Cryptography and Network Security

Name: Piyush Mhaske
Batch: B3
PRN : 2019BTECS00089

Assignment 6

Objective:

Rail Fence

Theory:

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

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

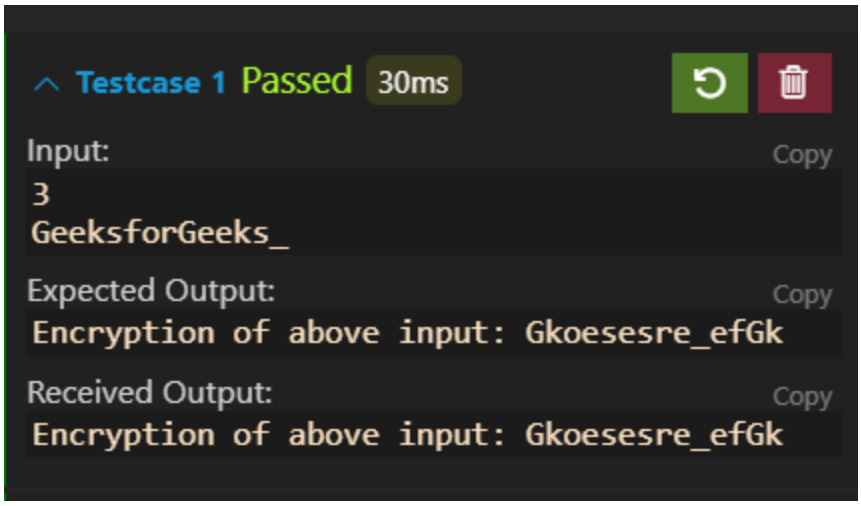
- In the rail fence cipher, the plain-text is written downwards and diagonally on successive rails of an imaginary fence.
- When we reach the bottom rail, we traverse upwards moving diagonally, after reaching the top rail, the direction is changed again. Thus the alphabets of the message are written in a zig-zag manner.
- After each alphabet has been written, the individual rows are combined to obtain the cipher-text.

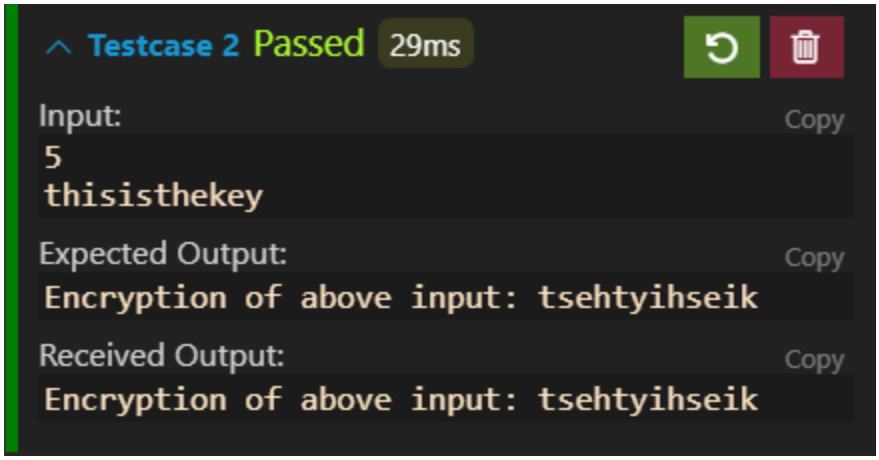
Code:

```
//code by :- Piyush Mhaske
#include <bits/stdc++.h>
#define ll long long
#define ul unsigned long long
#define pb emplace_back
#define po pop_back
#define vi vector<ll>
#define vii vector<vector<ll>>
using namespace std;
void file(){
     ios_base::sync_with_stdio(false);
```

```
     cin.tie(NULL);}
ll M = 1e9 + 7;
string RailFence(int key, string input){
     int n = input.length();
    vector<vector<char>> rail(key,vector<char>(n,'#'));
   int depth=0;
   bool good=true;
    for(int i=0;i<n;i++){
            // cout<<depth;
            rail[depth][i] = input[i];
             if(good){
                depth++;
            }else{
                depth--;
            }
             if(depth==0){
                good=true;
            }else if(depth==key-1){
                good=false;
            }
    }


    string ans="";

    for(int i=0;i<key;i++){
        for(int j=0;j<n;j++){
            cout<<rail[i][j];
            if(rail[i][j]!='#')
            ans += rail[i][j];
        }
        cout<<"\n";
    }

    return ans;

}
int main()
{   file();
    int key;
    string input;
    cin>>key>>input;

    string ans = RailFence(key,input);
    cout<<"Encryption of above input: ";
    cout<<ans<<"\n";

    return 0;
}
```

Output:


Testcase 1 Passed 30ms
Input:
Copy
3
GeeksforGeeks_
Expected Output:
Copy
Encryption of above input: Gkoesesre_efGk
Received Output:
Copy
Encryption of above input: Gkoesesre_efGk


Testcase 2 Passed 29ms
Input:
Copy
5
thisisthekey
Expected Output:
Copy
Encryption of above input: tsehtyihseik
Received Output:
Copy
Encryption of above input: tsehtyihseik

Conclusion: