**Лэ Никита**

1. **к. 4 гр. 1 подгруппа**

**1) Используя некоторый вектор S = (103, 107, 211, 430, 863, 1716, 3449, 6907, 13807, 27610), вычислить ключи для зашифрования и расшифрования сообщений (относится к ранцевому алгоритму).**

Так как в векторе S каждый следующий элемент больше суммы предыдущих, то это сверхвозрастающая последовательность.

**W =** 55203 (сумма всех элементов).

Возьмём простое Q, которое больше W:

**Q =** 55207

Теперь выберем число R из интервала [1, Q], взаимно простое 1-му:

**R** = 25013

**S, R, Q** – закрытый ключ

Сгенерируем **открытый ключ B**:

**Si \* R mod Q = Bi**

103 \* 25013 mod 55203 = 37001

107 \* 25013 mod 55203 = 26647

211 \* 25013 mod 55203 = 33458

430 \* 25013 mod 55203 = 46208

863 \* 25013 mod 55203 = 1846

1716 \* 25013 mod 55203 = 29577

3449 \* 25013 mod 55203 = 42751

6907 \* 25013 mod 55203 = 34604

13807 \* 25013 mod 55203 = 4523

27610 \* 25013 mod 55203 = 19400

**B =** (37001, 26647, 33458, 46208, 1846, 29577, 42751, 34604, 4523, 19400) – открытый ключ

Для того, чтобы зашифровать нам требуется знать ключ **B**, а для расширования **R** и **Q** (R берётся обратное по модулю Q, т.е. 25013)

Зашифруем сообщение 1001101100:

1\*37001+ 0\*26647 + 0\*33458 + 1\*46208 + 1\*1846 + 0\*29577 + 1\*42751 + 1\* 34604 + 0\*4523 + 0\*19400 = 162410

