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

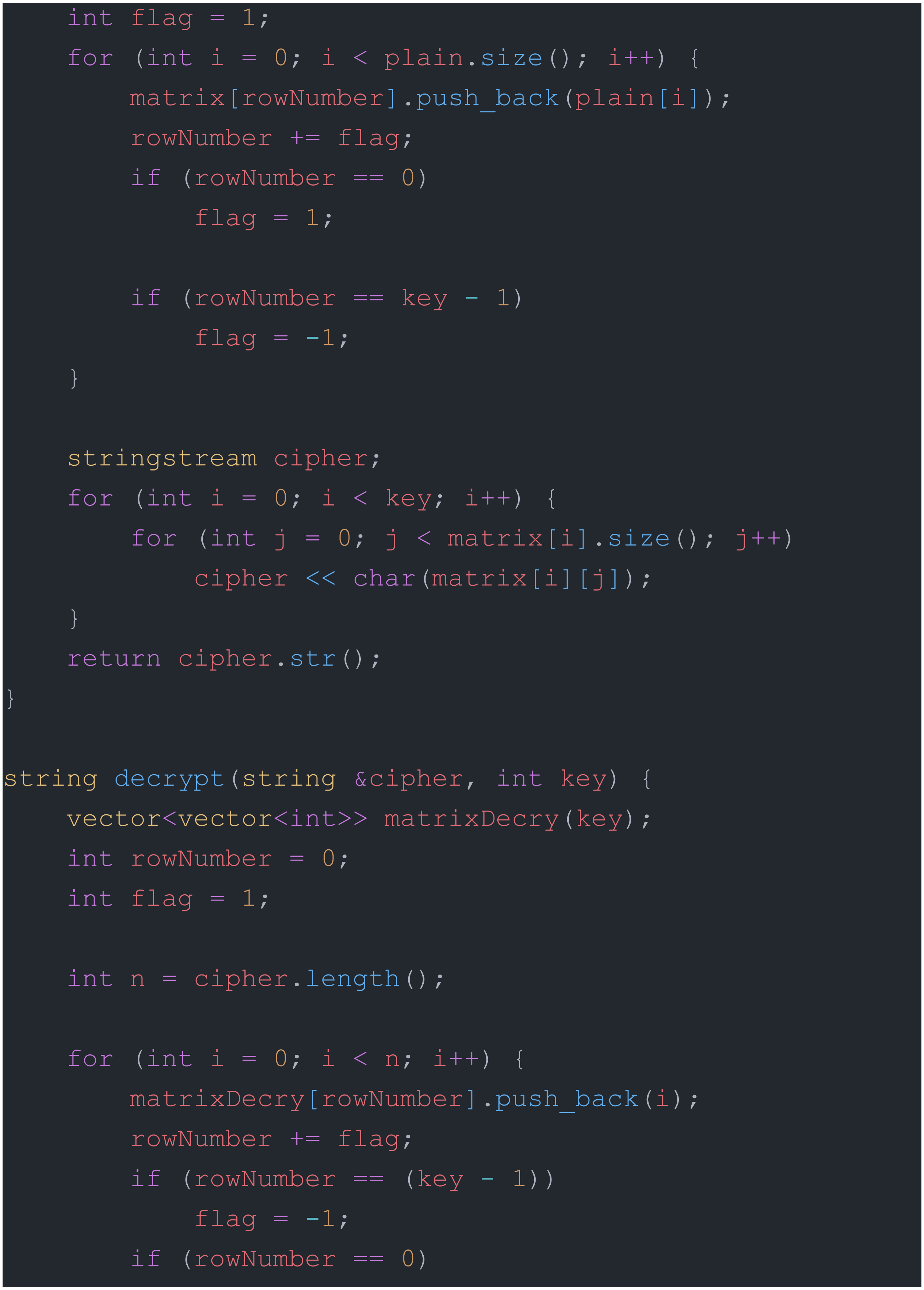
Assignment 5

Aim: To encrypt given plain text using railfence cipher.

