Cryptography and Network Security

Name: Piyush Mhaske Batch: B3

Assignment 5

PRN: 2019BTECS00089

Objective: Columnar Transposition

Theory:

The Columnar Transposition Cipher is a form of transposition cipher just like Rail Fence Cipher. Columnar Transposition involves writing the plaintext out in rows, and then reading the ciphertext off in columns one by one.

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 Decrypt(string CipherText, string key){
    unordered_map<int, vector<char>> mp;
    int n = key.size();
    int m = CipherText.size();
    string ans="";
    int col = m/n;
    int rem = m%n;
    vector<int> len(key.size(),col);
    if(rem!=0){
         for(int i=0;i<n;i++){</pre>
            if(rem>0){
                len[i] = col+1;
```

Assignment 5

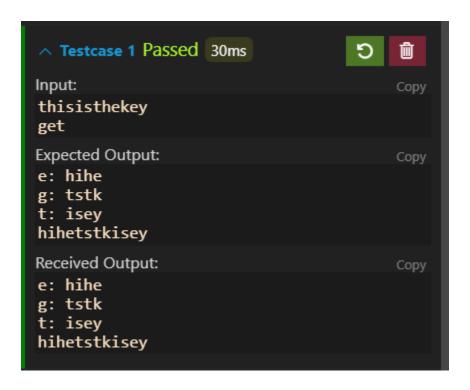
```
}else{
                 len[i] = col;
            }
             rem--;
    }else{
          for(int i=0;i<n;i++)</pre>
         len[i] = col;
   vector<pair<char,int>> temp;
   for(int i=0;i<n;i++){</pre>
    temp.push_back({key[i],i});
   }
   sort(temp.begin(),temp.end());
    int i=0;
    for(auto x:temp){
        while(len[x.second]--){
             // cout<<len[x.second];</pre>
        mp[x.second].push_back(CipherText[i]);
        i++;
        }
    }
     for(int i=0;i<key.size();i++){</pre>
        cout<<key[i]<<": ";
        for(auto x: mp[i]){
             cout<<x<<" ";
        cout<<"\n";
     }
    int j=0, k=0;
    while(j<CipherText.size()){</pre>
        for(int i=0;i<key.size();i++){</pre>
            // cout<<key[i]<<": ";
             ans+=mp[i][k];
        }
        k++;
        j++;
    }
    return ans;
}
string Columnar(string PlainText, string key){
    string ans;
    unordered_map<char, vector<char>> mp;
    vector<pair<char,int>> temp;
    for(int i=0;i<key.size();i++){</pre>
```

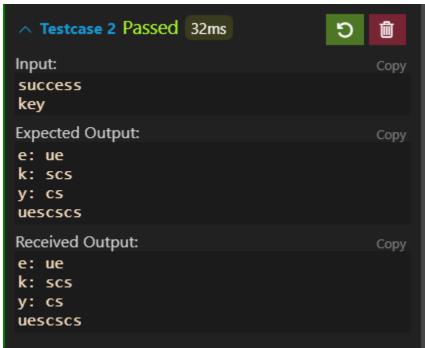
Assignment 5 2

```
temp.push_back({key[i],i});
    }
    sort(temp.begin(),temp.end());
    int j=0;
    int m = key.size();
    for(int i=0;i<PlainText.size();i++){</pre>
        mp[j].push_back(PlainText[i]);
        j = (j+1)%m;
    }
    for(auto x:temp){
        if(mp.count(x.second)){
            cout<<x.first<<": ";</pre>
            for(auto y:mp[x.second]){
                 cout<<y;
                 ans+=y;
            }
            cout<<"\n";
        }
    }
    return ans;
}
int main()
  file();
    string PlainText, CipherText, key;
    cin>>PlainText>>key;
    CipherText = Columnar(PlainText, key);
    cout<<"The Encrypted text is: "<<CipherText<<"\n";</pre>
    cout<<"Decrytping\n";</pre>
    string decrypt = Decrypt(CipherText, key);
    decrypt = decrypt.substr(0,CipherText.size());
    cout<<"The decrypt text is : "<<decrypt<<"\n";</pre>
    return 0;
}
```

Output:

Assignment 5





Conclusion:

Easy to Crack the message can be predicted if it is small.

Assignment 5 4