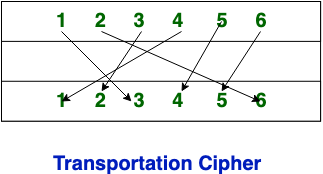
**Introduction**

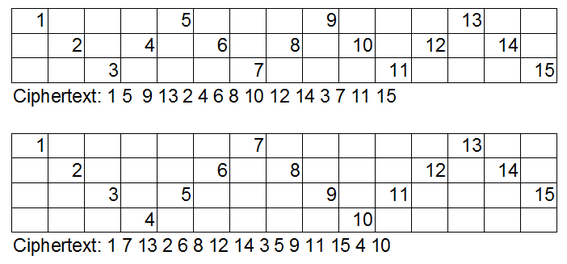
TRANSPOSITION CIPHER

Transposition Cipher Technique rearranges the position of the plain text’s characters.In transposition Cipher Technique, The position of the character is changed but character’s identity is not changed.



**ROW TRANSPOSITION**

The rail fence cipher (also called a zigzag cipher) is a form of transposition cipher. It derives its name from the way in which it is encoded.



**ENCRYPTION**

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.

**DECRYPTION**

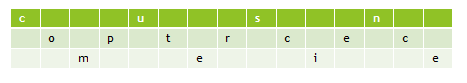
Hence, rail matrix can be constructed accordingly. Once we’ve got the matrix we can figure-out the spots where texts should be placed (using the same way of moving diagonally up and down alternatively ).

Then, we fill the cipher-text row wise. After filling it, we traverse the matrix in zig-zag manner to obtain the original text.

**EXAMPLE**

**Plain text:Computer science**

**Depth:3**

****

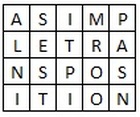
**Cipher text:cusnoptrcecmeie**

**COLUMNAR TRANSPOSITION**

Columnar Transposition involves writing the plaintext out in rows, and then reading the ciphertext off in columns.

In its simplest form, it is the Route Cipher where the route is to read down each column in order.

For example, the plaintext "a simple transposition" with 5 columns looks like the grid below



**ENCRYPTION**

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.

**EXAMPLE**

**Plain Text: Vignan university**

**Key:hack**



**Cipher text:inviguetvaisnnry**

**IMPLEMENTATION**

ROW TRANSPOSITION

#include<stdio.h>

#include<string.h>

#include<stdlib.h>

main()

{

int i,j,len,rails,count,code[100][1000];

char str[1000];

printf("Enter a Secret Message\n");

gets(str);

len=strlen(str);

printf("Enter number of rails\n");

scanf("%d",&rails);

for(i=0;i<rails;i++)

{

for(j=0;j<len;j++)

{

code[i][j]=0;

}

}

count=0;

j=0;

while(j<len)

{

if(count%2==0)

{

for(i=0;i<rails;i++)

{

code[i][j]=(int)str[j];

j++;

}

}

else

{

for(i=rails-2;i>0;i--)

{

code[i][j]=(int)str[j];

j++;

}

}

count++;

}

for(i=0;i<rails;i++)

{

for(j=0;j<len;j++)

{

if(code[i][j]!=0)

printf("%c",code[i][j]);

}

}

printf("\n");

}

**COLUMN TRANSPOSITION**

#include<bits/stdc++.h>

using namespace std;

// Key for Columnar Transposition

string key;

map<int,int> keyMap;

void setPermutationOrder()

{

// Add the permutation order into map

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

{

keyMap[key[i]] = i;

}

}

// Encryption

string encryptMessage(string msg)

{

int row,col,j;

string cipher = "";

/\* calculate column of the matrix\*/

col = key.length();

/\* calculate Maximum row of the matrix\*/

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')

{

/\* Adding the padding character '\_' \*/

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

j++;

}

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

{

/\* Adding only space and alphabet into matrix\*/

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

j++;

}

k++;

}

}

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

{

j=ii->second;

// getting cipher text from matrix column wise using permuted key

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

{

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

cipher += matrix[i][j];

}

}

return cipher;

