Живец Федор Михайлович

3 курс 5 группа 1 подгруппа

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

В нашем случае S – сверхвозрастающая последовательность (т.к. каждый последующий элемент больше суммы предыдущих)

Сумма всех элементов W = 55203

Выберем простое число Q, превосходящее W: Q = 55210

Выберем число R из интервала [1, Qо]: R = 1233

S, R и Q – образуют закрытый ключ.

Сгенерируем открытый ключ B:

Si×R mod Q = Bi

103×1233 mod 55210= 16759

107×1233 mod 55210= 21511

211×1233 mod 55210= 39323

430×1233 mod 55210= 33300

863×1233 mod 55210= 15089

1716×1233 mod 55210= 17848

3449×1233 mod 55210= 1447

6907×1233 mod 55210= 13991

13807×1233 mod 55210= 19351

27610×1233 mod 55210= 33770

B = (16759, 21511, 39323, 33300, 15089, 17848, 1447, 13991, 19351, 33770) – открытый ключ

Соответственно для зашифрования нам нужно знать ключ B, а для расшифрования R и Q. (Только R мы берём как обратное по модулю Q, а именно 30717)

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

p – модуль

q – первообразная (q < p)

i – степень (1<=i<=p-1)

qimod p

p=19

q=2

21mod19 = 2

22mod19 = 4

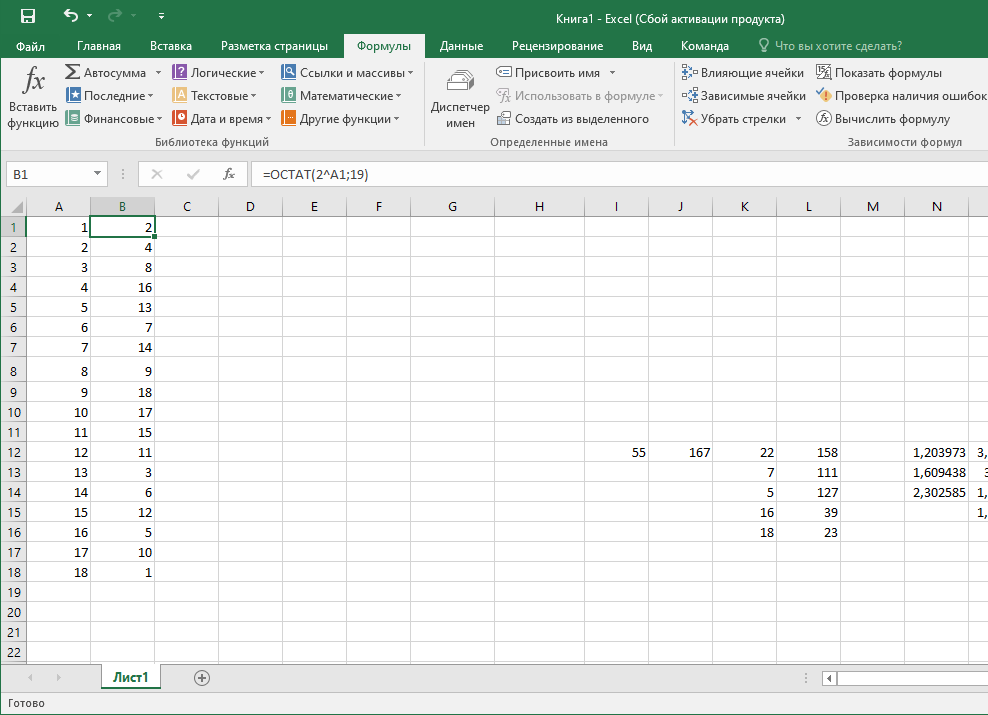
23mod19 = 8

24mod19 = 16

25mod19 = 13

…

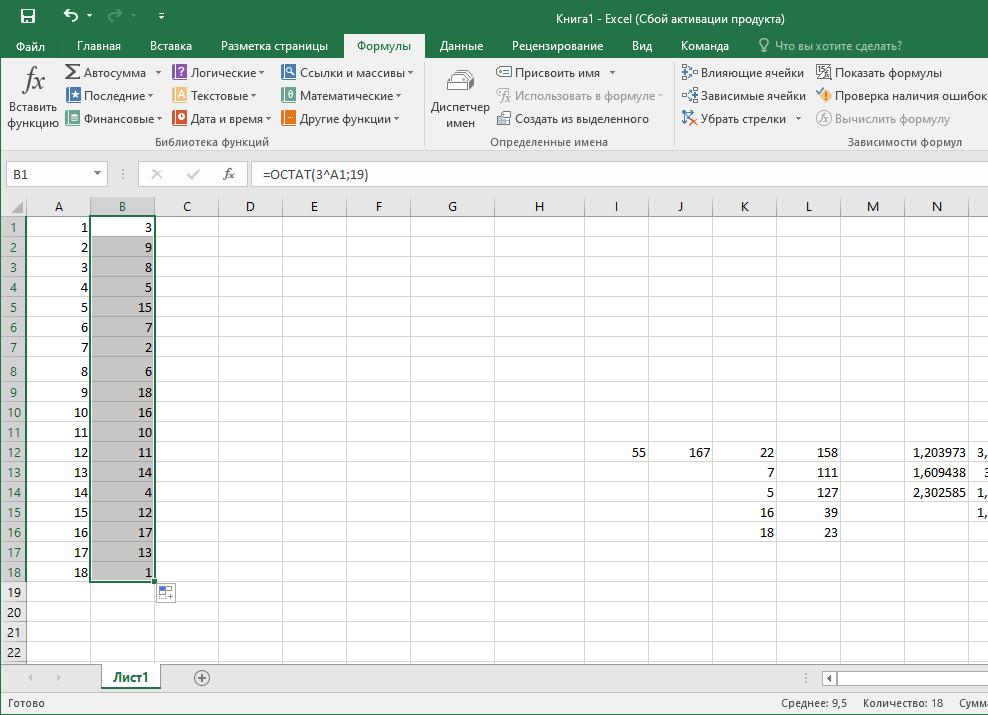
Для ускорения вычисления использовался Excel



Получили все возможные взаимнопростые с p остатки, следовательно число 2 является для 19 первообразным корнем.

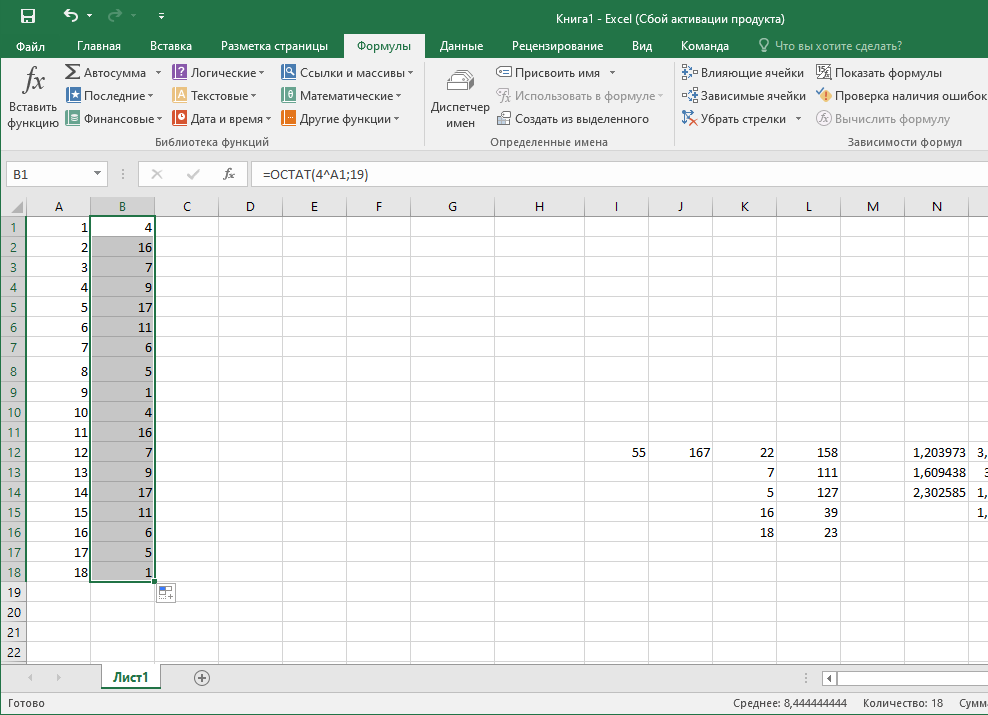
2 – да.

