U18CO018

Shekhaliya Shubham

CNS

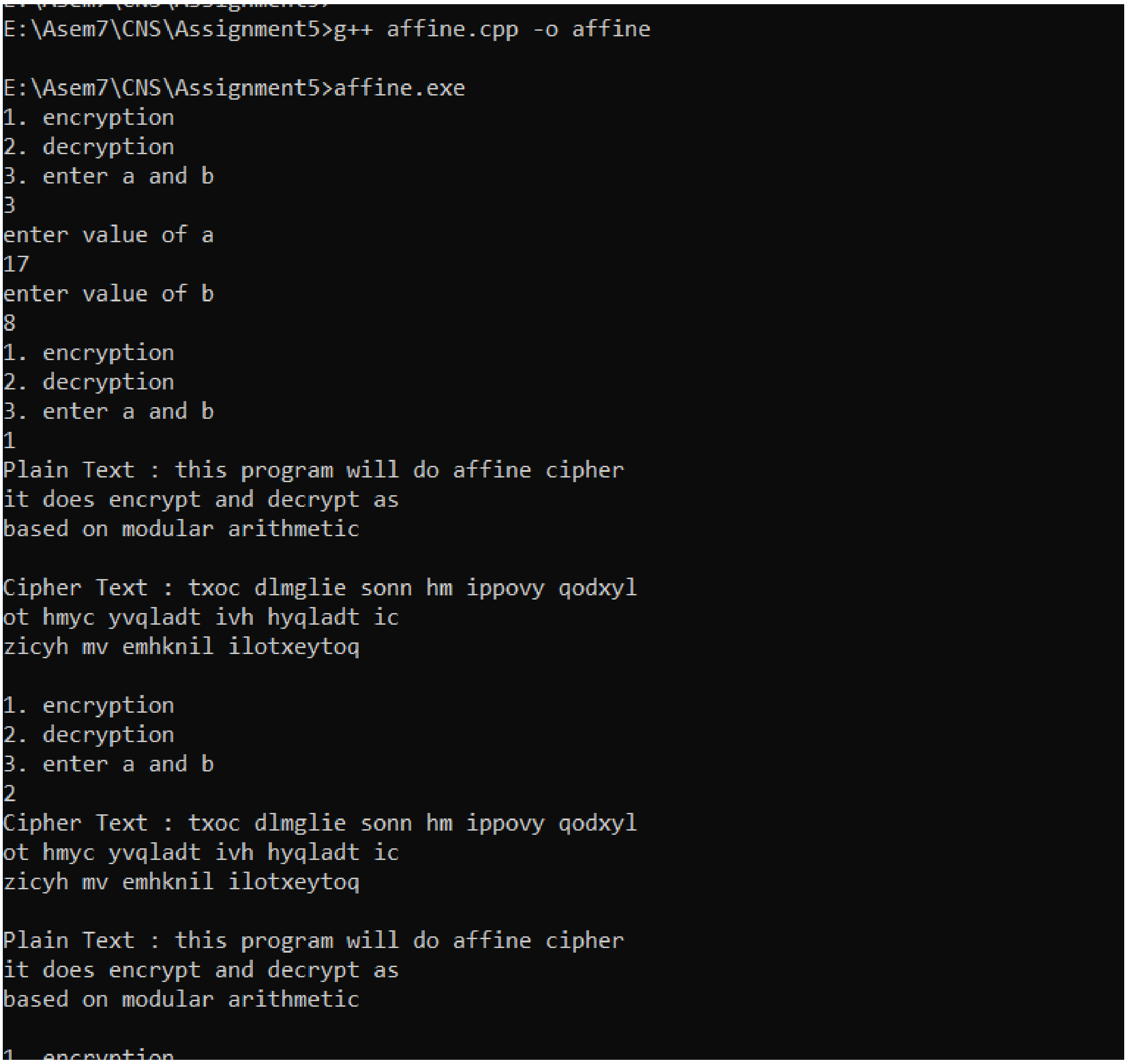
Lab Assignment 5

- Write a menu driven program with appropriate functions to implement the affine cipher i.e. E(x) = (a x + b) mod 26. Let the values of a and b be entered by the user. Your program must check for the feasibility of these values before encrypting the plaintext. The program must also output the decrypted values. Let the plaintext be input as a character array of defined size.

