SEM - VII - 2022-23

CNS Lab

B3 - 2019BTECS00094 - Sweety Shrawan Gupta

Assignment 5

Tranposition Ciphers

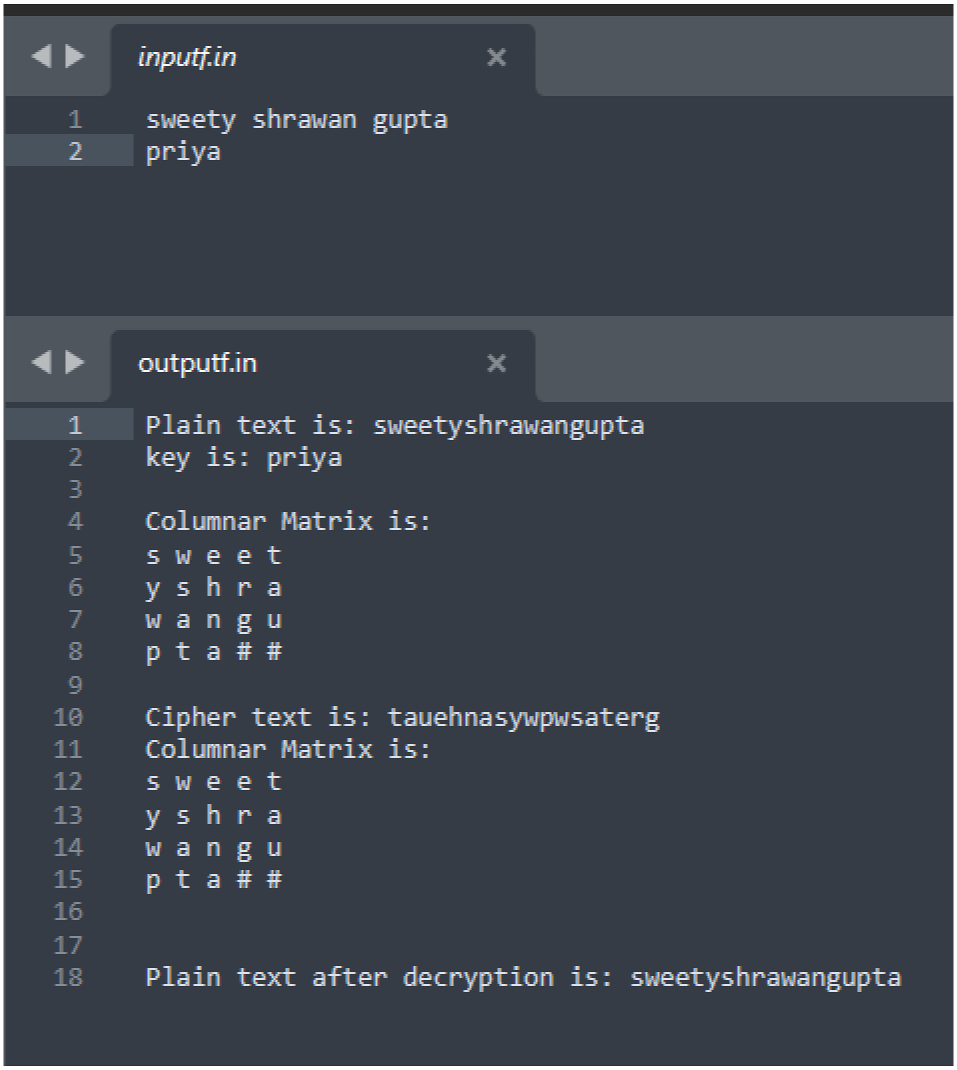
a)Columnar Cipher

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

Code:

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

int main()
{
    string pt, x, k, ct;
    getline(cin, pt);
    //removing the spaces
    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;
    int m = k.size(); //no. of columns
    int n = pt.size() / m + (pt.size() % m != 0); //no. of rows
    x = k;
    sort(x.begin(), x.end()); //sorting the key
    map<char, int> ma;
    for (int ii = 0; ii < m; ii++) {
        ma[k[ii]] = ii;
    }

