

1. Составить список из заглавных букв английского алфавита (в кодировке Unicode), посчитать среднее значение кодов.

```
In[17]:= listUpperEng = CharacterRange["A", "Z"]
codeListUpperEng = ToCharacterCode[listUpperEng]
N[Total[codeListUpperEng]/Length[codeListUpperEng]]

Out[17]= {A, B, C, D, E, F, G, H, I, J, K, L, M, N, O, P, Q, R, S, T, U, V, W, X, Y, Z}

Out[18]= {{65}, {66}, {67}, {68}, {69}, {70}, {71}, {72}, {73}, {74}, {75}, {76},
{77}, {78}, {79}, {80}, {81}, {82}, {83}, {84}, {85}, {86}, {87}, {88}, {89}, {90}}

Out[19]= {77.5}
```

2. Дано число 358 138 [number], найти его обратное число [res1] в конечном поле GF (6 502 301) [GF (div)].

```
In[156]:= number = 358 138
GF = 6 502 301
inverseNumber = PowerMod[number, -1, GF]
Mod[inverseNumber * number, GF](*проверка на то,
что число и правда обратное. НЕ ОТВЕТ!*)

Out[156]= 358 138

Out[157]= 6 502 301

Out[158]= 4 060 720

Out[159]= 1
```

3. Дано число [start], найти два ближайших к нему простых числа (b1, b2). Посчитать их сумму по модулю [mod]. (Если два ближайших числа, больших start, то параметры NextPrime + 1, +2)

In[174]:=

```

start = 4246
mod = 147
b1 = NextPrime[start, -1]
b2 = NextPrime[start, 1]
summ = Mod[b1 + b2, mod]

```

Out[174]=

4246

Out[175]=

147

Out[176]=

4243

Out[177]=

4253

Out[178]=

117

4. Найти количество разрядов в двоичной записи шестнадцатеричного числа 33 A855 [number].

In[210]:=

```

hexNumber = 16^^33A855
binaryNumber = BaseForm[hexNumber, 2](*проверка,
чтобы увидеть что за число. НЕ ОТВЕТ!*)
IntegerLength[binaryNumber, 2]

```

Out[210]=

3 385 429

Out[211]//BaseForm=

11001111010100001010101₂

Out[212]=

22

In[394]:=

```

(*Функция, выдающая список простых чисел в диапазоне от min до max*)
primeRange[min_, max_] := Module[{list = {}},
  If[PrimeQ[min], newNum = min, newNum = NextPrime[min, 1]];
  While[newNum ≤ max, AppendTo[list, newNum];
  newNum = NextPrime[newNum, 1]];
  list]

```

5. Найти количество простых чисел в диапазоне от 17 [R1] до 29 [R2].

```

{2, 3, 5, 7, 11, 13, 17, 19, 23, 29}
{17, 19, 23, 29}

```

In[395]:=

```

R1 = 17
R2 = 29
PrimePi[R2] - PrimePi[R1 - 1]
(*ну или*)
listPrime = primeRange[R1, R2]
Length[listPrime]

```

Out[395]=

17

Out[396]=

29

Out[397]=

4

Out[398]=

{17, 19, 23, 29}

Out[399]=

4

6. Получить множество целых чисел от $[min1]$ до $[max1]$ и множество простых чисел от $[min2]$ до $[max2]$. Найти произведение элементов пересечения этих множеств.

In[356]:=

```

min1 = 65
max1 = 105
min2 = 80
max2 = 145
mult = 1
listIntegerInRange = Range[min1, max1]
listPrimeInRange = primeRange[min2, max2]
listIntersection = Intersection[listIntegerInRange, listPrimeInRange]
Do[mult = mult * listIntersection[[i]], {i, 1, Length[listIntersection]}]
mult

```

Out[356]=

65

Out[357]=

105

Out[358]=

80

Out[359]=

145

Out[360]=

1

Out[361]=

```

{65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86,
 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100, 101, 102, 103, 104, 105}

```

Out[362]=

```

{83, 89, 97, 101, 103, 107, 109, 113, 127, 131, 137, 139}

```

Out[363]=

```

{83, 89, 97, 101, 103}

```

Out[365]=

7 454 155 217

7. Составить список из нечетных чисел в диапазоне от n_{min} до n_{max} . Сделать квадратную матрицу. Поменять местами строки $str1$ и $str2$. Циклично сдвинуть строку $str3$ вправо (влево) на $len1$. Поменять местами столбцы $col1$ и $col2$. Циклично сдвинуть столбец $col3$ вверх (вниз) на $len2$. Посчитать сумму элементов главной диагонали. (Получить список из первых $count$ простых чисел)

In[2029]:=

```

nmin = 1
nmax = 49

str1 = 2
str2 = 3

```

```

str3 = 4
len1 = 41

col1 = 4
col2 = 2

col3 = 3
len2 = 1

sum = 0

rangeOdd[min_, max_] :=
  Module[{list = {}}, Do[If[Mod[i, 2] ≠ 0, AppendTo[list, i]], {i, min, max}];
  list]

listOddNumbers = rangeOdd[nmin, nmax]

matrix = Partition[listOddNumbers, Sqrt[Length[listOddNumbers]] // MatrixForm

(*swapString[listOddNumbers_,str1_,str2_] := Module[{buf,matrix},
matrix = Partition[listOddNumbers, Sqrt[Length[listOddNumbers]]];
buf=matrix[[1,str1,All]];
matrix[[1,str1,All]]=matrix[[1,str2,All]];
matrix[[1,str2,All]]=buf;
matrix]

matrix=swapString[listOddNumbers,str1,str2]//MatrixForm*)

bufstr = matrix[[1, str1, All]]
matrix[[1, str1, All]] = matrix[[1, str2, All]]
matrix[[1, str2, All]] = bufstr
matrix

matrix[[1, str3]] = RotateRight[matrix[[1, str3]], len1]
matrix

bufcol = {}
Do[AppendTo[bufcol, matrix[[1, i, col1]]], {i, 1, Length[matrix[[1]]]}]
Do[Part[matrix, 1, i, col1] = Part[matrix, 1, i, col2], {i, 1, Length[matrix[[1]]]}]
Do[Part[matrix, 1, i, col2] = bufcol[[i]], {i, 1, Length[matrix[[1]]]}]