**2. Найти первообразные корни (если они существуют) чисел (р): 19, 23, 27, 31, 37.**

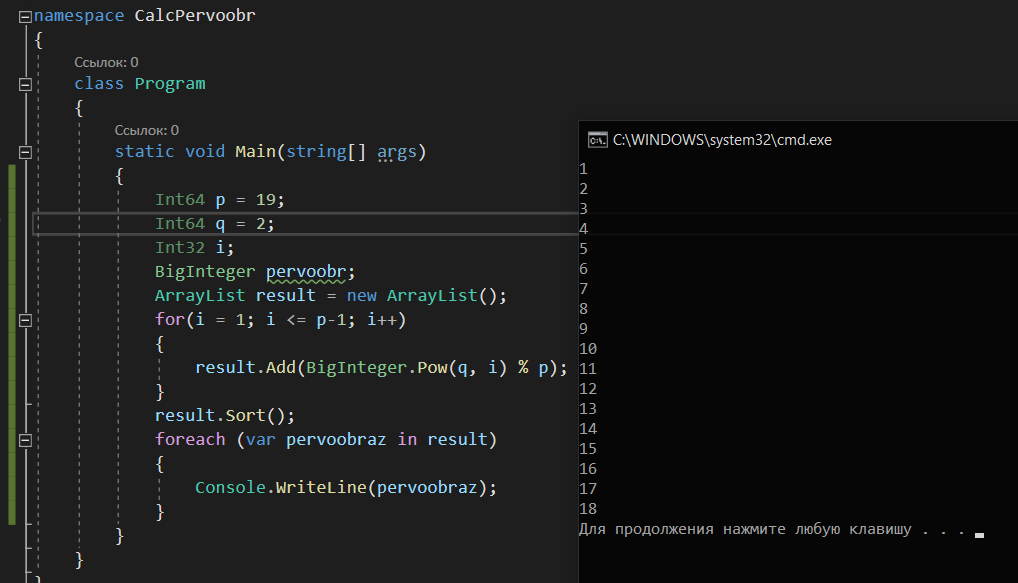
**(q^i) mod p**

**p** – модуль, **q** – первообразная, **i** – степень.

**(q** < **p ,1 <= i <= p-1)**

**1) p = 19.**

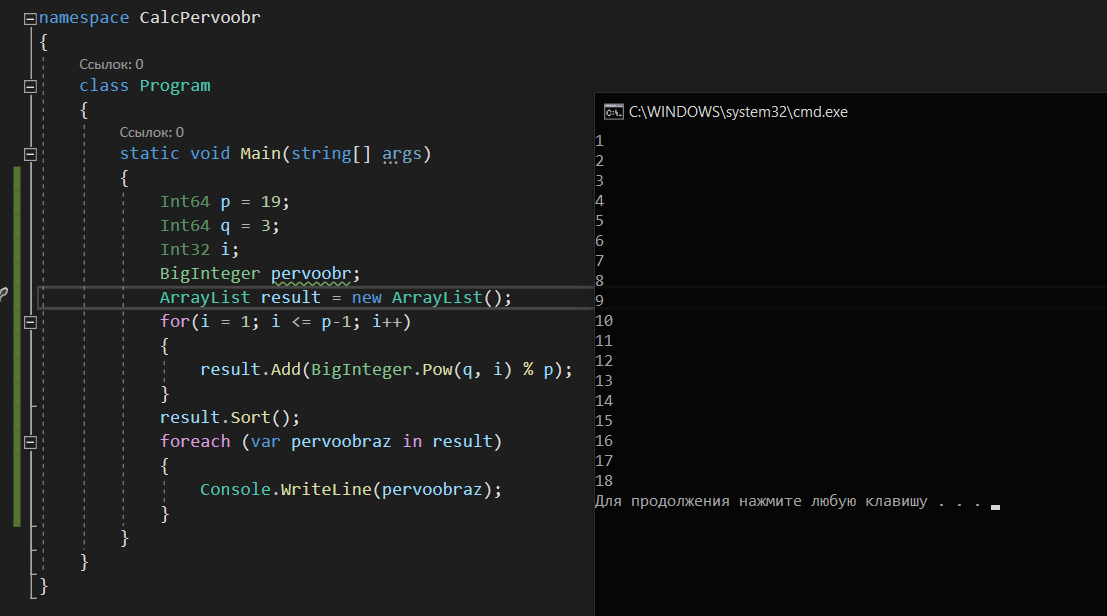
**q = 2**



Так как мы получили все возможные взаимнопростые остатки p => число 2

является первообразным корнем 19.

