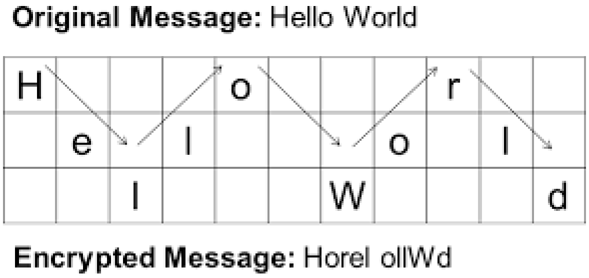
# Experiment No: 3

**Aim:** To study and implementation of Rail Fence Cipher.

## Introduction:

The Rail Fence Cipher is also called as zigzag cipher. It is a form of a transposition cipher. I.e., the alphabets in the plaintext are jumbled. In the rail fence cipher, the plaintext is written downwards diagonally on successive "rails" of an imaginary fence, then moving up when the bottom rail is reached, down again when the top rail is reached, and so on until the whole plaintext is written out. The ciphertext is then read off in rows.



**Original Message:** Hello World

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| H | | | | o | | | | r | | |
| | e | | l | | | | o | | l | |
| | | l | | | | W | | | | d |

**Encrypted Message:** Horel ollWd

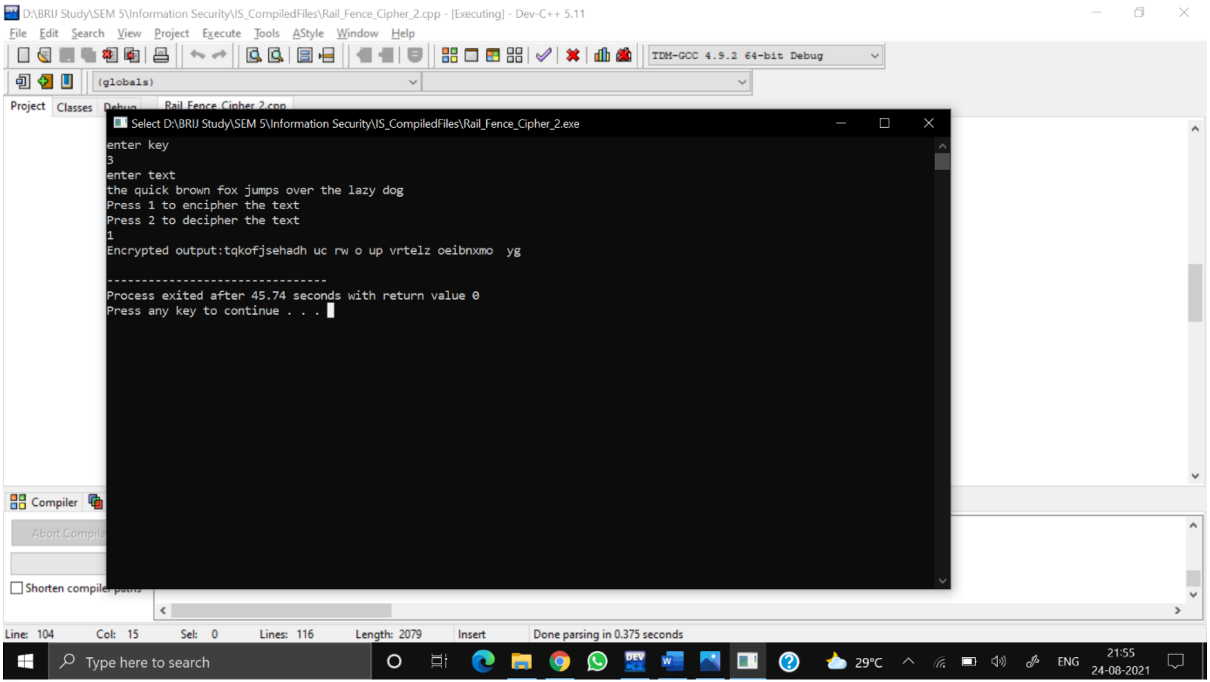
This is how Rail Fence Cipher encryption can be represented. While in decryption encrypted text is enter in a matrix by following zigzag pattern.

