

Name: Rutikesh Sawant
Batch: B2
Subject: CNS Lab
PRN: 2019BTECS00034

Assignment 6

Aim: To encrypt given plain text using columnar cipher

Theory:

Columnar cipher is a type of transposition cipher. Columnar cipher involves writing the plaintext in rows and columns and then reading columns one by one.

Code:

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

string format(string &str) {
    stringstream res;
    for (auto ch : str) {
        if (ch != ' ') {
            res << (char)tolower(ch);
        }
    }
    return res.str();
}

string encrypt(string &plain, string &key) {
    map<char, vector<char>> mp;
    int counter = 0;
```

```

    for (int i = 0; i < plain.size(); i++) {
        mp[key[counter++]].push_back(plain[i]);

        if (counter == key.size())
            counter = 0;
    }

    stringstream cipher;
    for (auto it : mp) {
        for (int i = 0; i < it.second.size(); i++) {
            cipher << it.second[i];
        }
    }
    return cipher.str();
}

string decrypt(string &cipher, string &key) {

    map<int, int> map1;

    int common = cipher.size() / key.size();
    int extra = cipher.size() % key.size();

    for (int i = 0; i < key.size(); i++) {
        if (i < extra)
            map1[i] = common + 1;
        else
            map1[i] = common;
    }

    map<int, vector<char>> map2;

```

```

    int start = 0;

    string sortedKey(key);
    sort(sortedKey.begin(), sortedKey.end());

    for (int i = 0; i < sortedKey.size(); i++) {
        for (int j = 0; j < key.size(); j++) {
            if (sortedKey[i] == key[j]) {
                for (int k = 0; k < map1[j]; k++) {
map2[key[j]].push_back(cipher[start++]);
                }
            }
        }
    }

    string plain;

    vector<int> counters(key.size(), 0);

    int i = 0;

    while (plain.size() < cipher.size()) {
        for (int i = 0; i < key.size(); i++) {
            if (counters[i] < map1[i])
                plain += map2[key[i]][counters[i]++];
        }
    }
    return plain;
}

int main() {
    int choice;

```

```

    cout << "1. Encrypt\n2. Decrypt\nEnter your choice:
";
    cin >> choice;
    cin.get();

    if (choice == 1) {
        string plain, key;
        cout << "\nEnter plain text: ";
        getline(cin, plain);
        plain = format(plain);
        cout << "\nEnter key: ";
        getline(cin, key);
        format(key);
        string cipher = encrypt(plain, key);
        cout << "\nEncrypted text is : " << cipher <<
endl;
    } else if (choice == 2) {
        string cipher, key;
        cout << "\nEnter cipher text: ";
        getline(cin, cipher);
        cipher = format(cipher);
        cout << "\nEnter key: ";
        getline(cin, key);
        format(key);
        string plain = decrypt(cipher, key);
        cout << "\nDecrypted text is : " << plain <<
endl;
    }

    return 0;
}

```

Output:

```
Rutikesh@Rutikesh MINGW64 ~/Desktop/FY I/C&NS Lab/Assignment 6
$ g++ columnar.cpp

Rutikesh@Rutikesh MINGW64 ~/Desktop/FY I/C&NS Lab/Assignment 6
$ ./a.exe
1. Encrypt
2. Decrypt
Enter your choice: 1

Enter plain text: fghji

Enter key:  klmn

Encrypted text is : fighj

Rutikesh@Rutikesh MINGW64 ~/Desktop/FY I/C&NS Lab/Assignment 6
$ ./a.exe
1. Encrypt
2. Decrypt
Enter your choice: 2

Enter cipher text: fghji

Enter key: klmn
$ ./a.exe
1. Encrypt
2. Decrypt
Enter your choice: 2

Enter cipher text: usirethk

Enter key: gamer

Decrypted text is : rutikesh

Rutikesh@Rutikesh MINGW64 ~/Desktop/FY I/C&NS Lab/Assignment 6
```