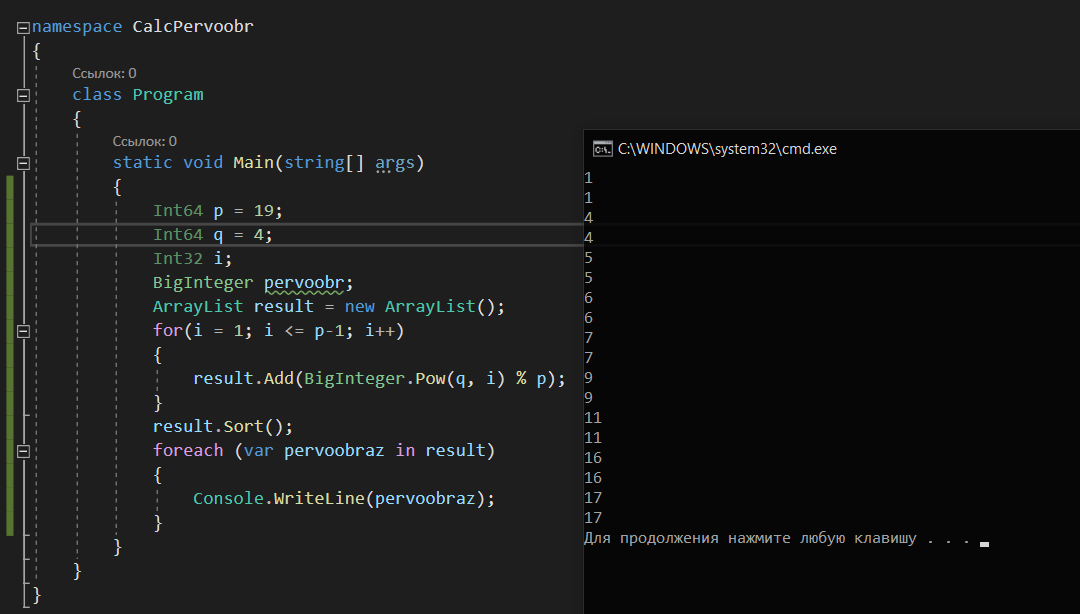
**q=3**



Так как мы получили все возможные взаимнопростые остатки p => число 3

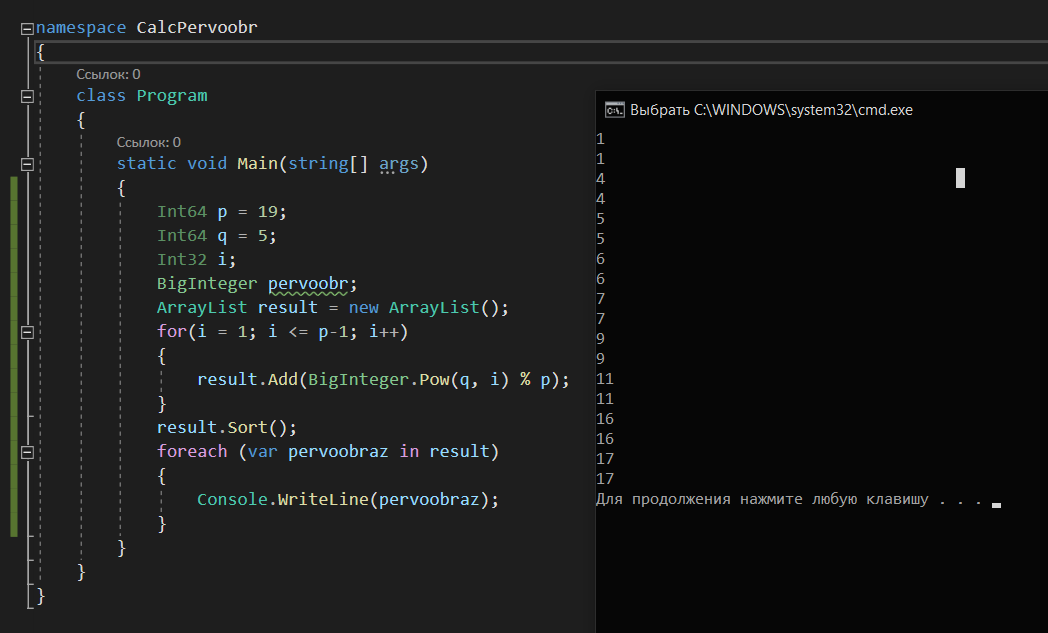
является первообразным корнем 19.

**q=4**



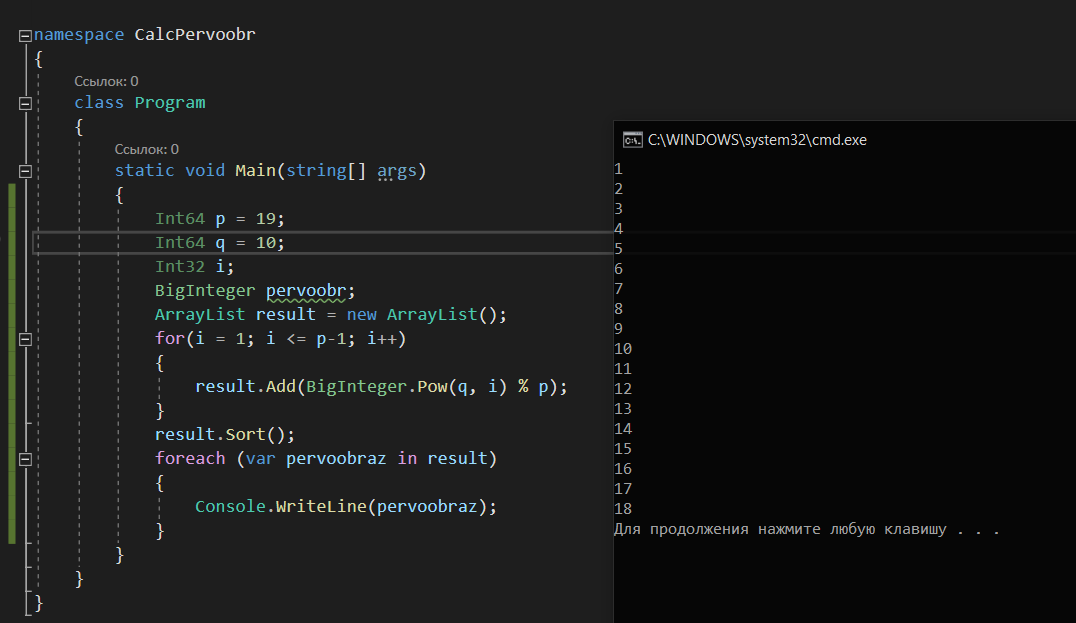
4 не является, так как не хватает {2, 3, 8, 10, 12, 13, 14, 15, 18}

