



МИНИСТЕРСТВО НАУКИ
И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное
образовательное учреждение высшего образования
«НОВОСИБИРСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»



**НГТУ
НЭТИ** | **Факультет прикладной
математики и информатики**

Кафедра прикладной математики
Практическое задание № 2
по дисциплине «Основы криптографии»

Место для ввода текста.

Бригада 2	ЗАТОЛОЦКАЯ ЮЛИЯ
Группа ПМ-91	БАРСУКОВА НАТАЛЬЯ
Вариант 2	ЧЕРНИКОВ ДАНИЛ

Преподаватели СТУПАКОВ ИЛЬЯ МИХАЙЛОВИЧ

Новосибирск, 2021

1 Задание

Алгоритм Эль-Гамала

Написать программы реализующие алгоритм Эль-Гамала.

1. Генерация ключей
 1. Прочитать из консоли числа p и g .
 2. Проверить что p простое число, а g первообразный корень по модулю p .
 3. Сгенерировать закрытый ключ x и открытый ключ y по алгоритму Эль-Гамала.
2. Шифрование текста
 1. Прочитать из консоли числа p , g и y .
 2. Прочитать сообщение M (число меньше p), зашифровать его алгоритмом Эль-Гамала и вывести результат.
3. Расшифровка текста
 1. Прочитать из консоли числа p , g и x .
 2. Прочитать зашифрованное сообщение, расшифровать его алгоритмом Эль-Гамала и вывести результат.

Для возведения в степень должен использоваться алгоритм быстрого возведения в степень. Программа должна корректно работать для значений $p < 2^{32}$.

2 Программа

```
1 using System;
2 using System.Numerics;
3
4 namespace kript2
5 {
6     class Program
7     {
8         static bool keys_generator(ref BigInteger y, ref BigInteger g, ref BigInteger p,
9         ref BigInteger x)
10         {
11             // Генерирует 2 ключа: публичный (y, g, p) и приватный (x)
12             // true - ключи успешно сгенерировались, иначе false
13             Console.WriteLine("Введите p: ");
14             p = BigInteger.Parse(Console.ReadLine());
15             Console.WriteLine("Введите g: ");
16             g = BigInteger.Parse(Console.ReadLine());
17
18             if (!primality_test(p))
19             {
20                 Console.WriteLine("Число p не является простым.");
21                 return false;
22             }
23
24             if (!is_primitive_root(g, p))
25             {
26                 Console.WriteLine("Число g не является первообразным корнем по модулю
27 p.");
28                 return false;
29             }
30
31             x = RandomIntegerBelow(p - 1);
32             y = BigInteger.ModPow(g, x, p);
```

```

31         return true;
32     }
33
34     static BigInteger gcd(BigInteger a, BigInteger b)
35     {
36         // Алгоритм Евклида
37         return b == 0 ? a : gcd(b, a % b);
38     }
39     static BigInteger phi(BigInteger n)
40     {
41         // Функция Эйлера
42         BigInteger result = 1;
43         BigInteger previous = -1;
44         for (int i = 2; i <= n / i; i++)
45         {
46             if (n % i == 0)
47             {
48                 if (i == previous)
49                     result *= i;
50                 else
51                 {
52                     result *= i - 1;
53                     previous = i;
54                 }
55                 n /= i;
56             }
57             else
58                 i++;
59         }
60         BigInteger p = n;
61         if (n > 1)
62             if (p == previous)
63                 result *= p;
64             else
65                 result *= p - 1;
66         return result;
67     }
68     static bool primality_test(BigInteger n)
69     {
70         // Является ли число n простым?
71         if (n == 1)
72             return false;
73         if (n % 2 == 0 && n != 2)
74             return false;
75         for (int i = 3; i <= n / i; i += 2)
76             if (n % i == 0)
77                 return false;
78         return true;
79     }
80
81     static BigInteger RandomIntegerBelow(BigInteger N)
82     {