```

```
matrix
```

```
bufcol = {}
```

```
Do[AppendTo[bufcol, matrix[[1, i, col3]], {i, 1, Length[matrix[[1]]}]]
```

```
bufcol = RotateRight[bufcol, len2]
```

```
Do[Part[matrix, 1, i, col3] = bufcol[[i]], {i, 1, Length[matrix[[1]]}]]
```

```
matrix
```

```
Do[sum = sum + Part[matrix, 1, i, i], {i, 1, Length[matrix[[1]]}]]
```

```
sum
```

```
Out[2029]=
```

```
1
```

```
Out[2030]=
```

```
49
```

```
Out[2031]=
```

```
2
```

```
Out[2032]=
```

```
3
```

```
Out[2033]=
```

```
4
```

```
Out[2034]=
```

```
41
```

```
Out[2035]=
```

```
4
```

```
Out[2036]=
```

```
2
```

```
Out[2037]=
```

```
3
```

```
Out[2038]=
```

```
1
```

```
Out[2039]=
```

```
0
```

```
Out[2041]=
```

```
{1, 3, 5, 7, 9, 11, 13, 15, 17, 19, 21,  
 23, 25, 27, 29, 31, 33, 35, 37, 39, 41, 43, 45, 47, 49}
```

```
Out[2042]//MatrixForm=
```

$$\begin{pmatrix} 1 & 3 & 5 & 7 & 9 \\ 11 & 13 & 15 & 17 & 19 \\ 21 & 23 & 25 & 27 & 29 \\ 31 & 33 & 35 & 37 & 39 \\ 41 & 43 & 45 & 47 & 49 \end{pmatrix}$$

```
Out[2043]=
```

```
{11, 13, 15, 17, 19}
```

Out[2044]=

{21, 23, 25, 27, 29}

Out[2045]=

{11, 13, 15, 17, 19}

Out[2046]//MatrixForm=

$$\begin{pmatrix} 1 & 3 & 5 & 7 & 9 \\ 21 & 23 & 25 & 27 & 29 \\ 11 & 13 & 15 & 17 & 19 \\ 31 & 33 & 35 & 37 & 39 \\ 41 & 43 & 45 & 47 & 49 \end{pmatrix}$$

Out[2047]=

{39, 31, 33, 35, 37}

Out[2048]//MatrixForm=

$$\begin{pmatrix} 1 & 3 & 5 & 7 & 9 \\ 21 & 23 & 25 & 27 & 29 \\ 11 & 13 & 15 & 17 & 19 \\ 39 & 31 & 33 & 35 & 37 \\ 41 & 43 & 45 & 47 & 49 \end{pmatrix}$$

Out[2049]=

{}

Out[2053]//MatrixForm=

$$\begin{pmatrix} 1 & 7 & 5 & 3 & 9 \\ 21 & 27 & 25 & 23 & 29 \\ 11 & 17 & 15 & 13 & 19 \\ 39 & 35 & 33 & 31 & 37 \\ 41 & 47 & 45 & 43 & 49 \end{pmatrix}$$

Out[2054]=

{}

Out[2056]=

{45, 5, 25, 15, 33}

Out[2058]//MatrixForm=

$$\begin{pmatrix} 1 & 7 & 45 & 3 & 9 \\ 21 & 27 & 5 & 23 & 29 \\ 11 & 17 & 25 & 13 & 19 \\ 39 & 35 & 15 & 31 & 37 \\ 41 & 47 & 33 & 43 & 49 \end{pmatrix}$$

Out[2060]=

133

8. Установить генератор случайных чисел в начальное состояние с параметром (param) $2^{15} \pmod{311}$. Получить список из 10 случайных простых чисел в диапазоне от 46 до 77. Найти произведение двух простых чисел, встречающихся в списке с максимальной (maxfr) и минимальной (minfr) частотами. В случае наличия чисел с одинаковыми частотами, выбирать первые в списке.

```
GF = 311
```

```
min = 46
```

```
max = 77
```

```
param = PowerMod[2, 15, GF]
```

```
SeedRandom[param]
```

```
listRandomPrime = RandomPrime[{min, max}, 10]
```

```
listTally = Tally[listRandomPrime]
```

```
listMAX = {}
```

```
Do[If[listTally[[i, 2]] == Max[listTally[[All, 2]]], AppendTo[listMAX, listTally[[i, 1]]],  
  {i, 1, Length[listTally]}]
```

```
listMAX
```

```
listMIN = {}
```

```
Do[If[listTally[[i, 2]] == Min[listTally[[All, 2]]], AppendTo[listMIN, listTally[[i, 1]]],  
  {i, 1, Length[listTally]}]
```

```
listMIN
```

```
listMAX[[1]] * listMIN[[1]]
```

```
Out[2070]=
```

```
311
```

```
Out[2071]=
```

```
46
```

```
Out[2072]=
```

```
77
```

```
Out[2073]=
```

```
113
```

```
Out[2074]=
```

```
RandomGeneratorState[  
  Method: ExtendedCA  
  State hash: 7806026626613991272  
]
```

```
Out[2075]=
```

```
{73, 67, 73, 61, 47, 47, 61, 59, 61, 73}
```

```
Out[2076]=
```

```
{{73, 3}, {67, 1}, {61, 3}, {47, 2}, {59, 1}}
```