//making a matrix and filling a plain text char row-wise
    vector<vector<char>> mat(n, vector<char>(m, '#'));
    int i = 0, j = 0, l = 0;
    while (i < n and l < pt.size()) {
        mat[i][j] = pt[l];
        j++;
        if (j == m) {
            i++; j = 0;
        }
        l++;
    }

    cout << "\nColumnar Matrix is: \n";
    for (int ii = 0; ii < n; ii++) {
        for (int jj = 0; jj < m; jj++) {
            cout << mat[ii][jj] << " ";
        }
        cout << "\n";
    }
//getting a index of letter/(column no. to read first) using a sorted key
    for (auto ch : x) {
        int jj = ma[ch];
        for (int ii = 0; ii < n and mat[ii][jj] != '#'; ii++) {
            ct += mat[ii][jj];
        }
    }

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

    int rem = ct.size() % m; //finding no. of column field to be filled in
last row
    pt = "";
    for (int ii = 0; ii < n; ii++) {
        for (int jj = 0; jj < m; jj++) {
            mat[ii][jj] = '#';
        }
    }
    l = 0;
    for (auto ch : x) {
        int jj = ma[ch];
        int cnt = (jj < rem) ? n : n - 1; //finding no. of row to be filled of that column no.
        for (int ii = 0; ii < n and cnt>0 and l < ct.size(); ii++) {
            mat[ii][jj] = ct[l];
            cnt--;
            l++;
        }
    }

    cout << "\nColumnar Matrix is: \n";
    for (int ii = 0; ii < n; ii++) {
        for (int jj = 0; jj < m; jj++) {
            if (mat[ii][jj] != '#')pt += mat[ii][jj];
            cout << mat[ii][jj] << " ";
        }
        cout << "\n";
    }

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

    return 0;
}
```

Output:



```
sweety shrawan gupta
priya
```

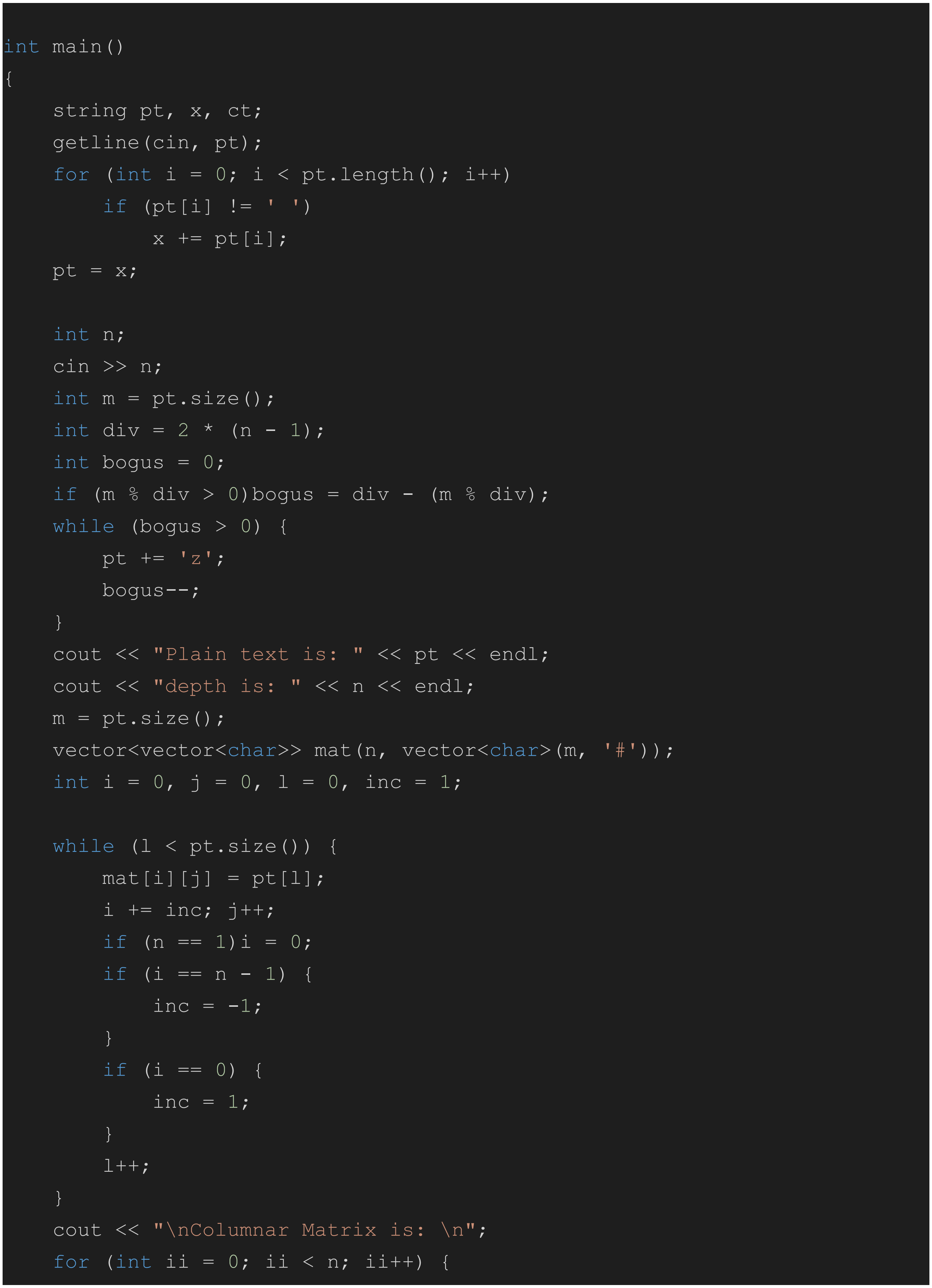
```
Plain text is: sweetyshrawangupta
key is: priya