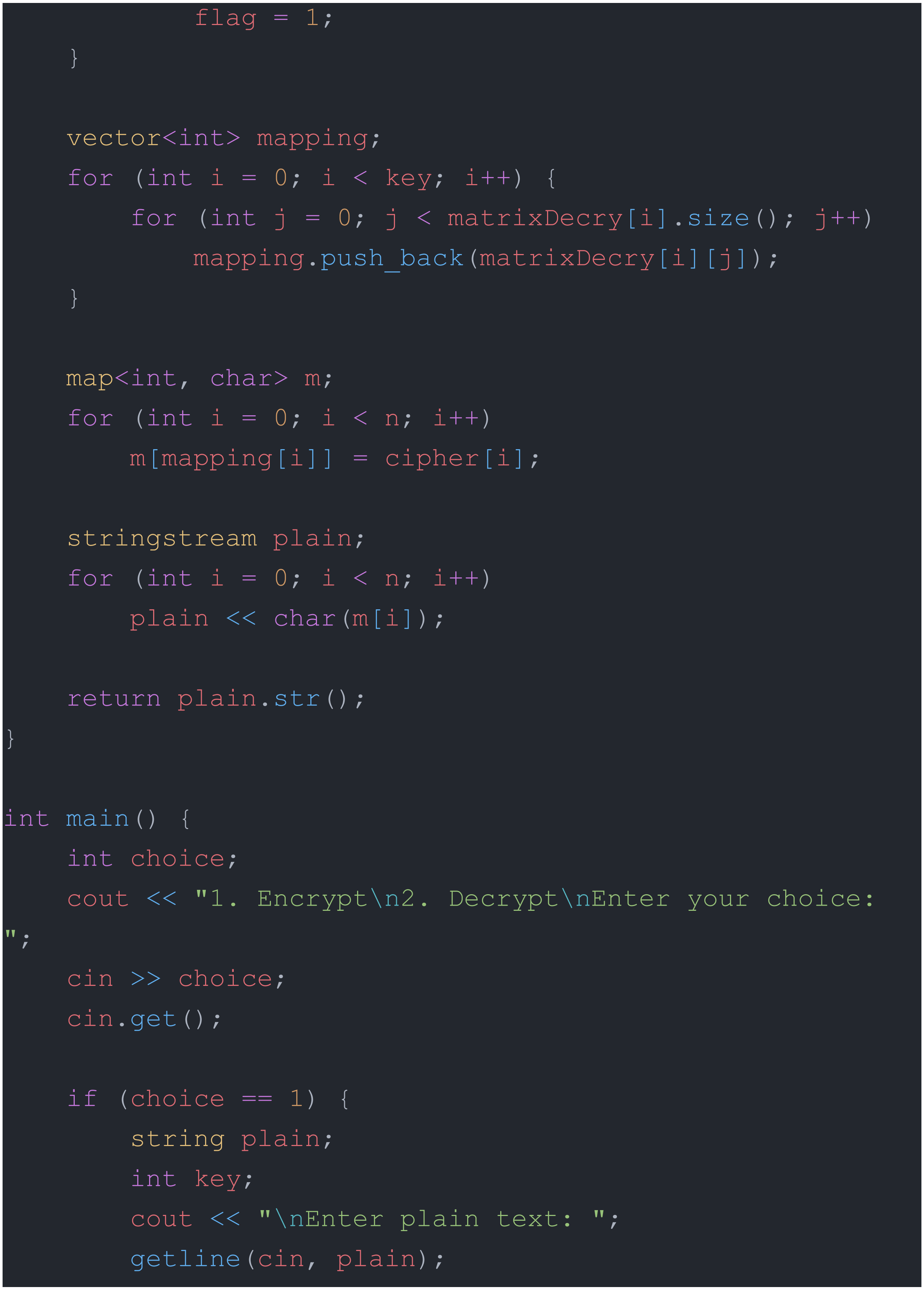
Theory:
Railfence cipher is a transposition type of cipher, It is also known as zigzag cipher. And its encryption algorithm is similer to fence of horizontal rails, so it is known as railfence cipher.

