Name : Patel Pratham A.
Roll No : CE100
ID : 20CEUOS052
Lab : 4

…..PlayFair…..

```
#include <bits/stdc++.h>
using namespace std;
string encryption(string text, char key[5][5])
{
    int k = 0;
    int s = 0;
    bool check = false;
    string ciphertext;
    vector<pair<int, int>> vec1;
    while (s != (text.length()))
    {
        for (int i = 0; i < 5; i++)
        {
            for (int j = 0; j < 5; j++)
            {
                if (k != (text.length()))
                {
                    if (key[i][j] == text[k])
```

```
            {
                vec1.push_back(make_pair(i, j));
                k++;
                break;
                check = true;
            }
            else
            {
                // cout << "key:" << key[i][j] << "  "
                //      << "text:" << text[k] << endl;
            }
        }
    }
    if (check)
    {
        break;
    }
}

s += 1;
// cout << "s:" << s << endl;
}
// for (int i = 0; i < vec1.size(); i++)
// {
//     cout << vec1[i].first << ", " << vec1[i].second << endl;
// }
// cout << vec1.size() << endl;

k = 0;
vector<pair<int, int>> ciphervec(vec1.size());
for (int i = 0; i < vec1.size(); i += 2)
{
    if (vec1[i].first == vec1[i + 1].first)
    {
        // cout<<"for i = "<<i<<" i am in if"<<endl;
        ciphervec[i].first = vec1[i].first;
        ciphervec[i + 1].first = vec1[i + 1].first;
        if (((vec1[i].second + 1) % 5) == 0)
        {
```

```
            ciphervec[i].second = (((vec1[i].second + 1) % 5));
        }
        else
        {

            ciphervec[i].second = (((vec1[i].second + 1) % 5));
        }
        if (((vec1[i + 1].second + 1) % 4) == 0)
        {
            ciphervec[i + 1].second = (((vec1[i + 1].second + 1) % 5));
        }
        else
        {
            ciphervec[i + 1].second = (((vec1[i + 1].second + 1) % 5));
        }

        // cout<<ciphervec[i].first<<","<<ciphervec[i].second<<endl;
        // cout<<ciphervec[i+1].first<<","<<ciphervec[i+1].second<<endl;
    }
    else if (vec1[i].second == vec1[i + 1].second)
    {
        // cout<<"for i = "<<i<<" i am in elseif"<<endl;
        ciphervec[i].second = vec1[i].second;
        ciphervec[i + 1].second = vec1[i + 1].second;
        if (((vec1[i].second + 1) % 5) == 0)
        {
            ciphervec[i].first = (((vec1[i].first + 1) % 5));
        }
        else
        {
            ciphervec[i].first = (((vec1[i].first + 1) % 5));
        }
        if (((vec1[i + 1].second + 1) % 5) == 0)
        {
            ciphervec[i + 1].first = (((vec1[i + 1].first + 1) % 5));
        }
        else
        {
            ciphervec[i + 1].first = (((vec1[i + 1].first + 1) % 5));
        }
```

```
            // cout<<ciphervec[i].first<<","<<ciphervec[i].second<<endl;
            // cout<<ciphervec[i+1].first<<","<<ciphervec[i+1].second<<endl;
        }
        else
        {
            // cout<<"for i = "<<i<<" i am in else"<<endl;
            ciphervec[i].first = vec1[i].first;
            ciphervec[i].second = vec1[i + 1].second;
            ciphervec[i + 1].first = vec1[i + 1].first;
            ciphervec[i + 1].second = vec1[i].second;
            // cout<<ciphervec[i].first<<","<<ciphervec[i].second<<endl;
            // cout<<ciphervec[i+1].first<<","<<ciphervec[i+1].second<<endl;
        }
        // cout << "i value is : " << i << endl;
    }
    // cout << "cipher vec size is : " << ciphervec.size() << endl;
    // for (int i = 0; i < ciphervec.size(); i++)
    // {
    //     cout << ciphervec[i].first << ", " << ciphervec[i].second << endl;
