A Project Report on Cryptography

By

Mercy Madhuri Kothapalli

K00369243

Professor: Dr. Habib Ammari

TEXAS A&M UNIVERSITY KINGSVILLE

TABLE OF CONTENTS

1. PROGRAM STATEMENT……………………………………………………………………......3
2. PROPOSED APPROCH…………………………………………………………………………………..........3
3. ENCRYPTION………………………………………………………………………………....….3
4. DECRYPTION……………………………………………………………………………..……...3
5. ALGORITHM……………………………………………………………………………..………3
6. PROGRAM………………………………………………………………………………..………4
7. OUTPUT SCREEN SHOTS………………………………………………………………………9

1. Problem Statement: The affine cipher is a substitution cipher, where a plaintext letter x is enciphered into a cipher text letter y as follows: x → y = α x + β (mod 26) The key for this encryption function is (α, β), where 0 ≤ α, β ≤ 25. There are 12 possible choices for α with gcd(α, 26) = 1, where gcd stands for the greatest common divisor, and 26 choices for β (i.e., 0, 1, 2, …, 25). Thus, there are 12 × 26 = 312 choices for the key. The decryption is accomplished by an affine function as follows: y → x = α –1(y – β) (mod 26) where αα –1 = 1 (mod 26).

2. Proposed Approach: In the Affine cipher, each letter in an alphabet is mapped to its numeric equivalent. It is a type of mono alphabetic substitution cipher. Encryption is done using a simple mathematical function and converted back to a letter. The letters of an alphabet of size m are first mapped to the integers in the range 0…….s-1, in then affine cipher. The key for the Affine cipher consists of 2 numbers, a and b( or alpha and beta respectively). α or a should be chosen to be relatively prime to m.

3. Encryption: To transform the integer, affine cipher uses modular arithmetic that each plain text letter corresponds into another integer that corresponds to a cipher text letter. The encryption function for a single letter is

E(x)= (ax+b) mod m. modulus m gives the size of the alphabet. (a,b) is the key of the cipher.

4. Decryption: In decryption, convert each of the cipher text letters into their integer values. The decryption function is D(x)= (1/a)(x-b)mod s where (1/a) is the modular multiplicative inverse of a modulo m. i.e, it satisfies the equation 1= (1/a) mod s.

5. Algorithm for Affine Cipher:

Begin

Function gcd(int x, int y)

{

// Everything divides 0

if (x == 0)

return y;

if (y == 0)

return x;

// base case

if (x == y)

return x;

// x is greater

if (x > y)

return gcd(x-y, y);

return gcd(x, y-x);

}

Begin

Function encryption(string m)

For i=0 to m.length()-1

if(m[i]!=’ ‘ )

c = c + (char) ((((a \* (m[i]- ‘A’) ) + b) % 26) + ‘A’)

else

c += m[i]

return c

end

Begin

Function decryption(string c)

Initialize a\_inverse = 0

Initialize flag = 0

For i = 0 to 25

Flag = (a\*i) % 26

If (flag == 1)

a\_inverse = i

done

done

For i = 0 to c.length() – 1

If(c[i]!=’ ’)

m = m + (char) (((a\_inverse \* ((c[i]+ ‘A’ – b)) % 26))

Else

m = m+ c[i]

done

end

6. Program: Program to calculate Affine Cipher.

#include<iostream>

#include<conio.h>

char op;

static int a;

static int b;

static int s1,s2;

using namespace std;

string encryption(string m1) {

//Cipher Text initially empty

string cipher1 = "";

for (int i = 0; i < m1.length(); i++) {

// Avoid space to be encrypted

if(m1[i]!=' ')

// added 'A' to bring it in range of ASCII alphabet [ 65-90 | A-Z ]

cipher1 = cipher1 + (char) ((((a \* (m1[i])- 'A' ) + b) % s1) + 'A');

}

return cipher1;

}

string decryption(string cipher2) {

string m2 = "";

int a\_inverse = 0;

int flag = 0;

//Find a^-1 (the multiplicative inverse of a

//in the group of integers modulo m.)

for (int i = 0; i < s2; i++) {

flag = (a \* i) % s2;

//Check if (a \* i) % 26 == 1,

//then i will be the multiplicative inverse of a

if (flag == 1) {

a\_inverse = i;

