```cpp
//Clase del cifrado ----------------------------------------------

#include <iostream>
#include <stdlib.h>
#include <string.h>

using namespace std;

class afin
{
public:
  string alfabeto;
  int clave;
  int tam_alf;
  int _mod;
  afin(int a);
  int mod(int a, int b);
  int inversa(int a, int b);
  int euclides_ext(int a, int b);
  int MCD(int a, int b);
  string cifrar(string mensaje);
  string descifrar(string mensaje_cifrado);
};

afin::afin(int a)
{
  alfabeto = "ABCDEFGHIJKLMNOPQRSTUVWXYZ ";
  clave = a;
  tam_alf = alfabeto.length();
  _mod = mod(rand() % 100, tam_alf);
}

int afin::mod(int a, int b)
{
  int resto;
  resto = a - ((a / b) * b);
  if(resto < 0)
  {
    resto = a - (((a / b) - 1) * b);
  }
  return resto;
}

int afin::euclides_ext(int a, int b)
{
  int q, s, t, r;
  int r1 = a;
  int r2 = b;
  int s1 = 1;
  int s2 = 0;
  int t1 = 0;
  int t2 = 1;

  while(r2 > 0)
  {
    q = r1 / r2;
    r = r1 - q * r2;
    r1 = r2;
    r2 = r;

    s = s1 - q * s2;
    s1 = s2;
    s2 = s;

    t = t1 - q * t2;
    t1 = t2;
```

```
    t2 = t;
  }
  s = s1;
  t = t1;
  return s;
}

int afin::MCD(int a, int b)
{
  int resto;
  resto = mod(a, b);
  while(resto > 0)
  {
    a = b;
    b = resto;
    resto = mod(a, b);
  }
  return b;
}

int afin::inversa(int a, int b)
{
  if(MCD(a, b) == 1)
  {
    int inverso = euclides_ext(a, b);
    if (mod((inverso * a), b) == 1)
    {
      if(inverso < 0)
      {
        inverso = mod(inverso, b);
      }
      return inverso;
    }
  }
  else
  {
    return 0;
  }
}

string afin::cifrar(string mensaje)
{
  int auxiliar1;
  int auxiliar2;

  string mensaje_cifrado;

  for(int i = 0; i < mensaje.length(); i++)
  {
    auxiliar1 = alfabeto.find(mensaje[i]);
    auxiliar2 = mod(((clave * auxiliar1) + _mod), tam_alf);
    mensaje_cifrado = mensaje_cifrado + alfabeto[auxiliar2];
  }
  return mensaje_cifrado;
}

string afin::descifrar(string mensaje_cifrado)
{
  int auxiliar1;
  int auxiliar2;
  int auxiliar3;

  string mensaje_descifrado;

  auxiliar1 = inversa(clave, tam_alf);
  if(auxiliar1 == 0)
```

```
133     {
134       return 0;
135     }
136     for(int i = 0; i < mensaje_cifrado.length(); i++)
137     {
138       auxiliar2 = alfabeto.find(mensaje_cifrado[i]);
139       auxiliar3 = mod(auxiliar1 * (auxiliar2 - _mod), tam_alf);
140       mensaje_descifrado = mensaje_descifrado + alfabeto[auxiliar3];
141     }
142     return mensaje_descifrado;
143   }
144
145   //Main ------------------------------------------------------------
146
147   #include <iostream>
148   #include <stdlib.h>
149   #include <string.h>
150
151   #include "afin.h"
152
153   using namespace std;
154
155   int main()
156   {
157       afin afin(23);  // Clave que usé en la práctica grupal.
158       string mensaje;
159       string mensaje_cifrado;
160       string mensaje_descifrado;
161       cout << "Introduzca el mensaje a cifrar: " << endl;   // Escribir el mensaje en
mayúsculas.
162       getline(cin, mensaje);
163
164       mensaje_cifrado = afin.cifrar(mensaje);
165       cout << "Mensaje cifrado: " << endl;
166       cout << mensaje_cifrado << endl;
167
168       mensaje_descifrado = afin.descifrar(mensaje_cifrado);
169       cout << "Mensaje descifrado: " << endl;
170       cout << mensaje_descifrado << endl;
171   }
172
```