## Program (Source Code):

```
#include<iostream>
#include<sstream>
#include<string>
using namespace std;


string encry(int key, string t)
{
   string enc_s = "";
   int col = t.length();
   char rf[key][col];

   for(int i=0; i<key; i++)
   {
      for(int j=0; j<col; j++)
      {
         rf[i][j] = 0;
      }
   }

   for(int j=0,i=0,dir=1; j<col; i+=dir, j++)
   {
      rf[i][j] = t[j];
      if(i==0){dir=1;}
      else if(i==key-1){dir=-1;}
```

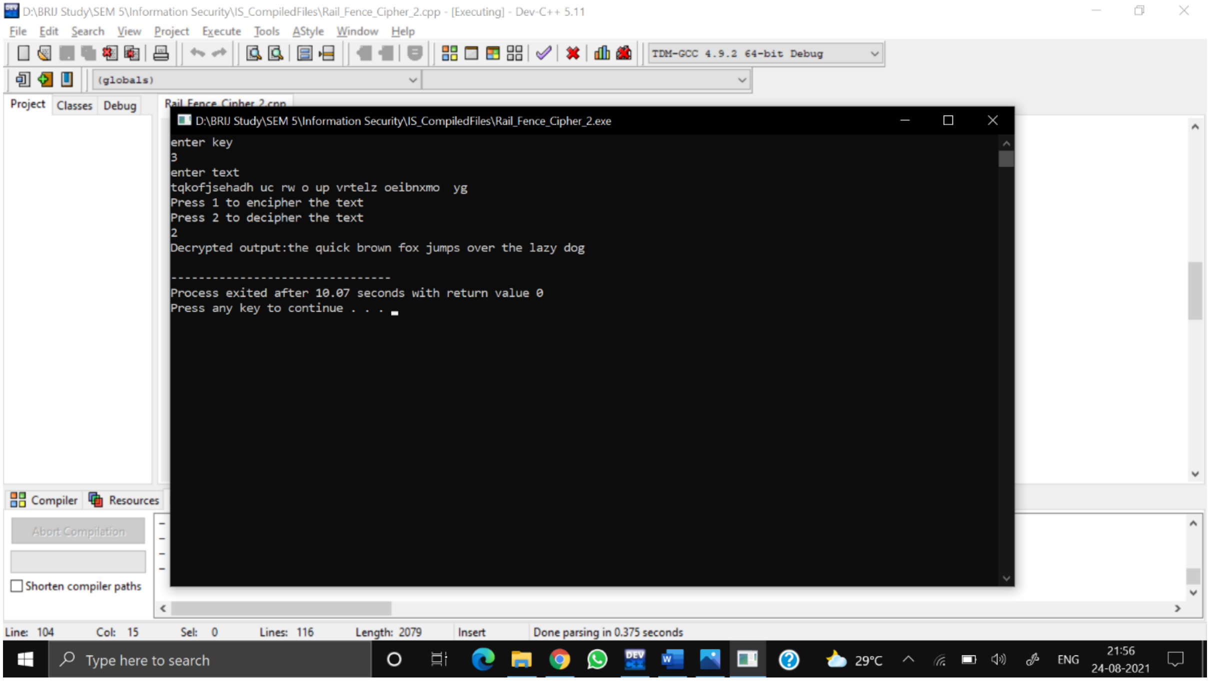
```
    }

    for(int i=0; i<key; i++)
    {
        for(int j=0; j<col; j++)
        {
            if(rf[i][j]!=0)
            {
                enc_s += rf[i][j];
            }
        }
    }
    return enc_s;
}

string decry(int key, string t)
{
    string dec_s = "";
    int col = t.length();
    char rf[key][col];

    for(int i=0; i<key; i++)
    {
        for(int j=0; j<col; j++)
        {
            rf[i][j] = 0;
        }
    }

    for(int j=0,i=0,dir=1; j<col; i+=dir, j++)
    {
        rf[i][j] = '$';
        if(i==0){dir=1;}
        else if(i==key-1){dir=-1;}
    }

    int z=0;
    for(int i=0; i<key; i++)
    {
        for(int j=0; j<col; j++)
        {
            if(rf[i][j] == '$')
            {
                rf[i][j] = t[z++];
            }
        }
    }

    for(int j=0,i=0,dir=1; j<col; i+=dir,j++)
    {
        dec_s += rf[i][j];
        if(i==0){dir=1;}
        else if(i==key-1){dir=-1;}
    }

    return dec_s;
}

int main()
{
    int key;
    string t;
    cout<<"enter key"<<endl;
    cin>>key;

    cout<<"enter text"<<endl;
    getline(cin>>ws, t);

    int choice;
    cout<<"Press 1 to encipher the text"<<endl;
    cout<<"Press 2 to decipher the text"<<endl;
    cin>>choice;

    switch(choice)
    {
    case 1:
        cout<<"Encrypted output:"<<encry(key, t)<<endl;
        break;

    case 2:
        cout<<"Decrypted output:"<<decry(key, t)<<endl;
        break;

    default:
        cout<<"Try again next time"<<endl;
    }

    return 0;
}
```

## Output (Program):



D:\BRIJ Study\SEM 5\Information Security\IS_CompiledFiles\Rail_Fence_Cipher_2.cpp - [Executing] - Dev-C++ 5.11
Select D:\BRIJ Study\SEM 5\Information Security\IS_CompiledFiles\Rail_Fence_Cipher_2.exe
Line: 104 Col: 15 Sel: 0 Lines: 116 Length: 2079 Insert Done parsing in 0.375 seconds