**q=5**

****

5 не является, так как не хватает [2, 3, 8, 10, 12, 13, 14, 15, 18]

Так же делаем и с остальными числами.  
Вот ещё один первообразный корень

**q=10**

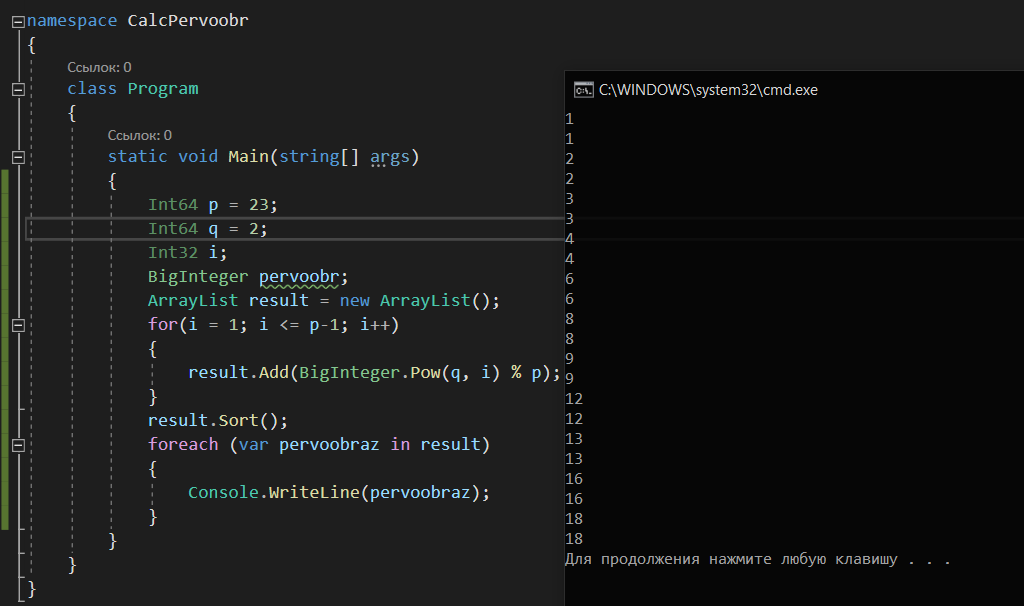
Так как мы получили все возможные взаимнопростые остатки p => число

является первообразным корнем 19.