}

}

for (int i = 0; i < cipher2.length(); i++) {

if(cipher2[i] != ' ')

// added 'A' to bring it in range of ASCII alphabet [ 65-90 | A-Z ]

m2 = m2 + (char) (((a\_inverse \* (((cipher2[i]+ 'A') - b)) % s2)) + 'A');

else

//else append space character

m2 += cipher2[i];

}

return m2;

}

int gcd(int x, int y)

{

// Everything divides 0

if (x == 0)

return y;

if (y == 0)

return x;

// base case

if (x == y)

return x;

// x is greater

if (x > y)

return gcd(x-y, y);

return gcd(x, y-x);

}

int main(void) {

cout<<"\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"<<endl;

cout<<"WELCOME. This is a program to implement Affine Cipher. In affine cipher each letter in an alphabet"<<endl;

cout<<"is mapped to its numeric equivalent. Affine cipher is a type of monoalphabetic substitution cipher."<<endl;

cout<<"The letters of an alphabet of size s are mapped to the integers in the range 0,....s-1.The key is (a,b)."<<endl;

cout<<"a should be chosen to be relatively prime to s. Also values a and b lies between 1 and size of the alphabet."<<endl;

cout<<"To find the encryption of plain text enter the plain text and give the values of a and b"<<endl;

cout<<"To find the decryption, enter cipher text and values for a and b"<<endl;

cout<<"\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"<<endl;

do

{

cout<<"\n\*\*\*which operation you want to perform\*\*\*\n";

cout<<"press 0 for exit\n";

cout<<"press 1 for ENCRYPTION \n";

cout<<"press 2 for DECRYPTION\n";

cout<<"press 3 for OVERVIEW\n";

cout<<"press option:";

cin>>op;

string msg1, msg2, c1, c2;

switch(op)

{

case '1':

cout << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*ENCRYPTION OF PLAINTEXT\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"<<endl;

cout << "enter plain text in capital letters without spacing:\n";

cin >> msg1;

cout<< "The gcd(a,s) should be 1 where s is the size of the alphabet."<<endl;

do

{

cout<< "enter the size of alphabet"<<endl;

cin>>s1;

cout << "enter value of a"<<endl;

cin>>a;

}while(gcd(a,s1)!=1);

cout<<"The GCD of a="<<a<<" and size of the alphabet = "<<s1<<" is "<<gcd(a,s1)<<endl;

cout<<" Go ahead and give the value of b"<<endl;

do

{

cout<<"enter value of b where b>=0 and b < "<<s1<<endl;

cin>>b;

}while(b>=s1);

while(a==1 && b==0)

{

cout<<"(a,b) can't be (1,0). enter the values of a and b again"<<endl;

cin>>a;

cin>>b;

}

c1 = encryption(msg1);

cout << "Encrypted Message is : " << c1<<endl;

break;

case '2':

cout<< "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*DECRYPTION OF THE CIPHER TEXT\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"<<endl;

cout << "enter cipher text in capital letters without spacing:\n";

cin >> msg2;

cout<< “The gcd(a,s) should be 1 where s is size of the alphabet."<<endl;

do

{

cout<< "enter the size of alphabet"<<endl;

cin>>s2;

cout << "enter value of a"<<endl;

cin>>a;

}while(gcd(a,s2)!=1);

cout<<"The GCD of a="<<a<<" and size of the alphabet = "<<s2<<" is "<<gcd(a,s2)<<endl;

cout<<" Go ahead and give the value of b"<<endl;

do

{

cout<<"enter value of b where b>=0 and b < "<<s1<<endl;

cin>>b;

}while(b>=s2);

while(a==1 && b==0)

{

cout<<"(a,b) can't be (1,0). enter the values of a and b again"<<endl;

cin>>a;

cin>>b;