**9. Найти значение функции Эйлера для числа x ,
которое определяется из соотношения : $a * x + b = c \pmod{n}$,
где $a = 34\,535$, $b = 34\,745$, $c = 26\,341$, $n = 11\,047$.**

In[2094]:=

```
a = 34 535
b = 34 745
c = 26 341
n = 11 047
x = Mod[Mod[Mod[c, n] - Mod[b, n], n] * PowerMod[a, -1, n], n]
euler = EulerPhi[x]
```

Out[2094]=

34 535

Out[2095]=

34 745

Out[2096]=

26 341

Out[2097]=

11 047

Out[2098]=

517

Out[2099]=

460

10. Определить ожидаемое время раскрытия пароля

длиной (length) символов и содержащего следующие наборы :

{цифры, строчные русские, строчные латинские, прописные латинские},

если скорость перебора пароля (в символах в секунду) (V) равна обратному

элементу числа 1939 по модулю 661. Ответ вводить как целое число суток.

In[2213]:=

```

length = 8
number = 2825
GF = 503
speed = PowerMod[number, -1, GF]

numbersLen = Length[Range[0, 9]]
rusLen = Length[CharacterRange["а", "я"]]
engLen = Length[CharacterRange["a", "z"]]

alphabetLength = numbersLen + rusLen + engLen + engLen

time = (1/2) * (Power[alphabetLength, length]) / (speed * 60 * 60 * 24)
N[time, 15]

```

Out[2213]=

8

Out[2214]=

2825

Out[2215]=

503

Out[2216]=

86

Out[2217]=

10

Out[2218]=

32

Out[2219]=

26

Out[2220]=

94

Out[2221]=

$$\frac{23\,811\,286\,661\,761}{58\,050}$$

Out[2222]=

$$4.10185816740069 \times 10^8$$

11. Архив текстового файла archive – 118. zip защищен паролем из 4 символов, содержащих строчные и заглавные латинские буквы, а также все цифры. Один из символов пароля можно определить из следующего условия : полусумма кода символа и кода позиции символа в пароле равна 69, полуразность кода символа и кода позиции символа равна 20. Исключить пробелы и подсчитать число символов в тексте.

Допустим взломали ... получили текстовый файл ...

In[2283]:=

```

text = "- Откуда у вас столько воды в
        дипломной работе? - Я использовал несколько источников."
textWithoutSpaces = StringReplace[text, " " -> ""]
StringLength[text]

```

Out[2283]=

```

- Откуда у вас столько воды в
  дипломной работе? - Я использовал несколько источников.

```

Out[2284]=

```

-Откудаувавастольководывдипломнойработе?-Яиспользовалнесколькоисточников.

```

Out[2285]=

85

12. Скачайте с сетевого диска (ftp – сервера) файл text – 02. txt, расположенный в папке Texts и определите энтропию сообщения, содержащегося в нем. Ответ представить в битах, с 5 знаками после запятой $N[\dots, 6]$.

In[2325]:=

```

text = "- Откуда у вас столько воды в
        дипломной работе? - Я использовал несколько источников."

```

```

text = ToCharacterCode[text]
entropy = N[Entropy[2, text], 6]

```

Out[2325]=

```

- Откуда у вас столько воды в
  дипломной работе? - Я использовал несколько источников.

```

Out[2326]=

```

{45, 32, 1054, 1090, 1082, 1091, 1076, 1072, 32, 1091, 32, 1074, 1072, 1089, 32,
  1089, 1090, 1086, 1083, 1100, 1082, 1086, 32, 1074, 1086, 1076, 1099, 32, 1074,
  32, 1076, 1080, 1087, 1083, 1086, 1084, 1085, 1086, 1081, 32, 1088, 1072, 1073,
  1086, 1090, 1077, 63, 32, 45, 32, 1071, 32, 1080, 1089, 1087, 1086, 1083, 1100,
  1079, 1086, 1074, 1072, 1083, 32, 1085, 1077, 1089, 1082, 1086, 1083, 1100,
  1082, 1086, 32, 1080, 1089, 1090, 1086, 1095, 1085, 1080, 1082, 1086, 1074, 46}

```

Out[2327]=

4.24670

13. Определите энтропию сектора с номером 795 виртуального флорпи – диска flptest.flp с точностью 5 знаков после запятой. Для округления результата применять функцию $N[,]$. Пример ввода : 5.55555.

(*Используя WinHex исследовать размещение информации на дискете: выделив курсором стертый блок и используя меню Правка\Copy Block запомнить в отдельном файле содержимое этого блока Для ввода данных использовать предварительное преобразование с помощью утилиты Converter.exe и последующей функции ReadList["file.dat",Number]

В РЕАЛЬНОСТИ НАДО ТАК!