Числа, являющиеся первообразными корнями **p =19**, являются **[2,3,10,13,14,15]**;

**2) p=23**

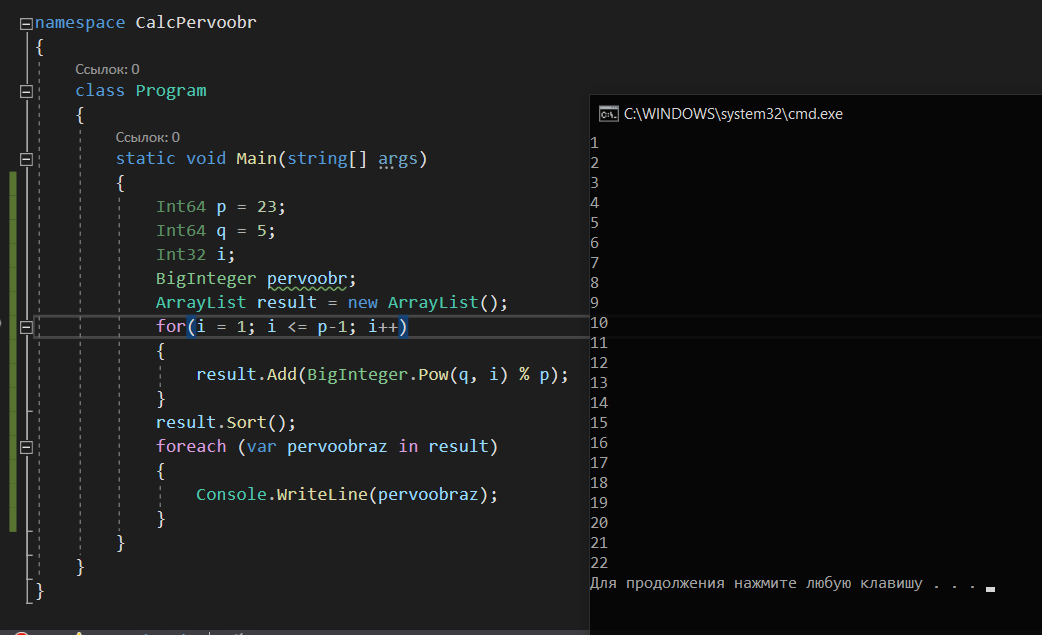
**q=2 - нет**

****

Так же делаем и с остальными числами.

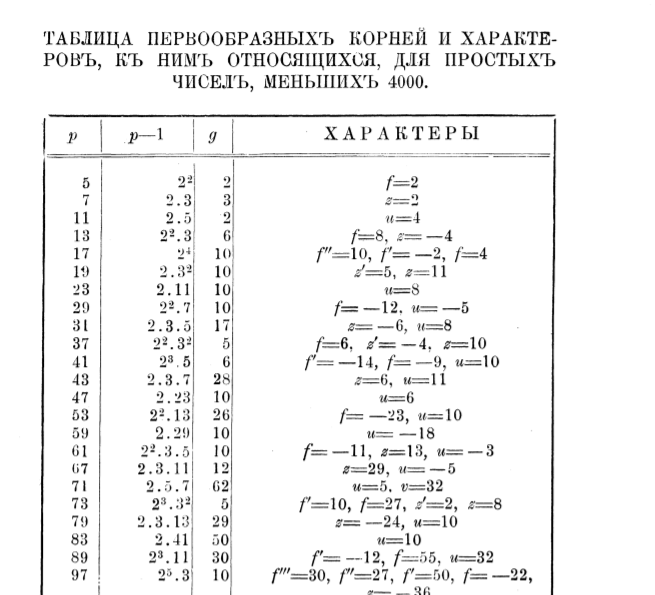
Вот первый попавшийся первообразный корень

**q=5-да**

****

Числа, являющиеся первообразными корнями **p =23**, являются **[5,7,10,11,14,15,17,19,20,21]**;