}

c2 = decryption(msg2);

cout << "decrypted Message is : " << c2<<endl;

break;

case '3':

cout<<"\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*OVERVIEW OF AFFINE CIPHER\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n";

cout<<" The affine cipher is a type of monoalphabetic substitution cipher, where each letter in an alphabet is mapped to \n";

cout<<" its numeric equivalent, encrypted using a simple mathematical function, and converted back to a letter.\n";

cout<<" The formula used means that each letter encrypts to one other letter, and back again, meaning the cipher is\n";

cout<<" essentially a standard substitution cipher with a rule governing which letter goes to which. As such, it has \n";

cout<<" the weaknesses of all substitution ciphers. Each letter is enciphered with the function (ax + b) mod 26, where\n";

cout<<" b is the magnitude of the shift.\n\n";

cout<<" DESCRIPTION: In the affine cipher the letters of an alphabet of size s are first mapped to the\n";

cout<<" integers in the range 0 … s − 1. It then uses modular arithmetic to transform the integer that each plaintext letter\n";

cout<<" corresponds to into another integer that correspond to a ciphertext letter. The encryption function for a single letter is\n";

cout<<" E(x)= (ax+b)mod s\n";

cout<<" where modulus s is the size of the alphabet and a and b are the keys of the cipher. The value a must be chosen such that\n";

cout<<" a and s are coprime. The decryption function is\n";

cout<<" D(x)= a^-1(y-b)mod s \n";

cout<<" where a^-1 is the modular multiplicative inverse of a modulo s. I.e., it satisfies the equation 1=a\*(a^-1)mod s\n";

cout<<" The multiplicative inverse of a only exists if a and s are co-prime. Hence without the restriction on a, decryption\n";

cout<<" might not be possible. It can be shown that decryption function is the inverse of encryption function\n";

cout<<" D(E(x))= a^-1(E(x)-b)mod s \n";

cout<<" = a^-1(((ax+b)mod s)-b)mod s \n";

cout<<" = a^-1(ax+b-b)mod s \n";

cout<<" = a^-1\*ax mod s \n";

cout<<" = x mod s. \n\n";

cout<<" WEAKNESS: The cipher's primary weakness comes from the fact that if the cryptanalyst can discover (by means of frequency\n";

cout<<" analysis, brute force, guessing or otherwise) the plaintext of two ciphertext characters then the key can be obtained by \n";

cout<<" solving a simultaneous equation. Since we know a and s are relatively prime this can be used to rapidly discard many false\n";

cout<<" keys in an automated system.\n";

break;

case '0':

exit(0);

default:

cout<<"invalid input"<<endl;

