Assignment 4

1. Affine Cipher

In this cipher, there are 2 types of keys.

In Additive cipher, the characters are shifted with the key interval to form the encrypted message. In multiplicative cipher, the characters are multiplied with the key to form the new encrypted message. But in affine cipher, there are two keys, of which one behave as additive key, and the other as multiplicative key.

The general Encryption formula for the key :

New character = ((old character number * key1) + key2)%mod;

The Decryption formula has a key changed, is as :

New character = (((old character – key2+mod)%mod)*key3)%mod;

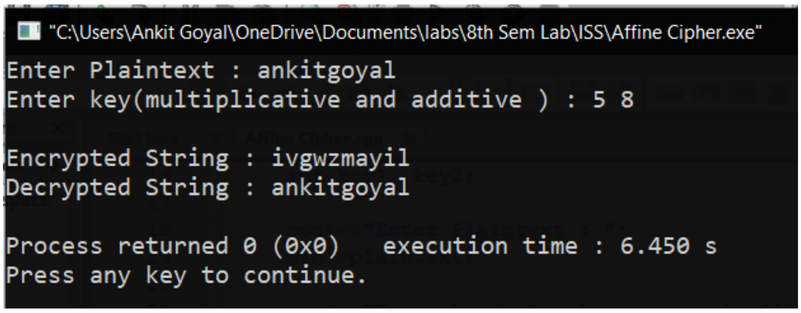
Where key3 is modular multiplicative inverse of key1 ➔ (key1*key3)%mod = 1;

Code :

```
#include <bits/stdc++.h>
using namespace std;
string lowercase(string s)
{
    for(int i = 0;i<s.length();i++)
    {
        if(s[i]>='A' && s[i]<='Z')
            s[i] = s[i]-'A'+'a';
    }
    return s;

}
int main()
{
    string plaintext;
    int key1, key2;

    cout<<"Enter Plaintext : ";
    cin>>plaintext;

    cout<<"Enter key(multiplicative and additive ) : ";
    cin>>key1>>key2;
    cout<<"\n";
    plaintext = lowercase(plaintext);

    /// Encryption
    for(int i = 0;i<plaintext.length();i++)
    {
        plaintext[i] = 'a' + ((plaintext[i]-'a')*key1+key2)%26;
    }

    cout<<"Encrypted String : "<<plaintext<<"\n";


    /// Decryption
    /// For inverse of key1, we need to find some other key
    /// (key1*x)mod26=(1)mod26
    int i = 1;
    while((key1*i)%26!=1)
        i++;
    int key3 = i;
```

```
        for(int i = 0;i<plaintext.length();i++)
        {
            plaintext[i] = 'a'+(((plaintext[i]-'a'-key2+26)%26)*key3)%26;
        }

        cout<<"Decrypted String : "<<plaintext;

        cout<<"\n";
    }
```

Result :



"C:\Users\Ankit Goyal\OneDrive\Documents\labs\8th Sem Lab\ISS\Affine Cipher.exe"

```
Enter Plaintext : ankitgoyal
Enter key(multiplicative and additive ) : 5 8

Encrypted String : ivgwzmayil
Decrypted String : ankitgoyal

Process returned 0 (0x0)   execution time : 6.450 s
Press any key to continue.
```

2. Autokey Cipher

This is a polyalphabetic substitution cipher. This cipher is same as vigenere cipher, but the key generation in this and vigenere ciphers are different. In this cipher, keytext is used as it is available, and then the plaintext itself is used to generate the encrypted text.

The encryption formula is
new character = (key character + old character)%mod;

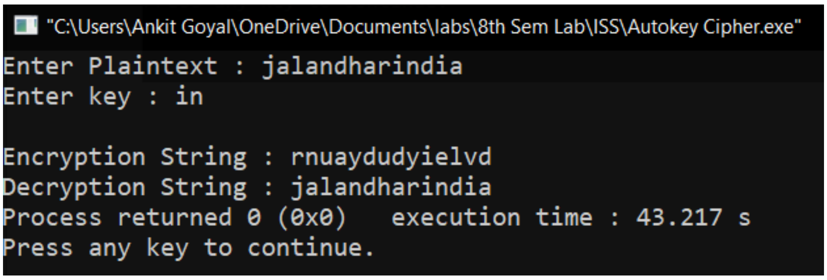
The Decryption formula is
New character = (old character – key character)%mod;

This is more secure than vigenere cipher as well.

Code :

```
#include <bits/stdc++.h>
using namespace std;
string lowercase(string s)
{
    for(int i = 0;i<s.length();i++)
    {
        if(s[i]>='A' && s[i]<='Z')
            s[i] = s[i]-'A'+'a';
    }
    return s;

}
int main()
{
    string plaintext;
   string key;

    cout<<"Enter Plaintext : ";
    cin>>plaintext;

    cout<<"Enter key : ";
    cin>>key;
    cout<<"\n";
    plaintext = lowercase(plaintext);
    key = lowercase(key);

    int n = plaintext.size();
    if(n>key.length())      key += plaintext;

    /// Encryption

    for(int i=0;i<plaintext.length();i++)
        plaintext[i] = 'a' + ((plaintext[i]-'a') + (key[i]-'a'))%26;

    cout<<"Encryption String : "<<plaintext<<"\n";

   for(int i=0;i<plaintext.length();i++)
        plaintext[i] = 'a' + (plaintext[i]- key[i]+26)%26;

   cout<<"Decryption String : "<<plaintext;
}
```

Result :



"C:\Users\Ankit Goyal\OneDrive\Documents\labs\8th Sem Lab\ISS\Autokey Cipher.exe"

```
Enter Plaintext : jalandharindia
Enter key : in

Encryption String : rnuaydudyielvd
Decryption String : jalandharindia
Process returned 0 (0x0)   execution time : 43.217 s
Press any key to continue.
```