

SEM - VII - 2022-23

CNS Lab

B3 - 2019BTECS00094 - Sweety Shrawan Gupta

Assignment 3

Playfair Cipher

Playfair Cipher :

The Playfair cipher encryption technique can be used to encrypt or encode a message. It operates exactly like typical encryption. The only difference is that it encrypts a digraph, or a pair of two letters, as opposed to a single letter.

An initial 5×5 matrix key table is created. The plaintext encryption key is made out of the matrix's alphabetic characters. Be mindful that you shouldn't repeat the letters. There are 26 alphabets, however, there are only 25 spaces in which we can place a letter. The matrix will delete the extra letter because there is an excess of one letter (typically J). Despite this, J is there in the plaintext before being changed to I.

Rules for Encryption:

- **If both the letters are in the same column:** Take the letter below each one (going back to the top if at the bottom).
- **If both the letters are in the same row:** Take the letter to the right of each one (going back to the leftmost if at the rightmost position).
- **If neither of the above rules is true:** Form a rectangle with the two letters and take the letters on the horizontal opposite corner of the rectangle.

Code:

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

int main()
{
    string pt, x, k, ct;
    getline(cin, pt);
    for (int i = 0; i < pt.length(); i++)
        if (pt[i] != ' ')
            x += pt[i];
    pt = x;

    cin >> k;
    cout << "Plain text is: " << pt << endl;
    cout << "key is: " << k << endl;

    map<char, pair<int, int>> char_index;
    vector<vector<char>> mat(5, vector<char>(5, '#')));
    int i = 0, j = 0, l = 0;

    while (i < 5 and l < k.size()) {
        if (k[l] == 'j') k[l] = 'i';
        if (char_index.find(k[l]) == char_index.end()) {
            mat[i][j] = k[l];
            char_index[k[l]] = {i, j};
            j++;
            if (j == 5) {
                i++; j = 0;
            }
        }
        l++;
    }

    for (char ch = 'a'; i < 5 and ch <= 'z'; ch++) {
        if (ch == 'j') ch++;
        if (char_index.find(ch) == char_index.end()) {
```

```

        mat[i][j] = ch;
        char_index[ch] = {i, j};
        j++;
        if (j == 5) {
            i++; j = 0;
        }
    }
}

cout << "\nPlayfair Matrix is: \n";
for (int ii = 0; ii < 5; ii++) {
    for (int jj = 0; jj < 5; jj++) {
        cout << mat[ii][jj] << " ";
    }
    cout << "\n";
}

for (int i = 1; i < pt.size(); i++) {
    if (i % 2 and pt[i - 1] == pt[i]) {
        pt.insert(i, "x");
    }
}

if (pt.size() % 2) pt.push_back('z');

cout << "\n\nPlain text after decryption is: " << pt;

for (int i = 1; i < pt.size(); i += 2) {
    int fi, fj, si, sj;
    fi = char_index[pt[i - 1]].first;
    fj = char_index[pt[i - 1]].second;
    si = char_index[pt[i]].first;
    sj = char_index[pt[i]].second;
    if (si == fi) {
        sj++;
        fj++;
        if (fj == 5) fj = 0;
        if (sj == 5) sj = 0;
        ct.push_back(mat[fi][fj]);
        ct.push_back(mat[si][sj]);
    } else if (sj == fj) {
        si++;

```

```

        fi++;
        if (fi == 5) fi = 0;
        if (si == 5) si = 0;
        ct.push_back(mat[fi][fj]);
        ct.push_back(mat[si][sj]);
    } else {
        ct.push_back(mat[fi][sj]);
        ct.push_back(mat[si][fj]);
    }
}

cout << "\nCipher text is: " << ct;

pt = "";

for (int i = 1; i < ct.size(); i += 2) {
    int fi, fj, si, sj;
    fi = char_index[ct[i - 1]].first;
    fj = char_index[ct[i - 1]].second;
    si = char_index[ct[i]].first;
    sj = char_index[ct[i]].second;
    if (si == fi) {
        sj--;
        fj--;
        if (fj == -1) fj = 4;
        if (sj == -1) sj = 4;
        pt.push_back(mat[fi][fj]);
        pt.push_back(mat[si][sj]);
    } else if (sj == fj) {
        si--;
        fi--;
        if (fi == -1) fi = 4;
        if (si == -1) si = 4;
        pt.push_back(mat[fi][fj]);
        pt.push_back(mat[si][sj]);
    } else {
        pt.push_back(mat[fi][sj]);
        pt.push_back(mat[si][fj]);
    }
}
}

```

```

if (pt.back() == 'z')pt.pop_back();
for (int i = 2; i < pt.size(); i++) {
    if (pt[i - 2] == pt[i] and pt[i - 1] == 'x') {
        pt.erase(i - 1, 1);
    }
}

cout << "\n\nPlain text after decryption is: " << pt;

return 0;
}

```

Output:

The screenshot shows a code editor with two files: `inputf.in` and `outputf.in`.

`inputf.in` contains:

```

1  sweety shrawan gupta
2  priya

```

`outputf.in` contains:

```

1  Plain text is: sweetyshrawangupta
2  key is: priya
3
4  Playfair Matrix is:
5  p r i y a
6  b c d e f
7  g h k l m
8  n o q s t
9  u v w x z
10
11  Cipher text is: qxlyfsexocizptnpanfa
12
13  Plain text after decription is: swexetyshrawangupta

```