    // }
    for (int i = 0; i < ciphervec.size(); i++)
    {
        ciphertext += key[ciphervec[i].first][ciphervec[i].second];
    }
    return ciphertext;
}

string decription(string text, char key[5][5])
{
    int k = 0;
    int s = 0;
    bool check = false;
    string ciphertext;
    // vector<char> ciphertext(text.length());
    vector<pair<int, int>> vec1;
    while (s != (text.length()))
    {
        for (int i = 0; i < 5; i++)
        {
```

```
        for (int j = 0; j < 5; j++)
        {
            if (k != (text.length()))
            {
                if (key[i][j] == text[k])
                {
                    vec1.push_back(make_pair(i, j));
                    k++;
                    break;
                    check = true;
                }
                else
                {
                    // cout << "key:" << key[i][j] << "  "
                    //      << "text:" << text[k] << endl;
                }
            }
        }
        if (check)
        {
            break;
        }
    }

    s += 1;
    // cout << "s:" << s << endl;
}
// for (int i = 0; i < vec1.size(); i++)
// {
//     cout << vec1[i].first << ", " << vec1[i].second << endl;
// }
// cout << vec1.size() << endl;

k = 0;
vector<pair<int, int>> ciphervec(vec1.size());
for (int i = 0; i < vec1.size(); i += 2)
{
    if (vec1[i].first == vec1[i + 1].first)
    {
        // cout<<"for i = "<<i<<" i am in if"<<endl;
```

```
    ciphervec[i].first = vec1[i].first;
    ciphervec[i + 1].first = vec1[i + 1].first;
    if (((vec1[i].second - 1) % 5) == -1)
    {

        ciphervec[i].second = ((vec1[i].second + 4));
    }
    else
    {

        ciphervec[i].second = (((vec1[i].second - 1) % 5));
    }
    if (((vec1[i + 1].second - 1) % 4) == -1)
    {
        ciphervec[i + 1].second = ((vec1[i + 1].second + 4));
    }
    else
    {
        ciphervec[i + 1].second = (((vec1[i + 1].second - 1) % 5));
    }

    // cout<<ciphervec[i].first<<","<<ciphervec[i].second<<endl;
    // cout<<ciphervec[i+1].first<<","<<ciphervec[i+1].second<<endl;
}
else if (vec1[i].second == vec1[i + 1].second)
{
    // cout<<"for i = "<<i<<" i am in elseif"<<endl;
    ciphervec[i].second = vec1[i].second;
    ciphervec[i + 1].second = vec1[i + 1].second;
    if (((vec1[i].second + 1) % 5) == 0)
    {
        ciphervec[i].first = (((vec1[i].first - 1) % 5));
    }
    else
    {
        ciphervec[i].first = (((vec1[i].first - 1) % 5));
    }
    if (((vec1[i + 1].second + 1) % 5) == 0)
    {
        ciphervec[i + 1].first = (((vec1[i + 1].first - 1) % 5));
```

```
            }
            else
            {
                ciphervec[i + 1].first = (((vec1[i + 1].first - 1) % 5));
            }

            // cout<<ciphervec[i].first<<","<<ciphervec[i].second<<endl;
            // cout<<ciphervec[i+1].first<<","<<ciphervec[i+1].second<<endl;
        }
        else
        {
            // cout<<"for i = "<<i<<" i am in else"<<endl;
            ciphervec[i].first = vec1[i].first;
            ciphervec[i].second = vec1[i + 1].second;
            ciphervec[i + 1].first = vec1[i + 1].first;
            ciphervec[i + 1].second = vec1[i].second;
            // cout<<ciphervec[i].first<<","<<ciphervec[i].second<<endl;
            // cout<<ciphervec[i+1].first<<","<<ciphervec[i+1].second<<endl;
        }
        // cout << "i value is : " << i << endl;
    }
    // cout << "cipher vec size is : " << ciphervec.size() << endl;
    // for (int i = 0; i < ciphervec.size(); i++)
    // {
    //     cout << ciphervec[i].first << ", " << ciphervec[i].second << endl;