q=3



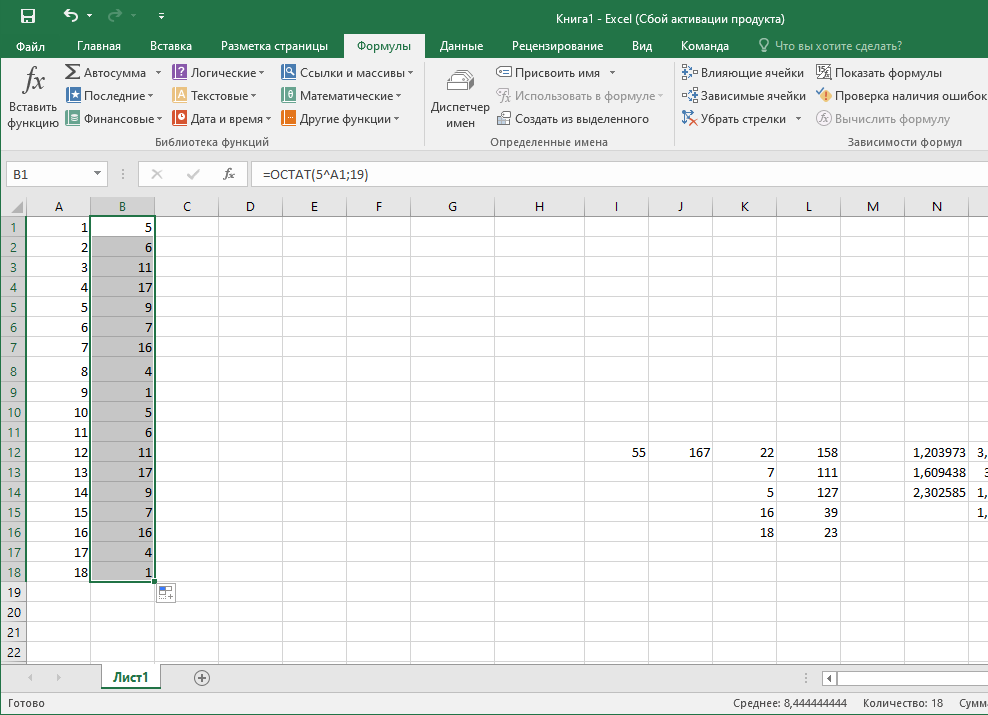
3 – да

q=4



4 – нет (для этого не хватает следующих чисел 2, 3, 8, 9, 12, 13, 14, 15)