}

// Decryption

string decryptMessage(string cipher)

{

/\* calculate row and column for cipher Matrix \*/

int col = key.length();

int row = cipher.length()/col;

char cipherMat[row][col];

/\* add character into matrix column wise \*/

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

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

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

/\* update the order of key for decryption \*/

int index = 0;

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

ii->second = index++;

/\* Arrange the matrix column wise according

to permutation order by adding into new matrix \*/

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];

}

}

/\* getting Message using matrix \*/

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;

cin>>key;

setPermutationOrder();

string cipher = encryptMessage(msg);

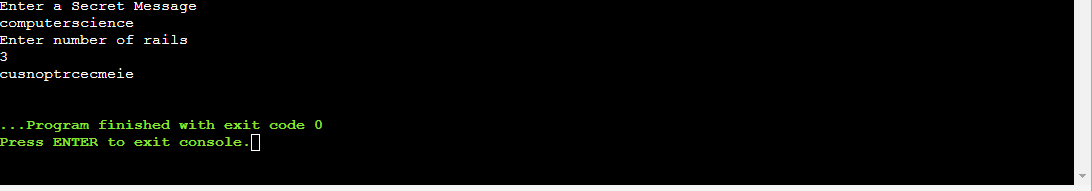
cout << "Encrypted Message: " << cipher << endl;

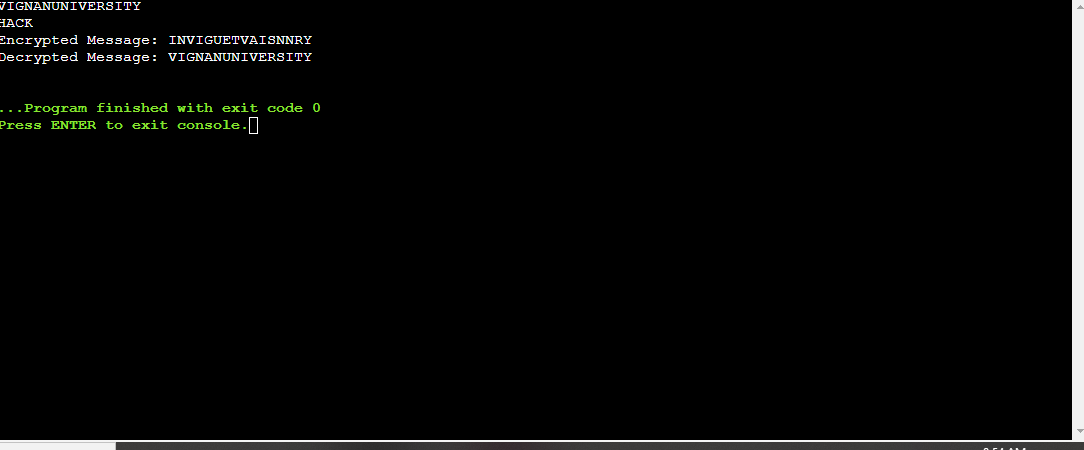
cout << "Decrypted Message: " << decryptMessage(cipher) << endl;

return 0;

}

**OUTPUT SCREENSHOTS**

****

****

**ADVANTAGES OF TRANSPOSITION**

* Has a better mixing of letters than some ciphers.
* The main benefit that transposition cipher have over substitution cipher is that transposition cipher can be applied more than once.(This is double Transposition).

**DISADVANTAGES OF TRANSPOSITION**

* Can easily be deciphered if it was a small message.

**REQUIREMENTS**

SOFTWARE REQUIREMENTS:

C compiler

HARDWARE REQUIREMENTS:

Processor:Intel(R) Core(TM) i5-8265U CPU @ 1.60GHz, 1800 Mhz, 4 Core(s), 8 Logical Processor(s)

RAM:8 GB for 64-bit

ROM:1TB for 64-bit

**REFERENCES**

* www.geeksforgeeks.org/columnar-transposition-cipher

**CONCLUSION**

Encryption of Plain text to cipher text and Decryption of cipher text to plain text is done using row and column transpositions.