Columnar Matrix is:
s w e e t
y s h r a
w a n g u
p t a # #

Cipher text is: tauehnasywpwsaterg
Columnar Matrix is:
s w e e t
y s h r a
w a n g u
p t a # #


Plain text after decryption is: sweetyshrawangupta
```

b)Railfence Cipher

- Plaintext is written downwards and diagonally on rails and then read as row-wise
- Each letter is written in a zigzag pattern.
- Keyless transposition cipher

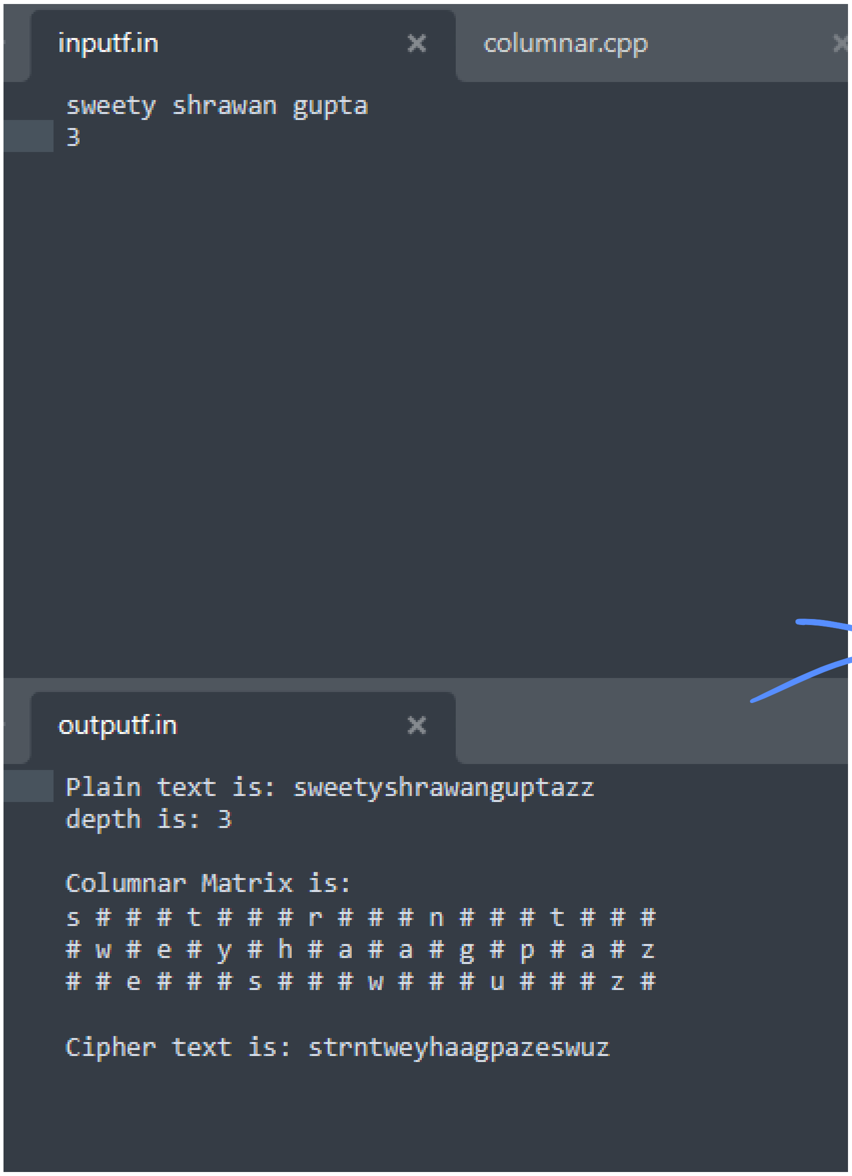
Code:

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



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

    int n;
    cin >> n;
    int m = pt.size();
    int div = 2 * (n - 1);
    int bogus = 0;
    if (m % div > 0)bogus = div - (m % div);
    while (bogus > 0) {
        pt += 'z';
        bogus--;
    }
    cout << "Plain text is: " << pt << endl;
    cout << "depth is: " << n << endl;
    m = pt.size();
    vector<vector<char>> mat(n, vector<char>(m, '#'));
    int i = 0, j = 0, l = 0, inc = 1;

    while (l < pt.size()) {