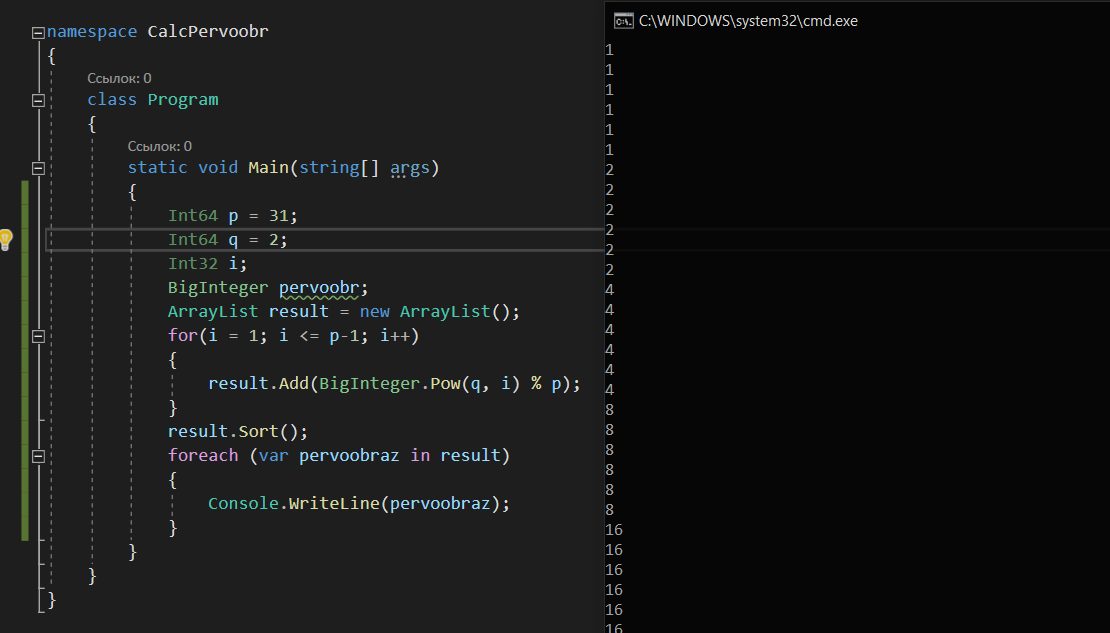
**3) p=27 –** у числа нет первообразных корней исходя из таблицы

****

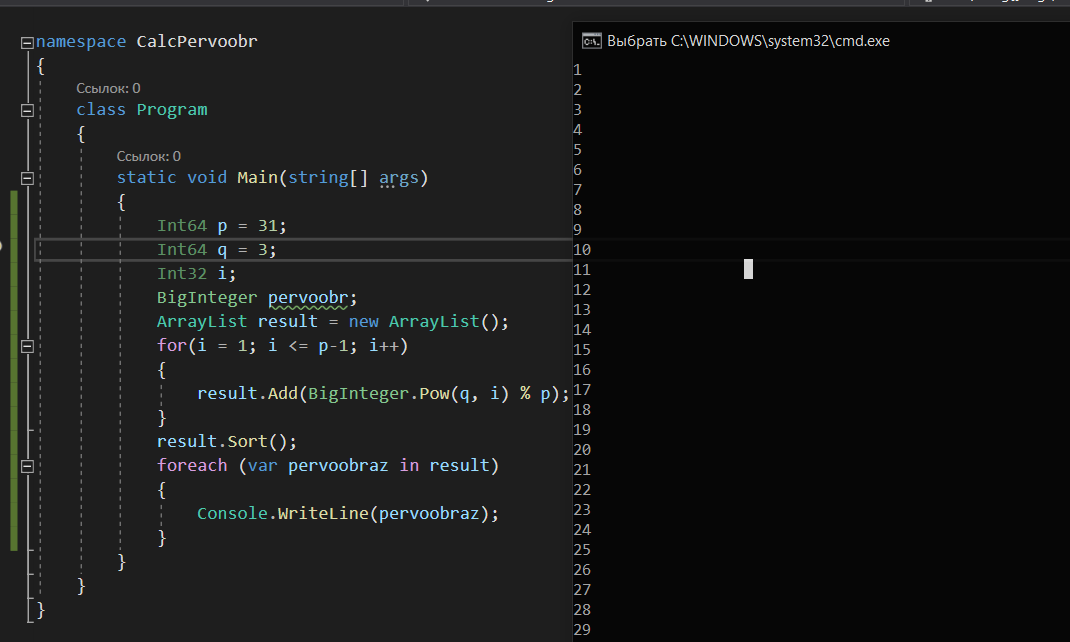
Число в ней отсутствует.