Code:

```
#include <bits/stdc++.h>

using namespace std;

string encrypt(string plainText, int a, int b) {
    string cipherText = "";
    int l = plainText.length();

    for(int i = 0;i<l;i++) {
        if(plainText[i]>='a' && plainText[i] <= 'z') {
            cipherText += ((a*(plainText[i] - 'a') + b)%26) + 'a';
        } else {
            cipherText += plainText[i];
        }
    }

    return cipherText;
}

int modInverse(int a, int m) {
    int m0 = m;
    int y = 0, x = 1;

    if (m == 1) {
        return 0;
    }

    while (a > 1) {
        int q = a / m;
        int t = m;

        m = a % m;
        a = t;

        t = y;

        y = x - q * y;
        x = t;
    }

    if (x < 0){
        x += m0;
    }

    return x;
}

string decrypt(string cipherText, int a, int b) {
    string plainText = "";
    int l = cipherText.length();

    int aInv = modInverse(a, 26);

    for(int i = 0; i < l;i++) {
        if(cipherText[i] >= 'a' && cipherText[i] <= 'z') {
            int c = (cipherText[i] - 'a' - b + 26)%26;

            plainText += ((aInv*c)%26) + 'a';
        } else {
            plainText += cipherText[i];
        }
    }

    return plainText;
}

string readFrom(string filename) {
    ifstream file;
    string input = "", result = "";
    file.open(filename);
    while (!file.eof()) {
        getline(file, input);
        result += input + "\n";
    }
    file.close();
    return result.substr(0, result.length() - 1);
```

```cpp
}

void writeTo(string filename, string message) {
    ofstream file;
    file.open(filename);
    file << message;
    file.close();
}

int main()
{
    bool run = true;
    int a  = 0, b = 0;

    while (run) {
        cout << "1. encryption\n2. decryption\n3. enter a and b\n";
        int ch;
        cin >> ch;

        if(ch == 1) {

            string plainText = readFrom("input.txt");
            string cipherText = encrypt(plainText, a, b);

            cout<<"Plain Text : "<<plainText<<endl<<endl;

            cout<<"Cipher Text : "<<cipherText<<endl<<endl;

            writeTo("output1.txt", cipherText);


        } else if (ch == 2) {

            string cipherText = readFrom("output1.txt");
            string plainText = decrypt(cipherText, a, b);

            cout<<"Cipher Text : "<<cipherText<<endl<<endl;
            cout<<"Plain Text : "<<plainText<<endl<<endl;

            writeTo("output2.txt", plainText);


        } else if (ch == 3) {

            while(__gcd(a, b) != 1) {
                cout<<"enter value of a\n";
                cin>>a;
                cout<<"enter value of b\n";
                cin>>b;
            }

        } else {
            break;
        }
    }
}
```



```
E:\Asem7\CNS\Assignment5>g++ affine.cpp -o affine

E:\Asem7\CNS\Assignment5>affine.exe
1. encryption
2. decryption
3. enter a and b
3
enter value of a
17
enter value of b
8
1. encryption
2. decryption
3. enter a and b
1
Plain Text : this program will do affine cipher
it does encrypt and decrypt as
based on modular arithmetic

Cipher Text : txoc dlmglie sonn hm ippovy qodxyl
ot hmyc yvqladt ivh hyqladt ic
zicyh mv emhknil ilotxeytoq

1. encryption
2. decryption
3. enter a and b
2
Cipher Text : txoc dlmglie sonn hm ippovy qodxyl
ot hmyc yvqladt ivh hyqladt ic
zicyh mv emhknil ilotxeytoq

Plain Text : this program will do affine cipher
it does encrypt and decrypt as
based on modular arithmetic

1. encryption
```

- Input.txt

```
this program will do affine cipher
it does encrypt and decrypt as
based on modular arithmetic
```

- output1.txt

```
txoc dlmglie sonn hm ippovy qodxyl
ot hmyc yvqladt ivh hyqladt ic
zicyh mv emhknil ilotxeytoq
```

- output2.txt

```
this program will do affine cipher
it does encrypt and decrypt as
based on modular arithmetic
```