$N[2, \text{Entropy}[\text{ReadList}["file.dat", \text{Number}]], 8]$

***)**

In[2331]:=

text = "Мне коллега сказала, что у её мужа совпадают айкью и вес. Вот сижу и думаю, что же она имела в виду. То ли её муж жирный ботаник, то ли худой дебил."

text = ToCharacterCode[text]

entropy = N[Entropy[2, text], 6]

Out[2331]=

Мне коллега сказала, что у её мужа совпадают айкью и вес. Вот сижу и думаю, что же она имела в виду. То ли её муж жирный ботаник, то ли худой дебил.

Out[2332]=

{1052, 1085, 1077, 32, 1082, 1086, 1083, 1083, 1077, 1075, 1072, 32, 1089, 1082, 1072, 1079, 1072, 1083, 1072, 44, 32, 1095, 1090, 1086, 32, 1091, 32, 1077, 1105, 32, 1084, 1091, 1078, 1072, 32, 1089, 1086, 1074, 1087, 1072, 1076, 1072, 1102, 1090, 32, 1072, 1081, 1082, 1100, 1102, 32, 1080, 32, 1074, 1077, 1089, 46, 32, 1042, 1086, 1090, 32, 1089, 1080, 1078, 1091, 32, 1080, 32, 1076, 1091, 1084, 1072, 1102, 44, 32, 1095, 1090, 1086, 32, 1078, 1077, 32, 1086, 1085, 1072, 32, 1080, 1084, 1077, 1083, 1072, 32, 1074, 32, 1074, 1080, 1076, 1091, 46, 32, 1058, 1086, 32, 1083, 1080, 32, 1077, 1105, 32, 1084, 1091, 1078, 32, 1078, 1080, 1088, 1085, 1099, 1081, 32, 1073, 1086, 1090, 1072, 1085, 1080, 1082, 44, 32, 1090, 1086, 32, 1083, 1080, 32, 1093, 1091, 1076, 1086, 1081, 32, 1076, 1077, 1073, 1080, 1083, 46}

Out[2333]=

4.34650

14. При стартовом значении генератора равном start сформировать последовательность, состоящую из count случайных чисел, лежащих в диапазоне $[\text{min}, \text{max}]$. Найти произведение элементов последовательности, принадлежащих подмножеству, содержащему четырехкратную коллизию. В поле ответа ввести количество разрядов для шестнадцатеричного представления полученного произведения.

In[2427]:=

```

start = 34
count = 879
min = 21
max = 869
mult = 1

```

```
SeedRandom[start]
```

```
listRandom = RandomInteger[{min, max}, count]
```

```
listTally = Tally[listRandom]
```

```
For[i = 1, i < Length[listTally], i++,
```

```
  If[listTally[[i, 2]] == 5, mult = mult * listTally[[i, 1]]]
```

```
mult
```

```
Length[IntegerDigits[mult, 16]]
```

Out[2427]=

```
34
```

Out[2428]=

```
879
```

Out[2429]=

```
21
```

Out[2430]=

```
869
```

Out[2431]=

```
1
```

Out[2432]=

```
RandomGeneratorState[ Method: ExtendedCA
                      State hash: -4858488829994941523 ]
```

Out[2433]=

```

{829, 358, 290, 489, 315, 126, 623, 189, 158, 855, 269, 108, 311, 441, 174, 763, 450, 317,
 419, 36, 231, 502, 453, 454, 861, 513, 70, 260, 777, 74, 654, 307, 374, 153, 494,
 793, 83, 797, 564, 859, 453, 743, 810, 325, 781, 431, 240, 179, 592, 337, 120, 145,
 817, 250, 867, 818, 428, 601, 236, 541, 514, 215, 337, 484, 733, 817, 622, 476,
 236, 702, 557, 437, 846, 814, 470, 812, 314, 716, 836, 580, 356, 705, 691, 36, 446,
 522, 722, 767, 639, 861, 31, 470, 558, 494, 712, 27, 244, 163, 685, 240, 792, 586,
 229, 253, 172, 440, 327, 355, 505, 415, 150, 361, 168, 573, 108, 715, 792, 117,
 678, 287, 593, 60, 255, 51, 788, 390, 259, 649, 41, 754, 487, 749, 510, 73, 692,
 674, 408, 271, 462, 718, 458, 661, 575, 688, 296, 810, 307, 828, 833, 786, 762,
 344, 536, 227, 633, 671, 38, 253, 25, 396, 707, 354, 795, 834, 279, 545, 724, 238,
 568, 69, 599, 120, 421, 404, 245, 533, 747, 863, 621, 356, 405, 224, 565, 835,
 382, 366, 400, 713, 333, 374, 795, 466, 303, 829, 311, 820, 431, 365, 387, 651,
 44, 595, 54, 252, 789, 839, 162, 731, 302, 144, 473, 796, 27, 713, 565, 461, 638,