    // }
    for (int i = 0; i < ciphervec.size(); i++)
    {
        ciphertext += key[ciphervec[i].first][ciphervec[i].second];
    }
    return ciphertext;
}

int main()
{
    string key1 = "MONARCHY";
    string text;
    int k = 0;
    vector<char> ciphertextfinal;
    vector<char> decrypttextfinal;
```

```
    char key[5][5] = {{'M', 'O', 'N', 'A', 'R'}, {'C', 'H', 'Y', 'B', 'D'}, {'E', 'F', 'G', 'I', 'K'}, {'L', 'P',
'Q', 'S', 'T'}, {'U', 'V', 'W', 'X', 'Z'}};
    cout << "Enter Some String:";
    cin >> text;
    string str = "X";
    bool check = true;
    vector<pair<char, char>> vec;

    // cout << text << endl;
    // cout << text.length() << endl;
    for (int i = 0; i < (text.length()); i += 2)
    {
        // cout << "i am in pair" << endl;
        if (text[i] != text[i + 1])
        {
            // cout << text[i] << text[i + 1] << endl;
            vec.push_back(make_pair(text[i], text[i + 1]));
            // cout << i << endl;
        }
        else if (text[i] == text[i + 1])
        {
            // cout << "i am in else if" << endl;
            text.insert((i + 1), str);
            // cout << text[i] << text[i + 1] << endl;
            vec.push_back(make_pair(text[i], text[i + 1]));
            // cout << i << endl;
            check = false;
        }
    }
    if ((text.length() % 2) != 0 && check)
    {
        // cout << "i am in length" << endl;
        text += 'X';
    }
    string cipher;
    cipher = encryption(text, key);

    // for(int i=0;i<ciphertextfinal.size();i++){
    //     cipher[i]=ciphertextfinal[i];
```

```
    // }
    cout << "Encrypt text is :" << cipher << endl;
    string plain;
    plain = decription(cipher, key);

    // for(int i=0;i<ciphertextfinal.size();i++){
    //     plain[i]=ciphertextfinal[i];
    // }
    cout << "Decrypt text is :" << plain << endl;
}
```

…..AutoKey…..

```
#include <bits/stdc++.h>
using namespace std;
int main()
{
    char alphabet[26] = {'a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j', 'k', 'l', 'm', 'n', 'o', 'p', 'q', 'r', 's', 't',
'u', 'v', 'w', 'x', 'y', 'z'};
    string text;
    cout << "Enter Some Text : ";
    cin >> text;
    int key;
    cout << "Enter a Key : ";
    cin >> key;
    vector<int> map(text.length());
    vector<int> cipher(text.length());
    vector<int> pain(text.length());
    for (int i = 0; i < text.length(); i++)
    {
        map[i] = text[i] - 'a';
    }
    for (int i = 0; i < text.length(); i++)
    {
        if (i == 0)
        {
            cipher[i] = (map[i] + key) % 26;
```

```
        }
        else
        {
            cipher[i] = (map[i] + map[i - 1]) % 26;
        }
    }
    string cipher_text;
    for (int i = 0; i < text.length(); i++)
    {
        cipher_text += alphabet[cipher[i]];
    }
    cout << "Cipher Text Is : " << cipher_text << endl;
    for (int i = 0; i < text.length(); i++)
    {
        if (i == 0)
        {
            pain[i] = (cipher[i] - key) % 26;
            // cout<<pain[i]<<endl;
            if (pain[i] < 0)
            {
                pain[i] = 26 + pain[i];
            }
        }
        else
        {
            // cout<<cipher[i]<<"-"<<pain[i-1]<<endl;
            pain[i] = (cipher[i] - pain[i - 1]) % 26;
            if (pain[i] < 0)
            {
                pain[i] = 26 + pain[i];
            }
        }
    }
    string plain_text;
    for (int i = 0; i < text.length(); i++)
    {
        plain_text += alphabet[pain[i]];
    }
    cout << "Plain Text Is : " << plain_text << endl;
}
```