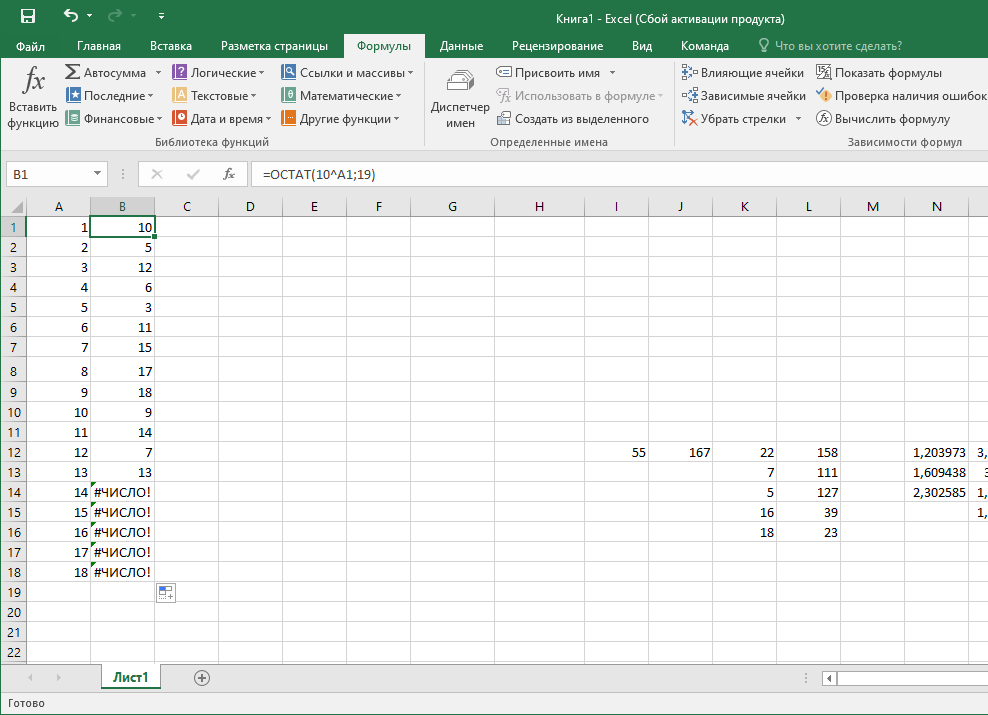
q=5



5 – нет

Далее все делается по аналогии. Ниже указаны числа, которые являются первообразными корнями

p = 19 q: (2, 3, 10, 13, 14, 15)