```

255, 353, 99, 628, 418, 389, 588, 173, 831, 28, 572, 184, 483, 375, 83, 537, 701, 461, 364, 456, 857, 116, 699, 465, 782, 597, 540, 175, 399, 281, 740, 289, 200, 366, 217, 122, 709, 413, 370, 592, 571, 440, 378, 827, 349, 269, 733, 674, 343, 471, 393, 399, 735, 696, 544, 660, 238, 439, 609, 104, 231, 581, 31, 388, 849, 404, 574, 484, 460, 92, 347, 207, 105, 570, 635, 310, 528, 237, 345, 616, 209, 122, 256, 266, 456, 798, 742, 576, 385, 488, 80, 192, 839, 452, 114, 441, 287, 221, 115, 180, 310, 714, 631, 754, 769, 235, 259, 732, 765, 312, 394, 68, 641, 194, 171, 609, 585, 632, 654, 656, 359, 661, 791, 502, 299, 397, 728, 334, 215, 457, 247, 536, 26, 433, 527, 294, 268, 561, 337, 352, 867, 274, 850, 609, 768, 726, 32, 469, 215, 553, 371, 231, 516, 546, 531, 355, 471, 121, 824, 526, 140, 768, 368, 262, 219, 84, 426, 347, 741, 489, 809, 111, 735, 221, 350, 825, 401, 424, 107, 230, 57, 460, 228, 65, 438, 827, 699, 308, 412, 97, 743, 517, 539, 792, 681, 566, 622, 117, 346, 179, 811, 243, 632, 742, 255, 753, 357, 482, 236, 794, 474, 704, 714, 542, 106, 518, 258, 194, 294, 278, 73, 451, 621, 788, 230, 548, 417, 72, 94, 647, 232, 752, 589, 378, 503, 484, 72, 198, 256, 212, 856, 542, 232, 335, 195, 166, 853, 337, 42, 247, 838, 809, 447, 268, 334, 304, 432, 647, 140, 182, 457, 333, 322, 295, 64, 377, 823, 149, 607, 700, 765, 832, 232, 512, 486, 684, 587, 719, 857, 99, 411, 94, 457, 846, 132, 326, 313, 72, 707, 382, 358, 362, 460, 362, 81, 577, 262, 856, 717, 717, 192, 246, 493, 769, 851, 818, 114, 724, 253, 476, 394, 725, 737, 464, 596, 766, 767, 546, 417, 363, 645, 368, 847, 205, 398, 708, 144, 530, 712, 319, 84, 808, 649, 132, 283, 379, 58, 624, 268, 688, 642, 634, 22, 756, 602, 753, 726, 449, 267, 104, 120, 474, 352, 768, 780, 509, 720, 271, 277, 437, 750, 195, 722, 708, 619, 501, 722, 119, 343, 862, 785, 605, 608, 422, 397, 423, 479, 326, 375, 683, 297, 261, 576, 823, 626, 361, 780, 367, 410, 674, 232, 833, 791, 306, 623, 318, 83, 88, 80, 564, 288, 87, 409, 719, 662, 540, 819, 34, 243, 636, 261, 465, 670, 145, 647, 195, 337, 112, 689, 55, 859, 415, 349, 144, 176, 491, 418, 488, 216, 832, 658, 287, 602, 132, 694, 341, 728, 595, 676, 415, 25, 733, 740, 300, 816, 833, 405, 356, 436, 68, 752, 777, 41, 803, 296, 506, 536, 398, 693, 384, 796, 230, 494, 382, 687, 78, 765, 562, 274, 341, 367, 199, 424, 22, 443, 757, 825, 820, 457, 212, 592, 593, 383, 235, 534, 202, 233, 65, 325, 27, 560, 117, 702, 763, 503, 215, 281, 683, 642, 54, 820, 142, 378, 384, 271, 568, 481, 421, 268, 869, 443, 577, 429, 751, 417, 403, 71, 590, 62, 693, 507, 502, 99, 607, 711, 713, 304, 24, 283, 115, 704, 245, 768, 145, 582, 149, 591, 680, 62, 130, 171, 289, 89, 705, 651, 643, 214, 113, 401, 266, 32, 707, 432, 144, 103, 643, 639, 211, 507, 220, 336, 298, 323, 31, 824, 813, 138, 796, 33, 653, 260, 868, 821, 347, 710, 423, 799, 390, 765, 756, 120, 375, 144, 736, 780, 204, 363, 831, 125, 30, 405, 309, 627, 421, 575, 658, 82, 459, 188, 148, 793, 674, 408, 388, 312, 379, 503, 587, 838, 782, 161, 589, 671, 442, 766, 581, 753, 416, 811, 66, 249, 586, 347, 488, 141, 282, 667, 262, 413, 37, 559, 587, 296, 446, 50, 749, 144, 72, 535, 863, 47, 172, 675, 590, 595, 708, 325, 349, 853, 562, 554, 60, 791, 421, 98, 450, 638, 446, 276, 493, 122, 495}

Out[2434]=

```
{ {829, 2}, {358, 2}, {290, 1}, {489, 2}, {315, 1}, {126, 1}, {623, 2}, {189, 1}, {158, 1}, {855, 1}, {269, 2}, {108, 2},
  {311, 2}, {441, 2}, {174, 1}, {763, 2}, {450, 2}, {317, 1}, {419, 1}, {36, 2}, {231, 3}, {502, 3}, {453, 2},
  {454, 1}, {861, 2}, {513, 1}, {70, 1}, {260, 2}, ... 496 ..., {736, 1}, {204, 1}, {125, 1}, {30, 1}, {309, 1},
  {627, 1}, {82, 1}, {459, 1}, {188, 1}, {148, 1}, {161, 1}, {442, 1}, {416, 1}, {66, 1}, {249, 1}, {141, 1},
  {282, 1}, {667, 1}, {37, 1}, {559, 1}, {50, 1}, {535, 1}, {47, 1}, {675, 1}, {554, 1}, {98, 1}, {276, 1}, {495, 1} }
```

Full expression not available (original memory size: 9.3 kB)



Out[2436]=

337

Out[2437]=

3

15. В поле целых чисел определить сумму элементов приведенной системы вычетов по модулю mod.

