```
#include <iostream>
#include <cstdlib>
#include <string>
#include <time.h>

/**
* Patrick Rodriguez Marquez
* https://github.com/pbrodriguezm/UCSP-AlgebraAbstracta/blob/master/afin.cpp
*/

using namespace std;

int mcd(int a, int b){

    int max = a;
    int min = b;
    int res;
    if(b>a){
        max=b;
    }

    while(b != 0 ){
        res=b;
        b=a%b;
        a=res;
    }

    return res;
}




int resto(int a, int n)
{


        int q = a/n;
        int r = a%n;


        if(r<0){
            r=n+r;
            q=q+1;
        }


        return r;

}




//EUCLIDES EXNTENDIDO
int eulidesExtendido(int a, int b)
{
      int d=0,q=0,r=0;
      int x=0,x1,x2;
      int y=0,y1,y2;


      if(b==0)
      {
```

```
            d=a;
            x=1;
            y=0;
            return x;
      }
      x2=1; x1=0; y2=0 ; y1=1;
    int i=0;

      while(b>0)
      {
            q=a/b; r=a-(q*b); x=x2-(q*x1); y=y2-(q*y1);
            a=b; b=r; x2=x1; x1=x; y2=y1; y1=y;
        i++;

      }
      d=a; x=x2; y=y2;

      return x;

}


class Receptor {
    public:
    int clave_a;
    int clave_b;
    string mensaje;
    string alfabeto;
    int inversa;

    Receptor();
    void setMensaje(string m_mensaje);
    void setClave(int m_clave, int m_clave_b );
    string Decifrar();
    string Decifrar_dos();
};

    Receptor::Receptor(){
        alfabeto= "abcdefghijklmnopqrstuvwxyz";


    }

    void Receptor::setMensaje(string m_mensaje){        //Mensaje escriptado
            mensaje= m_mensaje;

    }

    void Receptor::setClave(int m_clave, int m_clave_b){
        clave_a= m_clave;
        clave_b= m_clave_b;
    }

    string Receptor::Decifrar(){
        //descifrar DC=AM-B(MOD N)
        string text;
        for (size_t i = 0; i < mensaje.size(); i++)
        {
            char crip;
            if(mensaje[i]-clave_a < 'a'){
                crip= (mensaje[i]- clave_a) + alfabeto.size();
            }else {
                 crip= mensaje[i]-clave_a;
            }
```

```
            text +=crip;
        }
        return text;
    }


    string Receptor::Decifrar_dos(){
        string text;

            for (size_t i = 0; i < mensaje.size(); i++)
            {
                    int pos=alfabeto.find(mensaje[i]);

                    inversa = eulidesExtendido(clave_a,alfabeto.size());
                    //if(inversa<0) //modulo
                     int valortmp= inversa*(pos-clave_b);
                        if(valortmp > alfabeto.size()-1) {
                            valortmp = resto(valortmp,alfabeto.size());

                        }
                        cout<<"Pos: "<<valortmp;
                        text+=alfabeto[valortmp];

            }

            return text;

        }



class Emisor {
public:
    int clave;
    string mensaje;
    string alfabeto;
    int clave_a;
    int clave_b;
    int inversaa;

    Emisor();
    void setMensaje(string m_mensaje);
    void setClave(int m_clave);
    string Cifrar();
};


    Emisor::Emisor(){


        alfabeto= "abcdefghijklmnopqrstuvwxyz";

        while(!clave_a) {
             srand(time(NULL));

            int randon = (rand() % alfabeto.size());
            if(mcd(randon,alfabeto.size()) == 1 ) {
                clave_a=randon;
            }
        }

        srand48(time(NULL));

        clave_b = (rand() % alfabeto.size());
```

```
        cout<<"CLAVE A: "<<clave_a<<endl;

        cout<<"CLAVE B: "<<clave_b<<endl;


    }

    void Emisor::setMensaje(string m_mensaje){
            mensaje= m_mensaje;

    }

    void Emisor::setClave(int m_clave){
        clave= m_clave;
    }




/**
 * Cifrafo de AFIN Strings
*/
    string Emisor::Cifrar(){
    string text;

        for (size_t i = 0; i < mensaje.size(); i++)
        {
                int pos=alfabeto.find(mensaje[i]);
                 cout<<"CIFRADO: a:"<<clave_a<<"- x:"<<pos<<"- b:"<<clave_b<<"-
rpta: "<<(clave_a*pos)+clave_b;
                    int valortmp=(clave_a*pos)+clave_b;
                 if((clave_a*pos)+clave_b > alfabeto.size()-1) {
                     valortmp = resto(valortmp,alfabeto.size());

                 }
                    cout<<" Pos: "<<valortmp;
                 text+=alfabeto[valortmp];


        }

        return text;

    }


int main()
{
    Emisor a;
    string mensaje; cout<<"INICIANDO Encriptación \n** Mensaje: "; cin>>
mensaje;




    a.setMensaje(mensaje);


    string mensaje_escriptado = a.Cifrar();  cout << endl;



    std::cout<<"** Encriptado: " << mensaje_escriptado<< std::endl;
```

```
     std::cout<<endl<<"INICIANDO DESECRIPTACION  mensaje recibodo: " <<
mensaje_escriptado<< std::endl;
    Receptor b;
    b.setMensaje(mensaje_escriptado);
    b.setClave(a.clave_a, a.clave_b);
    string mensaje_desencriptado = b.Decifrar_dos();


    std::cout<<endl<<"** Desencriptado: " << mensaje_desencriptado<< std::endl;



}
```