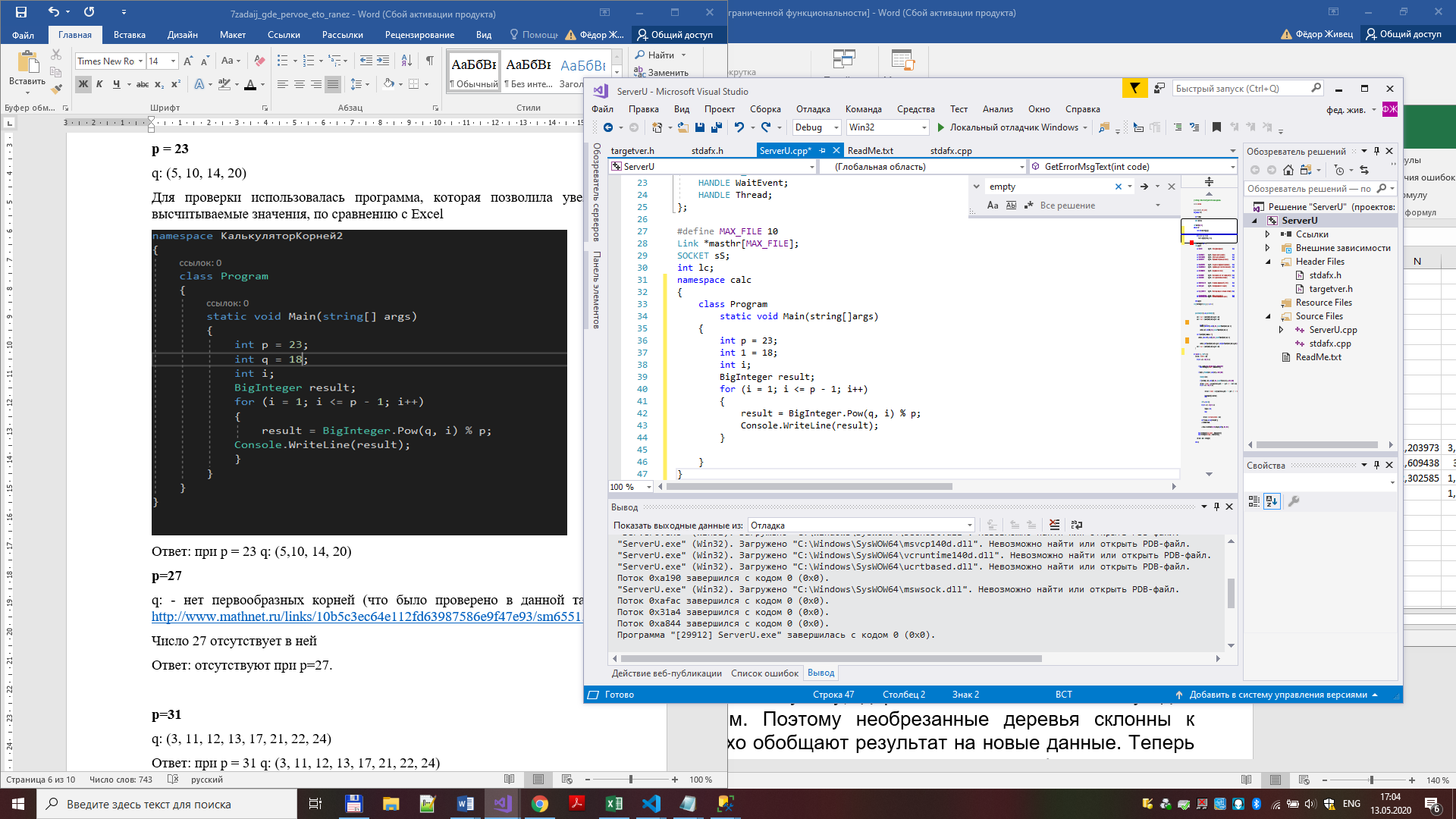


С определенных значений Excel не мог посчитать нужные значения из-за слишком больших чисел, поэтому приходилось далее использовать калькулятор

p = 23

q: (5, 7, 10, 11, 14, 15, 17, 19, 20, 21)

Для проверки использовалась программа, которая позволила увеличить высчитываемые значения, по сравнению с Excel



p=27

: - нет первообразных корней (что было проверено в данной таблице: [http://www.mathnet.ru/links/10b5c3ec64e112fd63987586e9f47e3/sm6551.pdf](http://www.mathnet.ru/links/10b5c3ec64e112fd63987586e9f47e93/sm6551.pdf),

Число 27 отсутствует в ней

p=31

q: (3, 11, 12, 13, 17, 21, 22, 24)

p=37

q: (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).

Например, пользователь получил блоки шифртекста (08, 22, 14 – соответствуют инициалам Живец Федор Михайлович. Нужно определить исходное сообщение.

Живец Федор Михайлович

Запишем сообщение из 3-х блоков: 08, 22, 14 – мой идентификатор, мой шифротекст.

n=3337

e=79

Высчитаем число d по формуле: d = e^-1 (mod n)

d=1019

Расшифруем наши блоки по формуле mi=(ci)dmod n, где m – исходное сообщение, а с – шифротекст.

1. (8)1019mod 3337 = 1469
2. (22)1019mod 3337 = 3310
3. (14)1019mod 3337 = 776

Исходное сообщение: 1469, 3310, 776.

4) Положим, что в системе применяется алгоритм шифрования/ расшифрования Эль-Гамаля. Известны некоторые параметры системы: р = 167, g = 5, y = g29 = 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.

Проанализировать результат.



Фёдор

а)21 6 4 15 17

б)22 7 5 16 18

p=167

g=5

y=55

x=29

Для каждого отдельного блока сообщения выбирается такое случайное число k, что: (1<k<p-1).

Блок шифротекста состоит из 2-х чисел, а и b.

ai = gk mod p

bi = (yk mi) mod p

1. 1) k1=7

m1=21

a1=136

b1=158

2) k2=5

m2=6

a­2=119

b2=41

3) k3=2

m3=4

a­3=25

b3=95

4) k4=6

m4= 15

a­4=94

b4=25

5) k5=4

m5=17

a­5=124

b5=152

са: ([136,166], [119, 59], [25, 76], [94, 13], [124, 2])

б)   
1) k1=7

m1=22

a1=136

b1=158

2) k2=5

m2=7

a­2=119

b2=41

3) k3=2

m3=5

a­3=25

b3=95

4) k4=6

m4= 16

a­4=94

b4=25

5) k5=4

m5=18

a­5=124

b5=152

са: ([136, 158], [119, 41], [25, 95], [94, 25], [124, 152])

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, сами же расчеты будут вестись следующим образом:

Итак, наш случай ограничивается частной задачей поиска условия, при котором выполняется тождество Рс(Аn) = 0.5

Или же Р(Аn) ≈ ехр (-n^2 /2m) = 0.5.