}

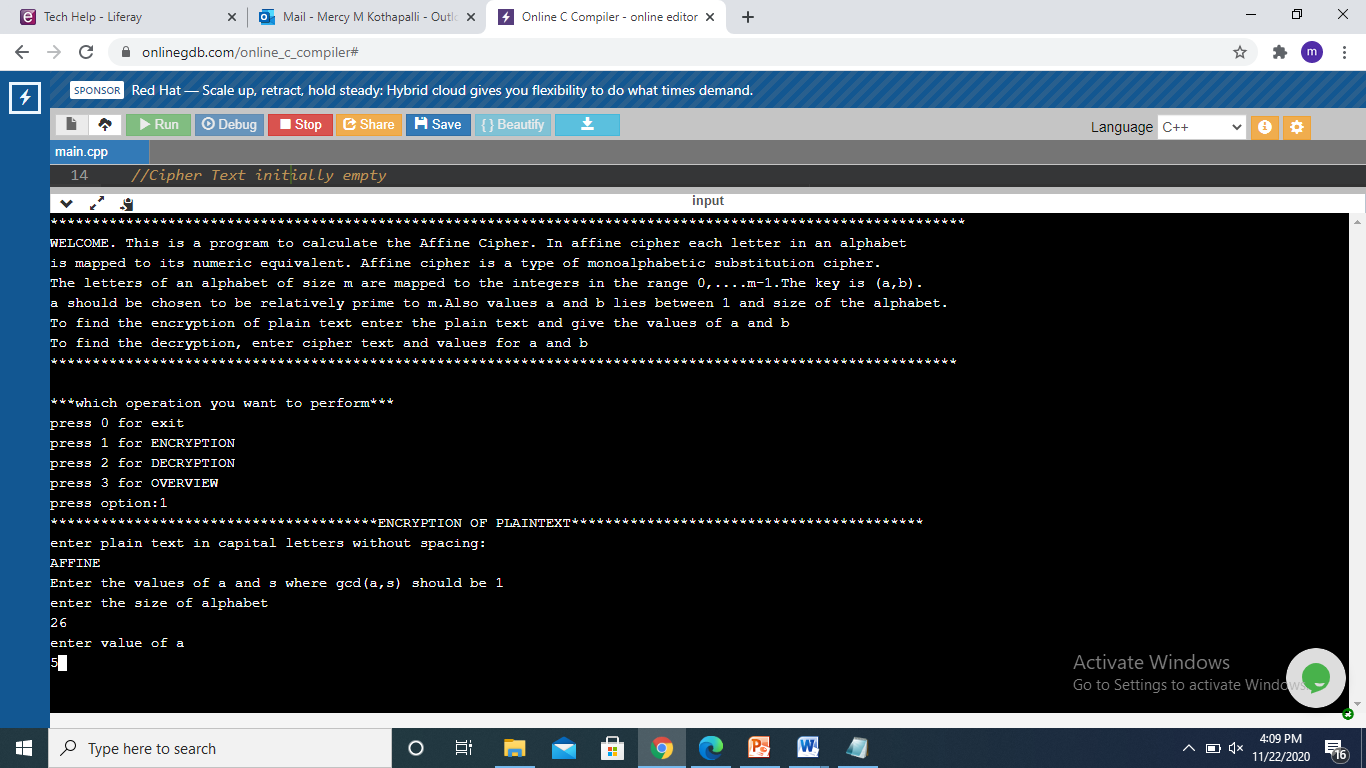
}

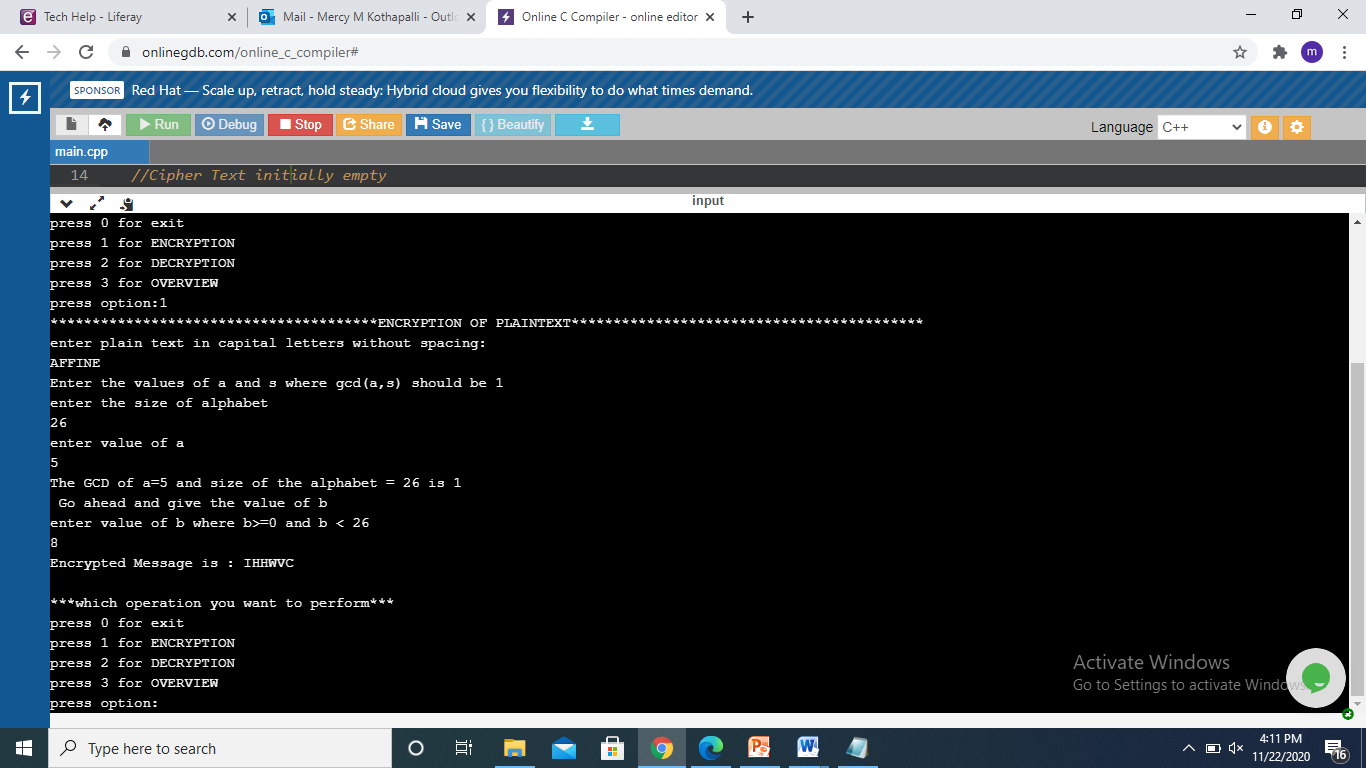
while(op!='0');

getch();