Code:

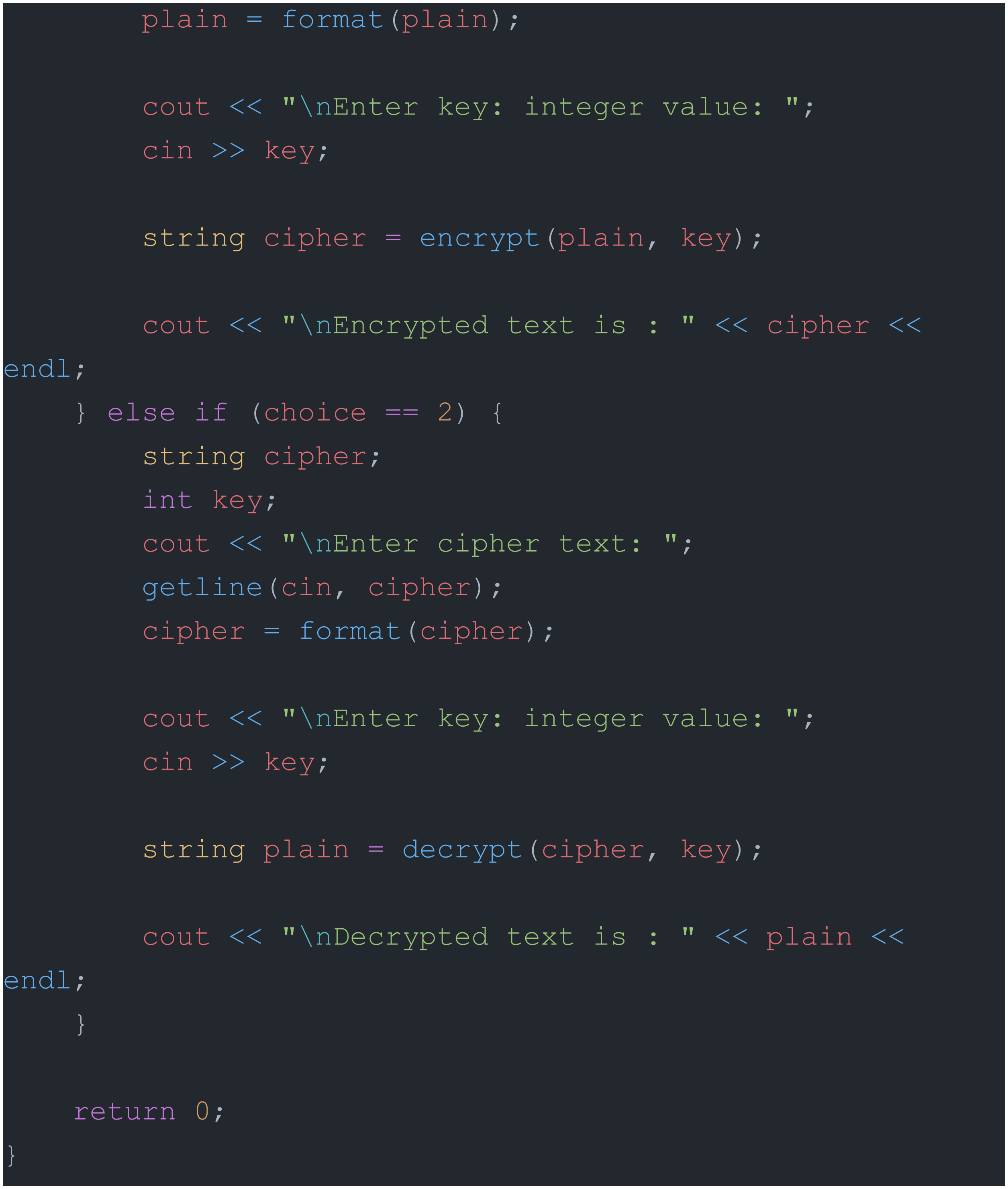
```
#include <bits/stdc++.h>
using namespace std;

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

string encrypt(string &plain, int key) {
    vector<vector<char>> matrix(key);

    int rowNumber = 0;
```



```cpp
    int flag = 1;
    for (int i = 0; i < plain.size(); i++) {
        matrix[rowNumber].push_back(plain[i]);
        rowNumber += flag;
        if (rowNumber == 0)
            flag = 1;

        if (rowNumber == key - 1)
            flag = -1;
    }

    stringstream cipher;
    for (int i = 0; i < key; i++) {
        for (int j = 0; j < matrix[i].size(); j++)
            cipher << char(matrix[i][j]);
    }
    return cipher.str();
}

string decrypt(string &cipher, int key) {
    vector<vector<int>> matrixDecry(key);
    int rowNumber = 0;
    int flag = 1;

    int n = cipher.length();

    for (int i = 0; i < n; i++) {
        matrixDecry[rowNumber].push_back(i);
        rowNumber += flag;
        if (rowNumber == (key - 1))
            flag = -1;
        if (rowNumber == 0)
```

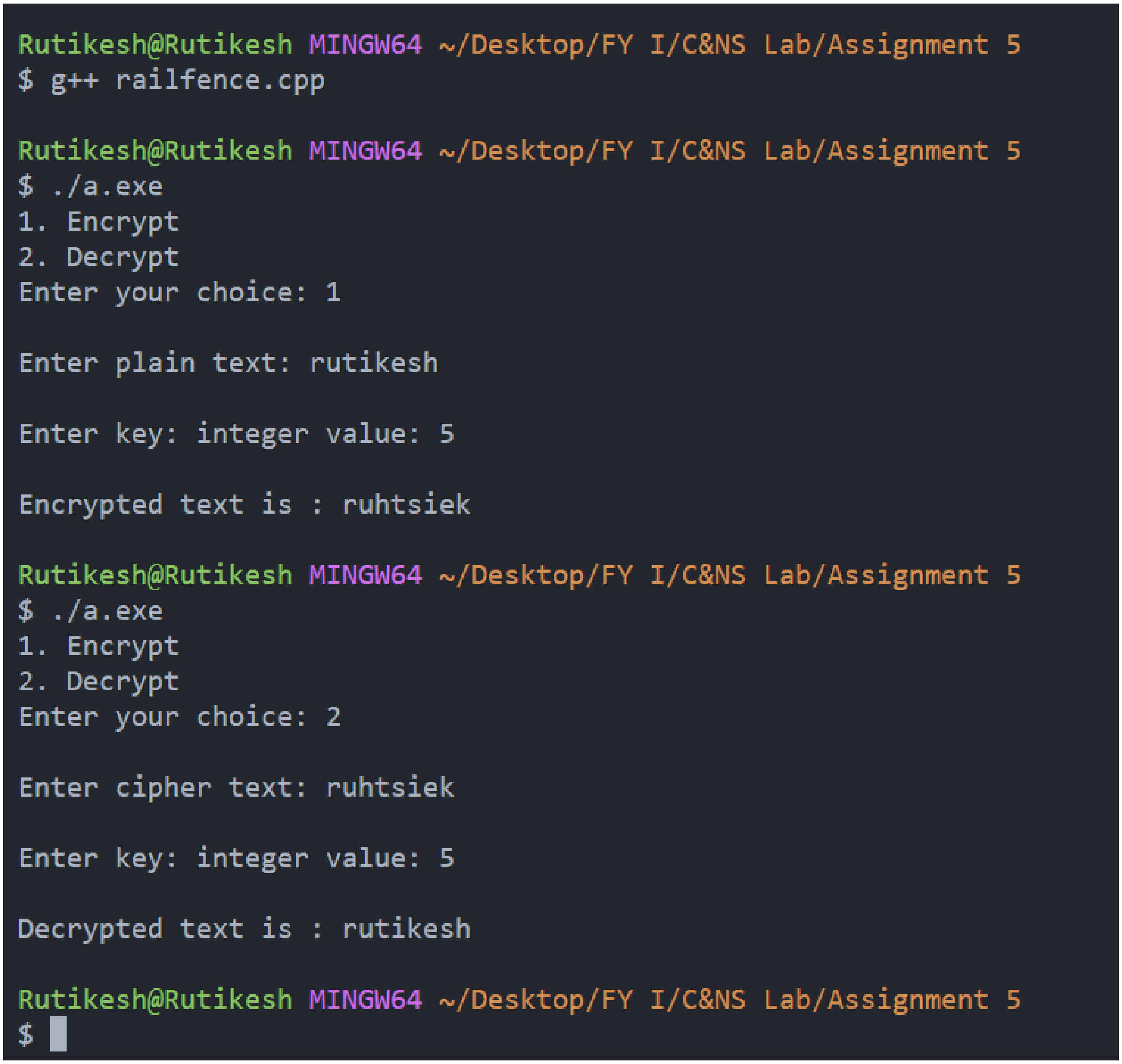


```
            flag = 1;
    }

    vector<int> mapping;
    for (int i = 0; i < key; i++) {
        for (int j = 0; j < matrixDecry[i].size(); j++)