Что можно записать в виде равенства (n^2 /2m) = ln 50

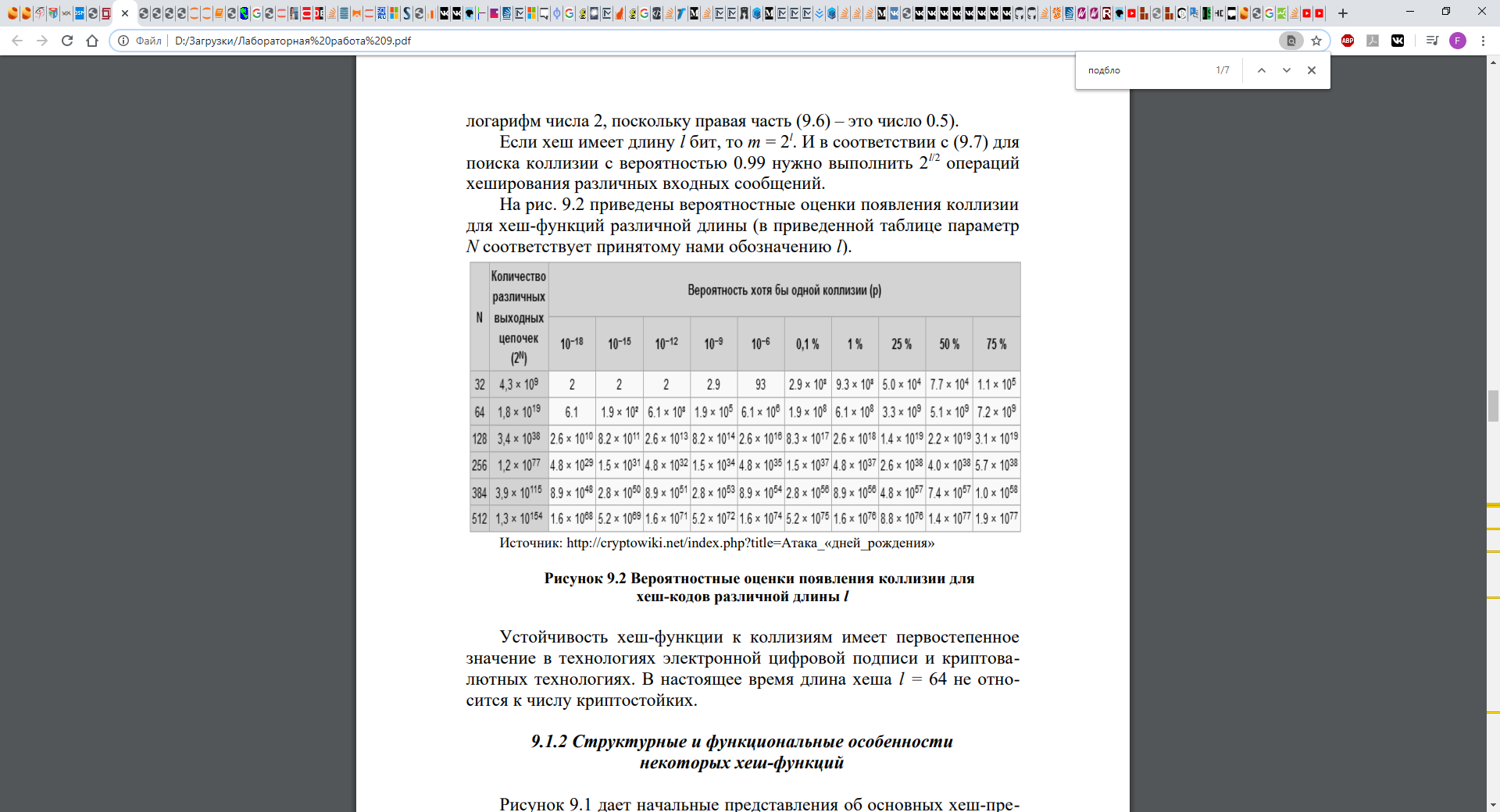
Откуда мы находим n = (2m\*ln 50)^0.5

m=2^l

l = длина хеша

Следовательно, получаем

n = ((2\*2^64)\*ln 50)^0.5 = 5,1 \* 10^9



По аналогии получаем значения и для других значений

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

6. Рассчитать общую длину (L') хешируемого сообщения после предварительной стадии на основе алгоритма MD, если объем (L) исходного сообщения составлял: 0; 484; 512; 1000; 2000; 16000 бит. Какова в каждом случае будет длина хеша?

Входное сообщение «дополняется» (расширяется) так, чтобы его длина (в битах) была конгруэнтной к размеру бит по модулю 512. Это значит, что сообщение начальной длиной L бит расширяется так, что остаются незаполненными всего лишь 64 бита, чтобы итоговая длина L' была кратной 512. В указанные 64 бита записывается двоичная длина L