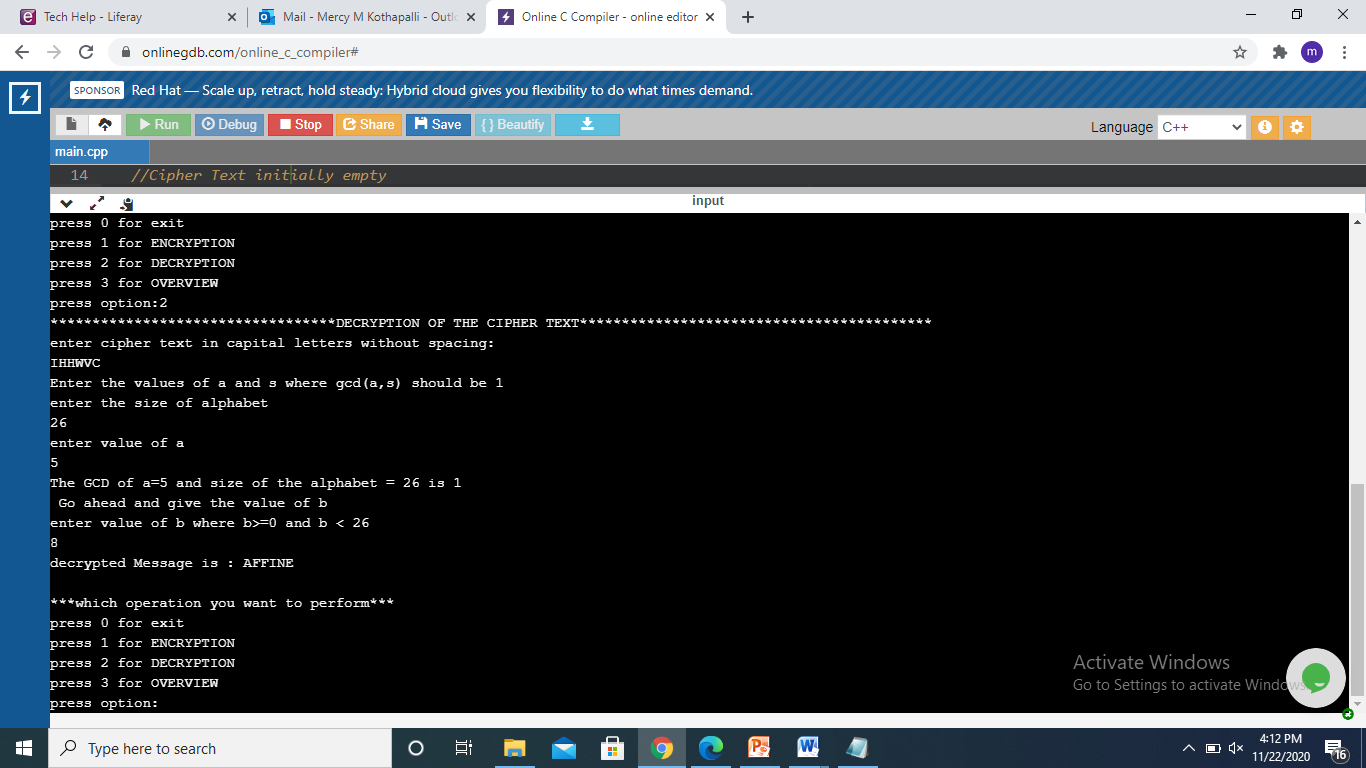
}

1. Output screen shots:
2. Encryption: plaintext: AFFINE, size of the alphabet: 26, a=5, b=8. Output: IHHWVC





1. Decryption: cipher text: IHHWVC, size of the alphabet: 26, a=5, b=8, output: AFFINE.



1. overview