            mapping.push_back(matrixDecry[i][j]);
    }

    map<int, char> m;
    for (int i = 0; i < n; i++)
        m[mapping[i]] = cipher[i];

    stringstream plain;
    for (int i = 0; i < n; i++)
        plain << char(m[i]);

    return plain.str();
}

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

    if (choice == 1) {
        string plain;
        int key;
        cout << "\nEnter plain text: ";
        getline(cin, plain);
```



```
        plain = format(plain);

        cout << "\nEnter key: integer value: ";
        cin >> key;

        string cipher = encrypt(plain, key);

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

        cout << "\nEnter key: integer value: ";
        cin >> 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 5
$ g++ railfence.cpp

Rutikesh@Rutikesh MINGW64 ~/Desktop/FY I/C&NS Lab/Assignment 5
$ ./a.exe
1. Encrypt
2. Decrypt
Enter your choice: 1

Enter plain text: rutikesh

Enter key: integer value: 5

Encrypted text is : ruhtsiek

Rutikesh@Rutikesh MINGW64 ~/Desktop/FY I/C&NS Lab/Assignment 5
$ ./a.exe
1. Encrypt
2. Decrypt
Enter your choice: 2

Enter cipher text: ruhtsiek

Enter key: integer value: 5

Decrypted text is : rutikesh

Rutikesh@Rutikesh MINGW64 ~/Desktop/FY I/C&NS Lab/Assignment 5
$ 
```