher.length()/col; char cipherMat[row][col];

for (int j=0,k=0; j<col; j++) for (int i=0; i<row; i++)

cipherMat[i][j] = cipher[k++];

int index = 0;

for( map<int,int>::iterator ii=keyMap.begin(); ii!=keyMap.end(); ++ii) ii->second = index++;

char decCipher[row][col]; map<int,int>::iterator ii=keyMap.begin(); int k = 0;

for (int l=0,j; key[l]!='\0'; k++)

{

j = keyMap[key[l++]];

for (int i=0; i<row; i++)

{

decCipher[i][k]=cipherMat[i][j];

}

}

string msg = "";

for (int i=0; i<row; i++)

{

for(int j=0; j<col; j++)

{

if(decCipher[i][j] != '\_') msg += decCipher[i][j];

}

}

return msg;

}

int main(void)

{

string msg ; cin>>msg;

cout<<"Enter key :- "; cin>>key; setPermutationOrder();

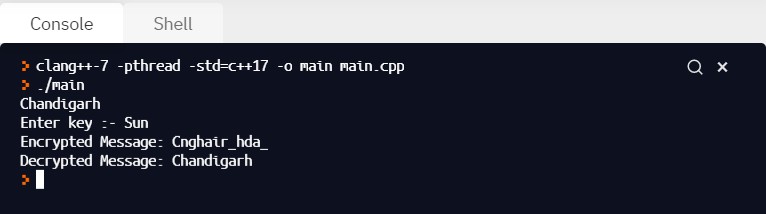
string cipher = encryptMessage(msg);

cout << "Encrypted Message: " << cipher << endl;

cout << "Decrypted Message: " << decryptMessage(cipher) << endl; return 0;

}

# Output:



**Learning outcomes (What I have learnt):**

* 1. Understood the Columnar Transposition Cipher Algorithm.
  2. Implementation of Columnar Transposition Cipher Algorithm.
  3. Applications of Columnar Transposition Cipher Algorithm**.**

**Evaluation Grid (To be created as per the SOP and Assessment guidelines by the faculty):**

|  |  |  |  |
| --- | --- | --- | --- |
| Sr. No. | Parameters | Marks Obtained | Maximum Marks |
| 1. |  |  |  |
| 2. |  |  |  |
| 3. |  |  |  |
|  |  |  |  |