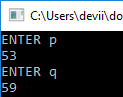
**Садов Д.В. ИВБО 7-14.**

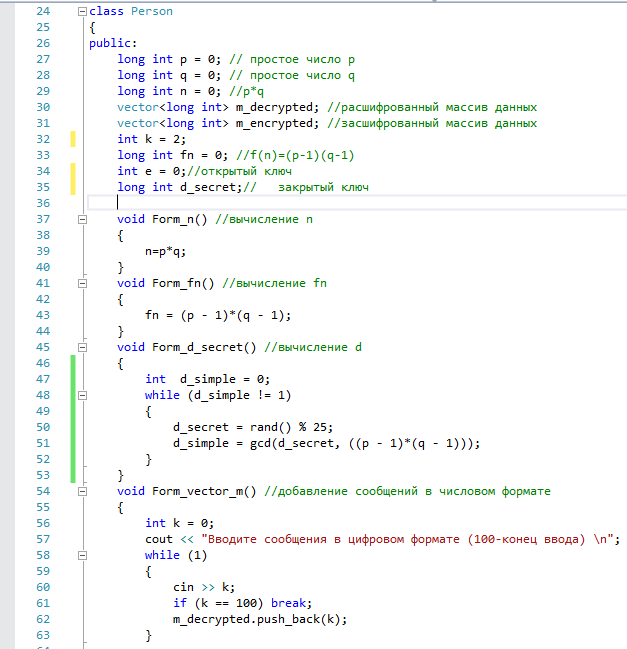
**Лабораторная работа №4** В данной лабораторной работе нужно было реализовать алгоритм RSA шифрования.

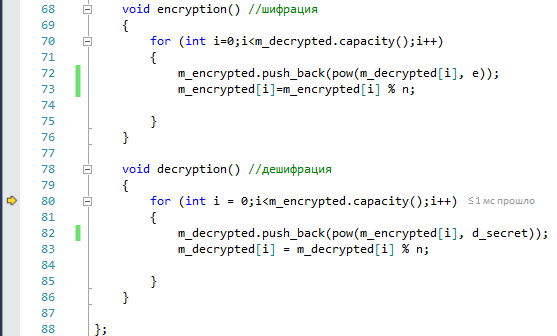
Реализована программа следующим образом:

1. Пользователь вводит пару простых чисел p и q



1. Создается класс со следующими полями и методами, а также его 2 экземпляра Alice и Bob:



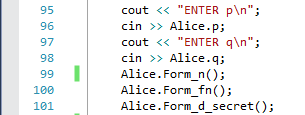


1. После ввода двух простых чисел Алиса вычисляет модуль, т.е. их произведение

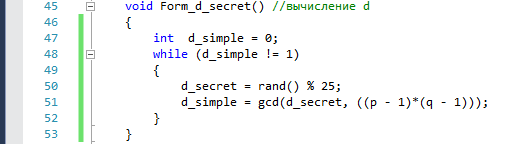
n=p\*q (метод Form\_n)

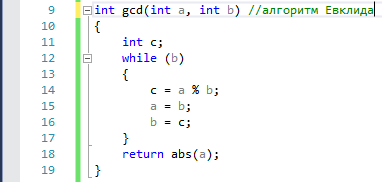
Далее вычисляется функция Эйлера fn=(p-1)\*(q-1) (метод Form\_fn),

А также генерирует число d\_secret для закрытого ключа и проверяет его на взаимопростоту.



Формирование d\_secret происходит в методе Form\_d\_secret, который использует алгоритм Евклида для проверки чисел на взаимопростоту:

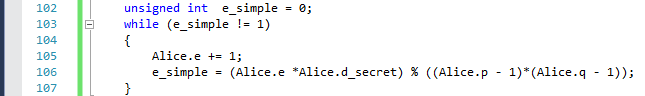




1. После того как сгенерировано d, Элис генерирует число е, таким образом, чтобы выполнялось следующее:

(e\*d)%((p\_simple-1)\*(q\_simple-1))=1

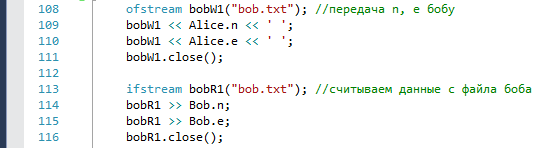
В программе это реализовано следующим образом:



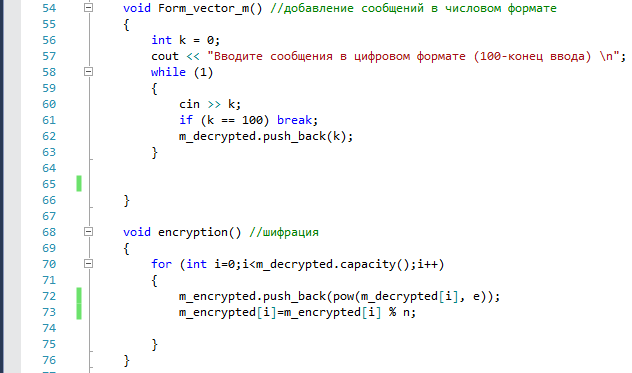
1. После формирования числа e, Элис передает Бобу пару значений – открытый ключ

(n и e)

После записи этих значений в файл Боба, он заполняет у себя этими значениями поля, считывая из файла