**4) p=31**

**q=2 – нет**

****

**q=3-да**

****

Числа, являющиеся первообразными корнями **p =31**, являются **[3,11,12,13,17,21,22,24]**;

**4) p=37**

Числа, являющиеся первообразными корнями **p =37**, являются **[2,13,15,17,18,19,20,22,24,32,35]**;

**3. Пользователю системы RSA с его собственными ключевыми параметрами n = 3337, е = 79 передано зашифрованное сообщение С, состоящее из трех блоков: чисел, соответствующих порядковым номерам Ваших фамилии, имени и отчества. Расшифровать это сообщение (взломав систему RSA пользователя).**

Блоки шифротекста соответствуют инициалам **Л**э **Н**икита **З**ыонгович =>

13, 15, 9.

Нужно определить исходное сообщение.

**N**= 3337

**e**= 79

**C**= (13,15,9)

**D**= 79^(-1) mod 3337= 1019

13^1019 mod 3337= 1543

15^1019 mod 3337 = 2061

9^1019 mod 3337 = 1236

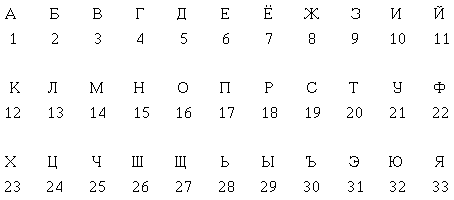
Исходное сообщение : 1543, 2061, 1236

**4. Положим, что в системе применяется алгоритм шифрования/ расшифрования Эль-Гамаля. Известны некоторые параметры системы: р = 167, g = 5, y = g^29 = 55 mod p.**

**Используя указанные и недостающие (выбрать самостоятельно) параметры, зашифровать свое имя (в любом языке) в предположении:**

**а) первая буква алфавита соответствует числу 0 и т. д.,**

**б) первая буква алфавита соответствует числу 1.**



Никита

**а)** 14 9 11 9 19 0

**б)** 15 10 12 10 20 1

p = 167, g=5, y= g^29 mod p = 55

x = 29 (закрытый ключ)

Для каждого блока сообщения выберем случайное число k (1<k<p-1)

k = 13

ai = g^k mod p

bi = (y^k \* mi) mod p

**a)**

**a** = 5^13 mod 167 = 92

**b = (**55^13 \* 14) mod 167 = 159

**b = (**55^13 \* 9) mod 167 = 138

**b = (**55^13 \* 11) mod 167 = 113

**b = (**55^13 \* 9) mod 167 = 138

**b = (**55^13 \* 19) mod 167 = 13

**b = (**55^13 \* 0) mod 167 = 0

(92, 159), (92, 138), (92, 113), (92, 138), (92, 13), (92, 0) – зашифрованное сообщение

**b)**

**b = (**55^13 \* 15) mod 167 = 63

**b = (**55^13 \* 10) mod 167 = 42

**b = (**55^13 \* 12) mod 167 = 17

**b = (**55^13 \* 10) mod 167 = 42

**b = (**55^13 \* 20) mod 167 = 84

**b = (**55^13 \* 1) mod 167 = 71

(92, 63), (92, 42), (92, 17), (92, 42), (92, 84), (92, 71) – зашифрованное сообщение

Шифровать надо с алфавита начинающийся на 1, потому что при 0 элементе шифроваться будет 0;

**5) Сколько попыток нужно сделать, чтобы с вероятностью более 0,5 (0,7; 0,8; 0,9) обнаружить коллизию при длине хеша (l) 64 (128; 256; 512) бит?**

Количество попыток для вероятности 0,5 мы можем посмотреть в данной таблице



**Рс(Аn) = 0.5**

Наша задача заключается в поиске условия, при котором выполняется тождество указанное сверху.