```

```

83         byte[] bytes = N.ToByteArray();
84         BigInteger R;
85         Random random = new Random();
86         do
87         {
88             random.NextBytes(bytes);
89             bytes[bytes.Length - 1] &= (byte)0x7F;
90             R = new BigInteger(bytes);
91         } while (R >= N);
92
93         if (R == 0)
94             R += 2;
95         if (R == 1)
96             R += 1;
97
98         return R;
99     }
100
101     static bool is_primitive_root(BigInteger g, BigInteger m)
102     {
103         // Является ли число g первообразным корнем по модулю m?
104         if (g >= m || g < 0 || gcd(g, m) != 1)
105             return false;
106
107         BigInteger n = phi(m);
108         BigInteger phi_m = n;
109         BigInteger previous = -1;
110         for (BigInteger i = 2; i <= n / i; i++)
111         {
112             while (n % i == 0)
113             {
114                 if (i != previous && BigInteger.ModPow(g, phi_m / i, m) == 1)
115                     return false;
116                 n /= i;
117                 previous = i;
118             }
119         }
120
121         if (n != 1)
122         {
123             if (n != previous && BigInteger.ModPow(g, phi_m / n, m) == 1)
124                 return false;
125         }
126         return true;
127     }
128     static void Encrytion(BigInteger p, BigInteger g, BigInteger y)
129     {
130         BigInteger k = RandomIntegerBelow(p - 1);
131         Console.WriteLine("Введите сообщение: ");
132         BigInteger M = Convert.ToInt32(Console.ReadLine());
133         BigInteger s = BigInteger.ModPow(y, k, p);
134         BigInteger a = BigInteger.ModPow(g, k, p);

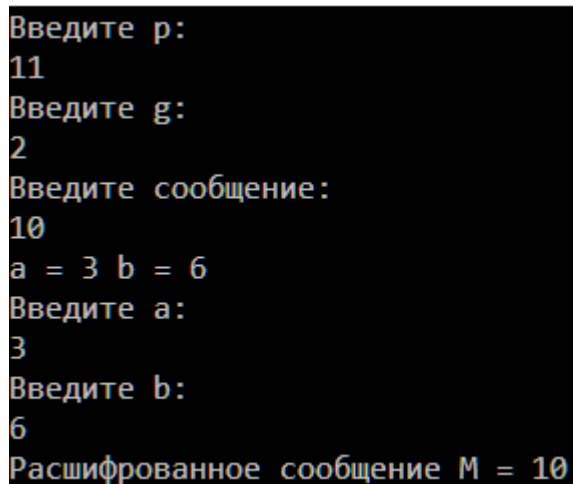
```

```

135
136     BigInteger b = (s * M) % p;
137     Console.WriteLine("a = " + a + " b = " + b);
138
139 }
140 static void Decryption(BigInteger x, BigInteger a, BigInteger b, BigInteger p)
141 {
142     BigInteger M;
143     BigInteger ret = BigInteger.ModPow(a, p - 1 - x, p);
144     M = ((b % p) * ret) % p;
145
146     Console.WriteLine("Расшифрованное сообщение M = " + M);
147 }
148 static void Main(string[] args)
149 {
150     BigInteger p = 0; BigInteger g = 0; BigInteger x = 0; BigInteger y = 0;
151     if (keys_generator(ref y, ref g, ref p, ref x))
152     {
153         Encrytion(p, g, y);
154         Console.WriteLine("Введите a: ");
155         BigInteger a = BigInteger.Parse(Console.ReadLine());
156         Console.WriteLine("Введите b: ");
157         BigInteger b = BigInteger.Parse(Console.ReadLine());
158         Decryption(x, a, b, p);
159     }
160
161 }
162 }
163 }

```

3 Тесты



```

Введите p:
11
Введите g:
2
Введите сообщение:
10
a = 3 b = 6
Введите a:
3
Введите b:
6
Расшифрованное сообщение M = 10

```

```
Введите p:  
4304021  
Введите g:  
3  
Введите сообщение:  
4000000  
a = 843694 b = 3874647  
Введите a:  
843694  
Введите b:  
3874647  
Расшифрованное сообщение M = 4000000
```