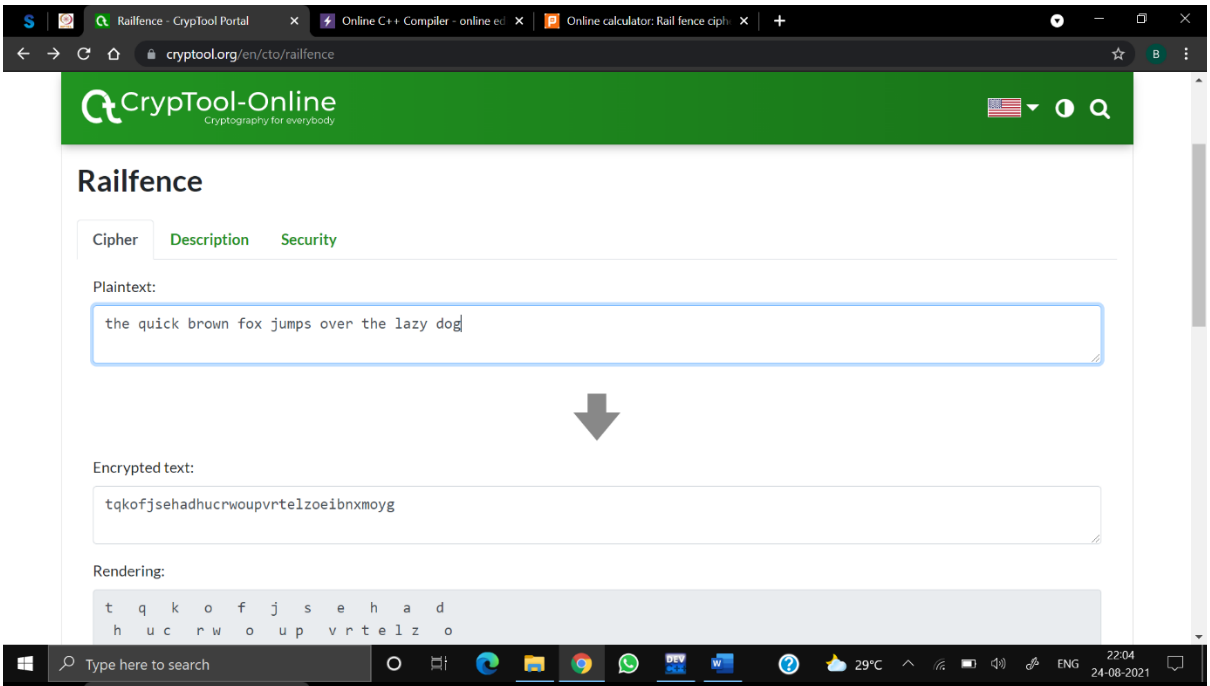
```
enter key
3
enter text
the quick brown fox jumps over the lazy dog
Press 1 to encipher the text
Press 2 to decipher the text
1
Encrypted output:tqkofjsehadh uc rw o up vrtelz oeibnxmo  yg

--------------------------------
Process exited after 45.74 seconds with return value 0
Press any key to continue . . .
```



D:\BRIJ Study\SEM 5\Information Security\IS_CompiledFiles\Rail_Fence_Cipher_2.cpp - [Executing] - Dev-C++ 5.11
D:\BRIJ Study\SEM 5\Information Security\IS_CompiledFiles\Rail_Fence_Cipher_2.exe
Line: 104 Col: 15 Sel: 0 Lines: 116 Length: 2079 Insert Done parsing in 0.375 seconds

```
enter key
3
enter text
tqkofjsehadh uc rw o up vrtelz oeibnxmo  yg
Press 1 to encipher the text
Press 2 to decipher the text
2
Decrypted output:the quick brown fox jumps over the lazy dog

--------------------------------
Process exited after 10.07 seconds with return value 0
Press any key to continue . . .
```

## Output (Cryptool):


Railfence - CrypTool Portal
Online C++ Compiler - online ed
Online calculator: Rail fence ciph
cryptool.org/en/cto/railfence
CrypTool-Online
Cryptography for everybody
Type here to search
29°C
ENG
22:04
24-08-2021

Railfence

Cipher | Description | Security

Plaintext:

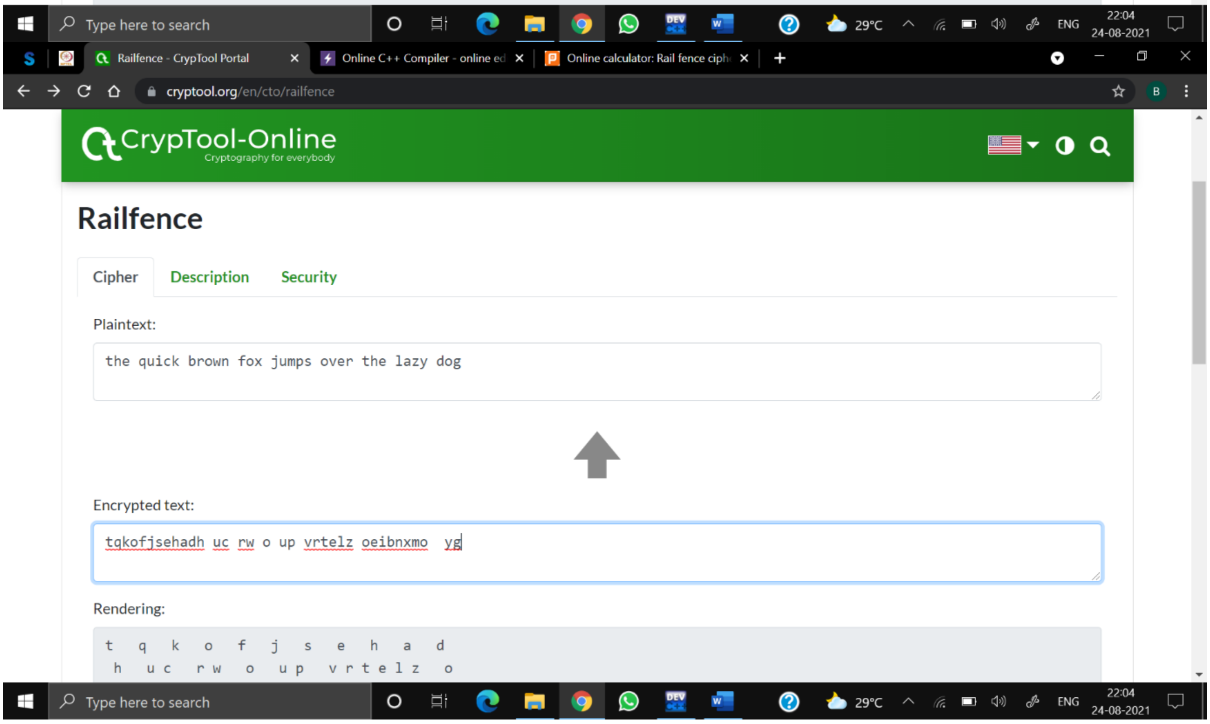
```
the quick brown fox jumps over the lazy dog
```

Encrypted text:

```
tqkofjsehadhucrwoupvrtelzoeibnxmoyg
```

Rendering:

```
t   q   k   o   f   j   s   e   h   a   d
 h   u c   r w   o   u p   v r t e l z   o
```


Railfence - CrypTool Portal
Online C++ Compiler - online ed
Online calculator: Rail fence ciph
cryptool.org/en/cto/railfence
CrypTool-Online
Cryptography for everybody
Type here to search
29°C
ENG
22:04
24-08-2021

Railfence

Cipher | Description | Security

Plaintext:

```
the quick brown fox jumps over the lazy dog
```

Encrypted text:

```
tqkofjsehadh uc rw o up vrtelz oeibnxmo  yg
```

Rendering:

```
t   q   k   o   f   j   s   e   h   a   d
 h   u c   r w   o   u p   v r t e l z   o
```

## Cryptanalysis:

The cipher's key is N, the number of rails. If N is known, the cipher text can be decrypted by using the above algorithm. Values of N equal to or greater than L, the length of the cipher text, are not usable, since then the ciphertext is the same as the plaintext. Therefore, the number of usable keys is low, allowing the brute-force attack of trying all possible keys. As a result, the rail-fence cipher is considered weak.

## Applications:

The Rail Fence Cipher was invented in ancient times. It was used by the Greeks, who created a special tool, called scytale, to make message encryption and decryption easier. Currently, it is usually used with a piece of paper. The letters are arranged in a way which is similar to the shape of the top edge of the rail fence.

## References:

1. https://www.geeksforgeeks.org/rail-fence-cipher-encryption-decryption/
2. https://en.wikipedia.org/wiki/Rail_fence_cipher
3. https://www.google.com/url?sa=i&url=https%3A%2F%2Fwww.researchgate.net%2Ffigure%2FEncrypting-using-Rail-Fence-Cipher5_fig5_333480277&psig=AOvVaw2PJYxSQqiFt00IKjl9dnMw&ust=1630074480696000&source=images&cd=vfe&ved=0CAsQjRxqFwoTCMDT0ufyzvICFQAAAAAdAAAAABAD