Р(Аn) ~~ ехр (-n^2 /2m) = 0.5

Переписываем в равенство (n^2 / 2m ) = ln50

Где n = (2m \* ln50 )^0.5

**m**=2^l, где l – длина хеша

**n =** ((2\*2^64)\*ln50)^0.5 = 5.1 \* 10^9

**1)** 0.5

l = 128, n = 2.2\* 10^19

l = 256, n = 4\*10^38

l = 512, n = 1.4\*10^77

**2)** 0.7

l = 64, n = 6.7\*10^9

l = 128, n = 2.86\*10^19

l = 256, n = 5.28\*10^38

l = 512, n = 1.79\*10^77

**3)** 0.8

l = 64, n = 7.7\*10^9

l = 128, n = 3.31\* 10^19

l = 256, n = 6.1\*10^38

l = 512, n = 2.08\*10^77

**4)** 0.9

l = 64, n = 9.2\*10^9

l = 128, n = 3.96\* 10^19

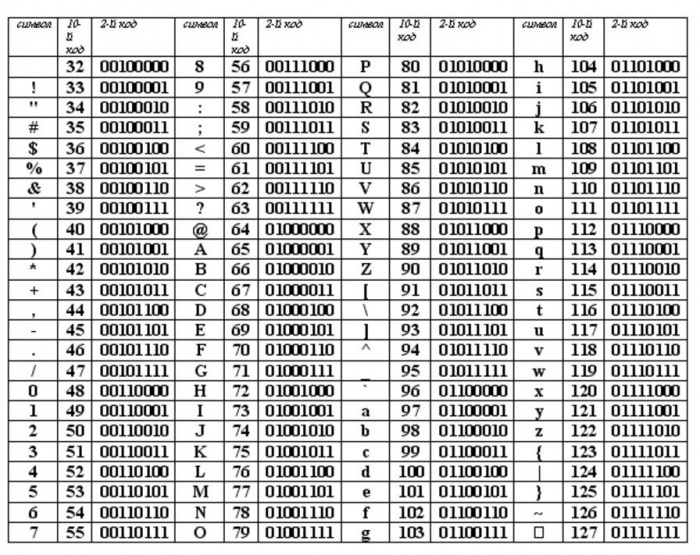
l = 256, n = 7.3\*10^19

l = 512, n = 2.48\*10^77

**7. Входное сообщение (прообраз) состоит из а) вашего имени, б) ваших фамилии\_имени\_отчества (алфавит – на свой выбор). Используя представление сообщения в кодах ASCII, представить в табличной форме (как выше в примере 5) содержание каждого 32- битного подблока расширенного входного сообщения**

**a)** Nikita

**b)** Le Nikita Ziongovich



**a)**  **Nikita** – 01001110 01101001 01101011 01101001 01110100 01100001

01001110 01101001 01101011 01101001

01110100 01100001 10000000 00000000

00000000 00000000 00000000 00000000

00000000 00000000 00000000 00000000

00000000 00000000 00000000 00000000

00000000 00000000 00000000 00000000

00000000 00000000 00000000 00000000

00000000 00000000 00000000 00000000

00000000 00000000 00000000 00000000

00000000 00000000 00000000 00000000

00000000 00000000 00000000 00000000

00000000 00000000 00000000 00000000

00000000 00000000 00000000 00000000

00000000 00000000 00000000 00000000

00000000 00000000 00000000 00000000

00000000 00000000 00000000 00110000

**b)**

**Le** – 01001100 01100101

**Nikita** – 01001110 01101001 01101011 01101001 01110100 01100001

**Ziongovich** – 01011010 01101001 01101111 01101110 01100111 01101111 01110110 01101001 01100011 01101000

01001100 01100101 01001110 01101001

01101011 01101001 01110100 01100001

01011010 01101001 01101111 01101110

01100111 01101111 01110110 01101001

01100011 01101000 10000000 00000000

00000000 00000000 00000000 00000000

00000000 00000000 00000000 00000000

00000000 00000000 00000000 00000000

00000000 00000000 00000000 00000000

00000000 00000000 00000000 00000000

00000000 00000000 00000000 00000000

00000000 00000000 00000000 00000000

00000000 00000000 00000000 00000000

00000000 00000000 00000000 00000000

00000000 00000000 00000000 00000000

00000000 00000000 00000000 10010000