In[2557]:=

```
mod = 30
list = Mod[Range[1, mod], mod]
listPriv = {}
sum = 0
Do[If[GCD[list[[i]], mod] == 1, AppendTo[listPriv, list[[i]]], {i, Length[list]}]
listPriv
EulerPhi[mod] == Length[listPriv]
Total[listPriv]
```

Out[2557]=

30

Out[2558]=

```
{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15,
 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 0}
```

Out[2559]=

{}

Out[2560]=

0

Out[2562]=

{1, 7, 11, 13, 17, 19, 23, 29}

Out[2563]=

True

Out[2564]=

120

16. В поле $GF[313]$ определить произведение обратного элемента по сложению числа $a = 241$ и обратного элемента по умножению для числа $b = 106$

```
In[2527]:=  
a = 241  
b = 106  
GF = 313  
reverseA = GF - a  
reverseB = PowerMod[b, -1, GF]  
result = Mod[reverseA * reverseB, GF]
```

```
Out[2527]=  
241
```

```
Out[2528]=  
106
```

```
Out[2529]=  
313
```

```
Out[2530]=  
72
```

```
Out[2531]=  
251
```

```
Out[2532]=  
231
```

17. Определите количество положительных целых чисел, меньших number, которые взаимно просты с number.

In[2565]:=

```

number = 5045
count = 0
Do[If[CoprimeQ[i, number], count = count + 1], {i, 1, number - 1}]
count

```

(*ИЛИ*)

```

number = 5045
count = 0
Do[If[GCD[i, number] == 1, count++], {i, 1, number}]
count

```

Out[2565]=

5045

Out[2566]=

0

Out[2568]=

4032

Out[2569]=

5045

Out[2570]=

0

Out[2572]=

4032

18. Создайте два списка чисел, объедините, отсортируйте, при значении генератора start осуществите перестановку элементов. Определить номера позиций максимального и минимального элемента в списке и найдите их (номеров позиций) произведение. RandomChoice

In[2462]:=

```
list1A = Range[1, 30]
list1B = Range[259, 1164]
listJoin = Join[list1A, list1B]
```

```
start = 667482789
SeedRandom[start]
```

```
listRes = RandomSample[listJoin]
posMin = Position[listRes, Min[listRes]]
posMax = Position[listRes, Max[listRes]]
```

```
res = posMin * posMax
```

Out[2462]=

```
{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15,
 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30}
```

Out[2463]=

```
{259, 260, 261, 262, 263, 264, 265, 266, 267, 268, 269, 270, 271, 272, 273, 274, 275, 276,
 277, 278, 279, 280, 281, 282, 283, 284, 285, 286, 287, 288, 289, 290, 291, 292, 293,
 294, 295, 296, 297, 298, 299, 300, 301, 302, 303, 304, 305, 306, 307, 308, 309, 310,
 311, 312, 313, 314, 315, 316, 317, 318, 319, 320, 321, 322, 323, 324, 325, 326, 327,
 328, 329, 330, 331, 332, 333, 334, 335, 336, 337, 338, 339, 340, 341, 342, 343, 344,
 345, 346, 347, 348, 349, 350, 351, 352, 353, 354, 355, 356, 357, 358, 359, 360, 361,
 362, 363, 364, 365, 366, 367, 368, 369, 370, 371, 372, 373, 374, 375, 376, 377,
 378, 379, 380, 381, 382, 383, 384, 385, 386, 387, 388, 389, 390, 391, 392, 393,
 394, 395, 396, 397, 398, 399, 400, 401, 402, 403, 404, 405, 406, 407, 408, 409,
 410, 411, 412, 413, 414, 415, 416, 417, 418, 419, 420, 421, 422, 423, 424, 425,
 426, 427, 428, 429, 430, 431, 432, 433, 434, 435, 436, 437, 438, 439, 440, 441,
 442, 443, 444, 445, 446, 447, 448, 449, 450, 451, 452, 453, 454, 455, 456, 457,
 458, 459, 460, 461, 462, 463, 464, 465, 466, 467, 468, 469, 470, 471, 472, 473, 474,
 475, 476, 477, 478, 479, 480, 481, 482, 483, 484, 485, 486, 487, 488, 489, 490,
 491, 492, 493, 494, 495, 496, 497, 498, 499, 500, 501, 502, 503, 504, 505, 506,
 507, 508, 509, 510, 511, 512, 513, 514, 515, 516, 517, 518, 519, 520, 521, 522, 523,
 524, 525, 526, 527, 528, 529, 530, 531, 532, 533, 534, 535, 536, 537, 538, 539,
 540, 541, 542, 543, 544, 545, 546, 547, 548, 549, 550, 551, 552, 553, 554, 555,
 556, 557, 558, 559, 560, 561, 562, 563, 564, 565, 566, 567, 568, 569, 570, 571, 572,
 573, 574, 575, 576, 577, 578, 579, 580, 581, 582, 583, 584, 585, 586, 587, 588,
 589, 590, 591, 592, 593, 594, 595, 596, 597, 598, 599, 600, 601, 602, 603, 604,
 605, 606, 607, 608, 609, 610, 611, 612, 613, 614, 615, 616, 617, 618, 619, 620, 621,
 622, 623, 624, 625, 626, 627, 628, 629, 630, 631, 632, 633, 634, 635, 636, 637,
 638, 639, 640, 641, 642, 643, 644, 645, 646, 647, 648, 649, 650, 651, 652, 653,
```