0 – длина сообщения 0 так как ничего не передаем

484 – при добавлении в конце 64 битов мы получаем сообщение размером 548, для того чтобы оно было кратно 512 увеличим его на 476 до 1024

512 – при добавлении 64 битов получаем 576, для кратности увеличиваем сообщение до 1024

1000 – при добавлении 64 битов получаем 1064, для кратности увеличиваем сообщение до 1536

2000 – при добавлении 64 битов получаем 2064, для кратности увеличиваем сообщение до 2560

16000 – при добавлении 64 битов получаем 16064, для кратности увеличиваем сообщение до 16384

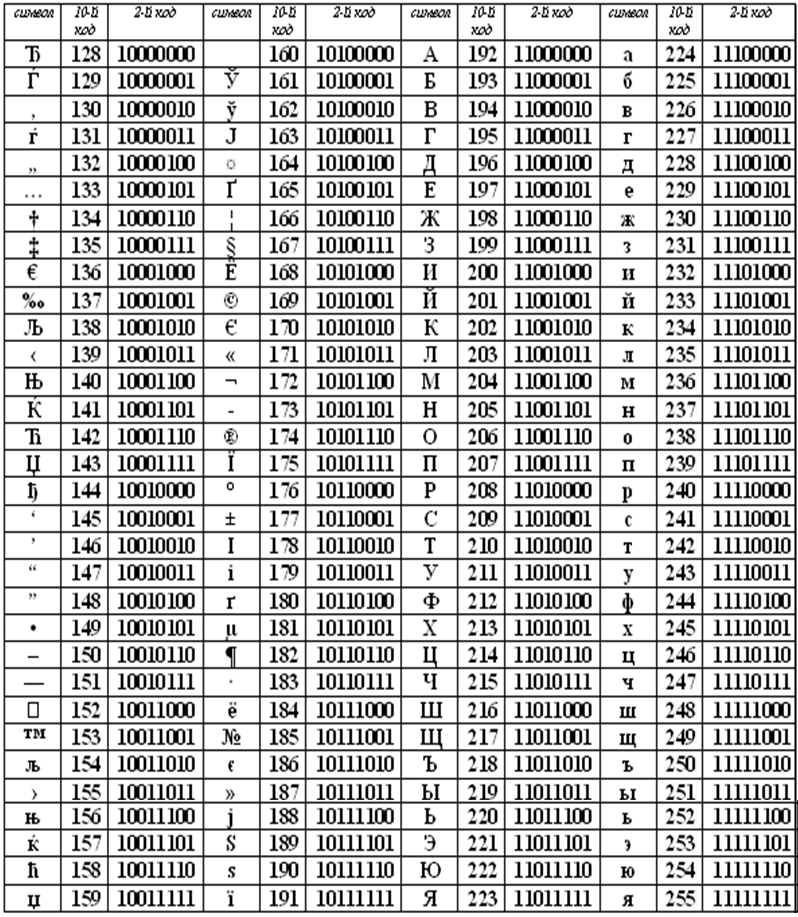
7. Входное сообщение (прообраз) состоит из а) вашего имени, б) ваших фамилии\_имени\_отчества (алфавит – на свой выбор). Используя представление сообщения в кодах ASCII, представить в табличной форме (как выше в примере 5) содержание каждого 32- битного подблока расширенного входного сообщения