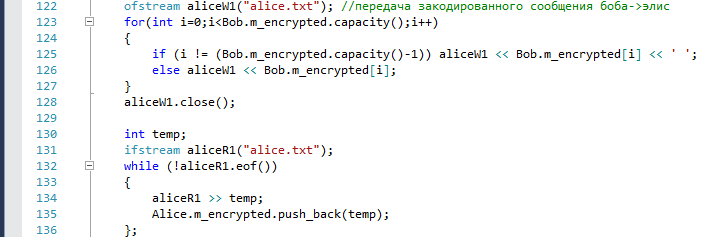
1. После получения n и e начинаем ввод числовых значений, которые хотим передать от Боба к Элис и шифруем их



Шифрация происходит по формуле m^e mod n

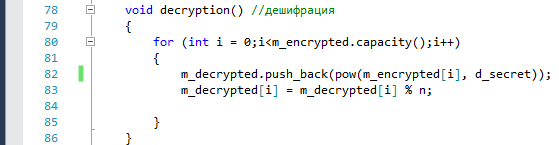
Зашифрованный массив чисел передается в файл Элис, откуда она заполняет массив

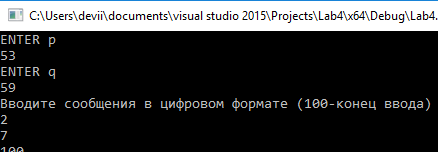
M\_encrypted полученными данными



1. Методом decryption Элис расшифровывает переданные ей данные используя формулу

m^d mod n



При введенных данных 

Сообщение дешифрируется верно :



**Листинг программы**

#include <iostream>

#include <fstream>

#include <time.h>

#include <stdlib.h>

#include <string>

#include <vector>

using namespace std;

int gcd(int a, int b) //алгоритм Евклида

{

int c;

while (b)

{

c = a % b;

a = b;

b = c;

}

return abs(a);

}

class Person

{

public:

long int p = 0; // простое число p

long int q = 0; // простое число q

long int n = 0; //p\*q

vector<long int> m\_decrypted; //расшифрованный массив данных

vector<long int> m\_encrypted; //засшифрованный массив данных

int k = 2;

long int fn = 0; //f(n)=(p-1)(q-1)

int e = 0;//открытый ключ

long int d\_secret;// закрытый ключ

void Form\_n() //вычисление n

{

n=p\*q;

}

void Form\_fn() //вычисление fn

{

fn = (p - 1)\*(q - 1);

}

void Form\_d\_secret() //вычисление d

{

int d\_simple = 0;

while (d\_simple != 1)

{

d\_secret = rand() % 25;

d\_simple = gcd(d\_secret, ((p - 1)\*(q - 1)));

}

}

void Form\_vector\_m() //добавление сообщений в числовом формате

{

int k = 0;

cout << "Вводите сообщения в цифровом формате (100-конец ввода) \n";

while (1)

{

cin >> k;

if (k == 100) break;

m\_decrypted.push\_back(k);

}

}

void encryption() //шифрация

{

for (int i=0;i<m\_decrypted.capacity();i++)

{

m\_encrypted.push\_back(pow(m\_decrypted[i], e));

m\_encrypted[i]=m\_encrypted[i] % n;

}

}

void decryption() //дешифрация

{

for (int i = 0;i<m\_encrypted.capacity();i++)

{

m\_decrypted.push\_back(pow(m\_encrypted[i], d\_secret));

m\_decrypted[i] = m\_decrypted[i] % n;

}

}

};

int main(int args, const char \* argv[])

{ //исходных данных нет, только 2 пустых файла

setlocale(LC\_ALL, "rus");

srand(time(NULL));

Person Alice, Bob; //человеки

cout << "ENTER p\n";

cin >> Alice.p;

cout << "ENTER q\n";

cin >> Alice.q;

Alice.Form\_n();

Alice.Form\_fn();

Alice.Form\_d\_secret();

unsigned int e\_simple = 0;

while (e\_simple != 1)

{

Alice.e += 1;

e\_simple = (Alice.e \*Alice.d\_secret) % ((Alice.p - 1)\*(Alice.q - 1));

}

ofstream bobW1("bob.txt"); //передача n, e бобу

bobW1 << Alice.n << ' ';

bobW1 << Alice.e << ' ';

bobW1.close();

ifstream bobR1("bob.txt"); //считываем данные с файла боба

bobR1 >> Bob.n;

bobR1 >> Bob.e;

bobR1.close();

Bob.Form\_vector\_m(); //формируем сообщение боба в числовом варианте

Bob.encryption(); //кодируем сообщение боба к элис

ofstream aliceW1("alice.txt"); //передача закодированного сообщения боба->элис

for(int i=0;i<Bob.m\_encrypted.capacity();i++)

{

if (i != (Bob.m\_encrypted.capacity()-1)) aliceW1 << Bob.m\_encrypted[i] << ' ';

else aliceW1 << Bob.m\_encrypted[i];

}

aliceW1.close();

int temp;

ifstream aliceR1("alice.txt");

while (!aliceR1.eof())

{

aliceR1 >> temp;

Alice.m\_encrypted.push\_back(temp);

};

Alice.decryption(); //расшифровка используя ключ d по формуле с^d mod n = X, где с-закодированное сообщение боба, d- ключ, n-p\*q, X-исходное сообщение

system("pause");

return 0;

}