        mat[i][j] = pt[l];
        i += inc; j++;
        if (n == 1)i = 0;
        if (i == n - 1) {
            inc = -1;
        }
        if (i == 0) {
            inc = 1;
        }
        l++;
    }
    cout << "\nColumnar Matrix is: \n";
    for (int ii = 0; ii < n; ii++) {
        for (int jj = 0; jj < m; jj++) {
            if (mat[ii][jj] != '#')ct += mat[ii][jj];
            cout << mat[ii][jj] << " ";
        }
        cout << "\n";
    }

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


    return 0;
}
```

Output:



```
sweety shrawan gupta
3
```

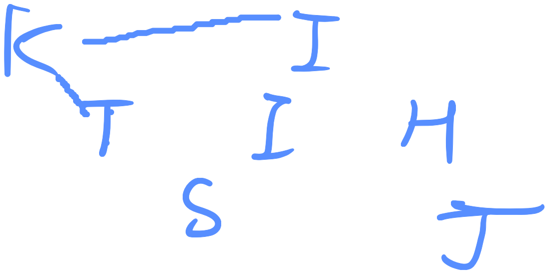
```
Plain text is: sweetyshrawanguptazz
depth is: 3

Columnar Matrix is:
s # # # t # # # r # # # n # # # t # # #
# w # e # y # h # a # a # g # p # a # z
# # e # # # s # # # w # # # u # # # z #

Cipher text is: strntweyhaagpazeswuz
```

Railfence

kshitij = 7
d=3

K				T		
	S		I		I	
		H				J

KTSIIHJ



depth = no. of row

length of pt = no. of column