Lab - 02

Aim:- write a Program to implement.
1. Extended Euclidean Algorithm for finding multiplicative Inverse
2. Multiplicative cipher
3. Affine Cipher.

Program - 1 : Extended Euclidean Algorithm

Code :-

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

int MultiplicativeInverse(int a, int b)
{
    int r1 = max(a,b), r2 = min(a,b), t1 = 0, t2 = 1;
    int q, r, t;
    while(r2 > 0)
    {
        q = r1/r2;
        r = r1 - r2*q;
        r1 = r2;
        r2 = r;
        t = t1 - q*t2;
        t1 = t2;
        t2 = t;
    }
```

```
        if(r1 != r)
            return -1;
        else
        {
            if(t1<0)
            {
                t1 += 26;
            }
            return t1;
        }
int main()
{
    int n,a, inverse;
    cin>>n>>a;
    if((inverse = MultiplicativeInverse(n,a)) != -1)
    {
        cout<<"inverse:" << inverse<<endl;
    }
    else
        cout<< "Inverse Not possible"<<endl;

    return 0;
}
```

Input:-

26 3

o/p inverse : 9

Program-2 :- Multiplicative Cipher

code :-

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

string Encrypt(string plainText, int k)
{
    string cipherText="";
    for(int i=0; i<plainText.length(); ++i)
    {
        cipherText+=((plainText[i]-'a')*k)%26)
                    +'a';
    }
    cout<<"Encrypted Text: "<<cipherText<<endl;
    return cipherText;
}

void Decrypt(string cipherText, int k)
{
    String plainText="";
    for(int i=0; i<cipherText.length(); ++i)
    {
        plainText+=(((cipherText[i]-'a')*k)%26)
                    +'a';
    }
    cout<<"Decrypted Text:"<<plainText<<endl;
}
```

```
int main()
{
    int k, n=26;
    string pt;
    cout<<"enter your msg: "<<endl;
    cin>>pt;
    cout<<"enter your key: ";
    cin>>k;
    while( multiplicativeInverse(k,n)==-1) //gcd(k,n)≠1
    {
        cout<<"Inverse not possible"<<endl;
        cout<<"enter other key: ";
        cin>>k;
    }
    string cipherText = Encrypt(pt, k);
    k = multiplicativeInverse(k,n); //Decryption key
                                      will be Inverse.
    Decrypt( CipherText, k);
    return 0;
}
```

Input:

```
Enter your msg: muneet
Enter your key: 2
Enter other key: 3
```

Output:-

Inverse not possible for key = 2

Encrypted Text = kinap

Decrypted Text = muneet.

Program-3 :- Affine cipher.

Source code:-

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

string Encrypt(string plainText, int k1, int k2)
{
    string cipherText="";
    for(int i=0; i<plainText.length(); ++i)
    {
        cipherText+=(((plainText[i]-'a')*k1+k2)
                    %26)+'a';
    }
    cout<<"Encrypted Text: "<<cipherText<<endl;
    return cipherText;
}

Void Decrypt(string cipherText, int k1, int k2)
{
    string plainText="";
    for(int i=0; i<cipherText.length(); ++i)
    {
        plainText+=(((cipherText[i]-'a')+k2)*k1)
                    %26)+'a';
    }
    cout<<"Decrypted Text: "<<plainText<<endl;
}
```

```
int main()
{
   int n=26, k1, k2;
   string pt;
   cout<<"enter your msg:";
   cin>>pt;
   cout<<"enter key1:";
   cin>>k1;
   cout<<"enter key2:";
   cin>>k2;
   while(MultiplicativeInverse(k1,n)==-1)
   {
      cout<<"key1 is not valid";<endl;
      cout<<"enter other key:";
      cin>>k1;
   }
   string cipherText = Encrypt(pt, k1, k2);
   k1 = MultiplicativeInverse(k1,n);
   k2 = n-k2; //(Additive Inverse)
   Decryption
   Decrypt(cipherText, k1, k2);
   return 0;
}
```

Input	out-put
Enter msg:- munaf	Encrypted Text: n1qds
Enter k1:- 3	decrypted Text: munaf
Enter k2: 3	