654, 655, 656, 657, 658, 659, 660, 661, 662, 663, 664, 665, 666, 667, 668, 669,
 670, 671, 672, 673, 674, 675, 676, 677, 678, 679, 680, 681, 682, 683, 684, 685, 686,
 687, 688, 689, 690, 691, 692, 693, 694, 695, 696, 697, 698, 699, 700, 701, 702,
 703, 704, 705, 706, 707, 708, 709, 710, 711, 712, 713, 714, 715, 716, 717, 718,
 719, 720, 721, 722, 723, 724, 725, 726, 727, 728, 729, 730, 731, 732, 733, 734,
 735, 736, 737, 738, 739, 740, 741, 742, 743, 744, 745, 746, 747, 748, 749, 750, 751,
 752, 753, 754, 755, 756, 757, 758, 759, 760, 761, 762, 763, 764, 765, 766, 767,
 768, 769, 770, 771, 772, 773, 774, 775, 776, 777, 778, 779, 780, 781, 782, 783,
 784, 785, 786, 787, 788, 789, 790, 791, 792, 793, 794, 795, 796, 797, 798, 799,
 800, 801, 802, 803, 804, 805, 806, 807, 808, 809, 810, 811, 812, 813, 814, 815,
 816, 817, 818, 819, 820, 821, 822, 823, 824, 825, 826, 827, 828, 829, 830, 831,
 832, 833, 834, 835, 836, 837, 838, 839, 840, 841, 842, 843, 844, 845, 846, 847,
 848, 849, 850, 851, 852, 853, 854, 855, 856, 857, 858, 859, 860, 861, 862, 863,
 864, 865, 866, 867, 868, 869, 870, 871, 872, 873, 874, 875, 876, 877, 878, 879,
 880, 881, 882, 883, 884, 885, 886, 887, 888, 889, 890, 891, 892, 893, 894, 895,
 896, 897, 898, 899, 900, 901, 902, 903, 904, 905, 906, 907, 908, 909, 910, 911,
 912, 913, 914, 915, 916, 917, 918, 919, 920, 921, 922, 923, 924, 925, 926, 927,
 928, 929, 930, 931, 932, 933, 934, 935, 936, 937, 938, 939, 940, 941, 942, 943,
 944, 945, 946, 947, 948, 949, 950, 951, 952, 953, 954, 955, 956, 957, 958, 959,
 960, 961, 962, 963, 964, 965, 966, 967, 968, 969, 970, 971, 972, 973, 974, 975,
 976, 977, 978, 979, 980, 981, 982, 983, 984, 985, 986, 987, 988, 989, 990, 991,
 992, 993, 994, 995, 996, 997, 998, 999, 1000, 1001, 1002, 1003, 1004, 1005, 1006,
 1007, 1008, 1009, 1010, 1011, 1012, 1013, 1014, 1015, 1016, 1017, 1018, 1019, 1020,
 1021, 1022, 1023, 1024, 1025, 1026, 1027, 1028, 1029, 1030, 1031, 1032, 1033,
 1034, 1035, 1036, 1037, 1038, 1039, 1040, 1041, 1042, 1043, 1044, 1045, 1046,
 1047, 1048, 1049, 1050, 1051, 1052, 1053, 1054, 1055, 1056, 1057, 1058, 1059,
 1060, 1061, 1062, 1063, 1064, 1065, 1066, 1067, 1068, 1069, 1070, 1071, 1072,
 1073, 1074, 1075, 1076, 1077, 1078, 1079, 1080, 1081, 1082, 1083, 1084, 1085,
 1086, 1087, 1088, 1089, 1090, 1091, 1092, 1093, 1094, 1095, 1096, 1097, 1098,
 1099, 1100, 1101, 1102, 1103, 1104, 1105, 1106, 1107, 1108, 1109, 1110, 1111,
 1112, 1113, 1114, 1115, 1116, 1117, 1118, 1119, 1120, 1121, 1122, 1123, 1124,
 1125, 1126, 1127, 1128, 1129, 1130, 1131, 1132, 1133, 1134, 1135, 1136, 1137,
 1138, 1139, 1140, 1141, 1142, 1143, 1144, 1145, 1146, 1147, 1148, 1149, 1150,
 1151, 1152, 1153, 1154, 1155, 1156, 1157, 1158, 1159, 1160, 1161, 1162, 1163, 1164}

Out[2464]=

{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25,
 26, 27, 28, 29, 30, 259, 260, 261, 262, 263, 264, 265, 266, 267, 268, 269, 270, 271,
 272, 273, 274, 275, 276, 277, 278, 279, 280, 281, 282, 283, 284, 285, 286, 287, 288,
 289, 290, 291, 292, 293, 294, 295, 296, 297, 298, 299, 300, 301, 302, 303, 304, 305,
 306, 307, 308, 309, 310, 311, 312, 313, 314, 315, 316, 317, 318, 319, 320, 321, 322,
 323, 324, 325, 326, 327, 328, 329, 330, 331, 332, 333, 334, 335, 336, 337, 338,
 339, 340, 341, 342, 343, 344, 345, 346, 347, 348, 349, 350, 351, 352, 353, 354,
 355, 356, 357, 358, 359, 360, 361, 362, 363, 364, 365, 366, 367, 368, 369, 370,