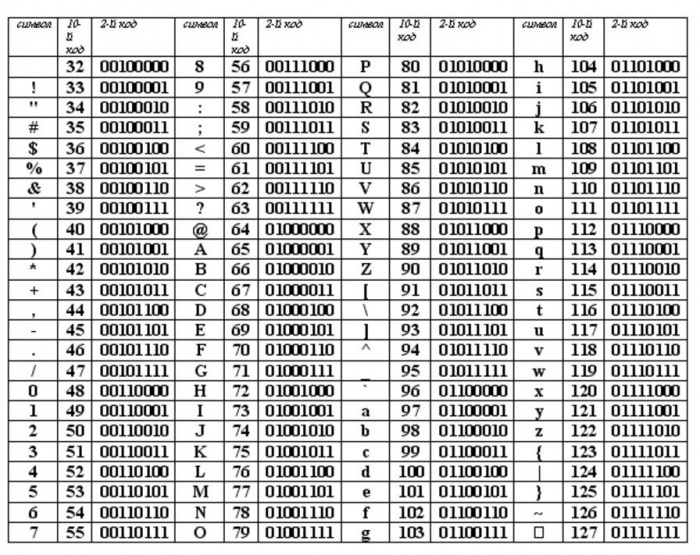
А) Федор

Б) Живец\_Федор\_Михайлович

Федор в кодах Ascii представлен в формате 198 232 226 229 246

Живец\_Федор\_Михайлович в виде 198 232 226 229 246 – 40 - 212 229 228 238 240 – 40 – 204 232 245 224 233 235 238 226 232 247



Значение для символа \_ бралось из таблицы ниже (код символа 95)

Соответствующие двоичные

Федор – 11010100-11100101-11100100-11101110-11110000

Живец\_Федор\_Михайлович

11000110-11101000-11100010-11100101-11110110-

1011111- 11010100-11100101-11100100-11101110-11110000-1011111-

11001100-11101000-11110101-11100000-11101001-11101011-11101110-11100010-11101000-11110111

В последнюю часть из 64 бит полученного 512-битного модуля (L' = 512) мы запишем справа двоичное представление числа L = 40: 101000. В остальные разряды мы впишем «0». После этого полученный расширенный блок делим на 16-ти 32-разрядных подблоков:

Федор

11010100111001011110010011101110

11110000111100000000000000000000

00000000000000000000000000000000

00000000000000000000000000000000

00000000000000000000000000000000

00000000000000000000000000000000

00000000000000000000000000000000

00000000000000000000000000000000

00000000000000000000000000000000

00000000000000000000000000000000

00000000000000000000000000000000

00000000000000000000000000000000

00000000000000000000000000000000

00000000000000000000000000000000

00000000000000000000000000000000

00000000000000000000000000101000

Живец\_Федор\_Михайлович

В последнюю часть из 64 бит полученного 512-битного модуля (L' = 512) мы запишем справа двоичное представление числа L = 174: 10101110. В остальные разряды мы впишем «0». После этого полученный расширенный блок делим на 16-ти 32-разрядных подблоков

11000110111010001110001011100101

11110110101111111010100111001011

11001001110111011110000101111111

00110011101000111101011110000011

10100111101011111011101110001011

10100011110111000000000000000000

00000000000000000000000000000000

00000000000000000000000000000000

00000000000000000000000000000000

00000000000000000000000000000000

00000000000000000000000000000000

00000000000000000000000000000000

00000000000000000000000000000000

00000000000000000000000000000000

00000000000000000000000000000000

00000000000000000000000010101110