371, 372, 373, 374, 375, 376, 377, 378, 379, 380, 381, 382, 383, 384, 385, 386,
387, 388, 389, 390, 391, 392, 393, 394, 395, 396, 397, 398, 399, 400, 401, 402,
403, 404, 405, 406, 407, 408, 409, 410, 411, 412, 413, 414, 415, 416, 417, 418,
419, 420, 421, 422, 423, 424, 425, 426, 427, 428, 429, 430, 431, 432, 433, 434,
435, 436, 437, 438, 439, 440, 441, 442, 443, 444, 445, 446, 447, 448, 449, 450,
451, 452, 453, 454, 455, 456, 457, 458, 459, 460, 461, 462, 463, 464, 465, 466,
467, 468, 469, 470, 471, 472, 473, 474, 475, 476, 477, 478, 479, 480, 481, 482,
483, 484, 485, 486, 487, 488, 489, 490, 491, 492, 493, 494, 495, 496, 497, 498,
499, 500, 501, 502, 503, 504, 505, 506, 507, 508, 509, 510, 511, 512, 513, 514,
515, 516, 517, 518, 519, 520, 521, 522, 523, 524, 525, 526, 527, 528, 529, 530,
531, 532, 533, 534, 535, 536, 537, 538, 539, 540, 541, 542, 543, 544, 545, 546,
547, 548, 549, 550, 551, 552, 553, 554, 555, 556, 557, 558, 559, 560, 561, 562,
563, 564, 565, 566, 567, 568, 569, 570, 571, 572, 573, 574, 575, 576, 577, 578,
579, 580, 581, 582, 583, 584, 585, 586, 587, 588, 589, 590, 591, 592, 593, 594,
595, 596, 597, 598, 599, 600, 601, 602, 603, 604, 605, 606, 607, 608, 609, 610,
611, 612, 613, 614, 615, 616, 617, 618, 619, 620, 621, 622, 623, 624, 625, 626,
627, 628, 629, 630, 631, 632, 633, 634, 635, 636, 637, 638, 639, 640, 641, 642,
643, 644, 645, 646, 647, 648, 649, 650, 651, 652, 653, 654, 655, 656, 657, 658,
659, 660, 661, 662, 663, 664, 665, 666, 667, 668, 669, 670, 671, 672, 673, 674,
675, 676, 677, 678, 679, 680, 681, 682, 683, 684, 685, 686, 687, 688, 689, 690,
691, 692, 693, 694, 695, 696, 697, 698, 699, 700, 701, 702, 703, 704, 705, 706,
707, 708, 709, 710, 711, 712, 713, 714, 715, 716, 717, 718, 719, 720, 721, 722,
723, 724, 725, 726, 727, 728, 729, 730, 731, 732, 733, 734, 735, 736, 737, 738,
739, 740, 741, 742, 743, 744, 745, 746, 747, 748, 749, 750, 751, 752, 753, 754,
755, 756, 757, 758, 759, 760, 761, 762, 763, 764, 765, 766, 767, 768, 769, 770,
771, 772, 773, 774, 775, 776, 777, 778, 779, 780, 781, 782, 783, 784, 785, 786,
787, 788, 789, 790, 791, 792, 793, 794, 795, 796, 797, 798, 799, 800, 801, 802,
803, 804, 805, 806, 807, 808, 809, 810, 811, 812, 813, 814, 815, 816, 817, 818,
819, 820, 821, 822, 823, 824, 825, 826, 827, 828, 829, 830, 831, 832, 833, 834,
835, 836, 837, 838, 839, 840, 841, 842, 843, 844, 845, 846, 847, 848, 849, 850,
851, 852, 853, 854, 855, 856, 857, 858, 859, 860, 861, 862, 863, 864, 865, 866,
867, 868, 869, 870, 871, 872, 873, 874, 875, 876, 877, 878, 879, 880, 881, 882,
883, 884, 885, 886, 887, 888, 889, 890, 891, 892, 893, 894, 895, 896, 897, 898,
899, 900, 901, 902, 903, 904, 905, 906, 907, 908, 909, 910, 911, 912, 913, 914,
915, 916, 917, 918, 919, 920, 921, 922, 923, 924, 925, 926, 927, 928, 929, 930,
931, 932, 933, 934, 935, 936, 937, 938, 939, 940, 941, 942, 943, 944, 945, 946,
947, 948, 949, 950, 951, 952, 953, 954, 955, 956, 957, 958, 959, 960, 961, 962,
963, 964, 965, 966, 967, 968, 969, 970, 971, 972, 973, 974, 975, 976, 977, 978,
979, 980, 981, 982, 983, 984, 985, 986, 987, 988, 989, 990, 991, 992, 993, 994,
995, 996, 997, 998, 999, 1000, 1001, 1002, 1003, 1004, 1005, 1006, 1007, 1008,
1009, 1010, 1011, 1012, 1013, 1014, 1015, 1016, 1017, 1018, 1019, 1020, 1021,
1022, 1023, 1024, 1025, 1026, 1027, 1028, 1029, 1030, 1031, 1032, 1033, 1034,
1035, 1036, 1037, 1038, 1039, 1040, 1041, 1042, 1043, 1044, 1045, 1046, 1047,

```
1048, 1049, 1050, 1051, 1052, 1053, 1054, 1055, 1056, 1057, 1058, 1059, 1060,
1061, 1062, 1063, 1064, 1065, 1066, 1067, 1068, 1069, 1070, 1071, 1072, 1073,
1074, 1075, 1076, 1077, 1078, 1079, 1080, 1081, 1082, 1083, 1084, 1085, 1086,
1087, 1088, 1089, 1090, 1091, 1092, 1093, 1094, 1095, 1096, 1097, 1098, 1099,
1100, 1101, 1102, 1103, 1104, 1105, 1106, 1107, 1108, 1109, 1110, 1111, 1112,
1113, 1114, 1115, 1116, 1117, 1118, 1119, 1120, 1121, 1122, 1123, 1124, 1125,
1126, 1127, 1128, 1129, 1130, 1131, 1132, 1133, 1134, 1135, 1136, 1137, 1138,
1139, 1140, 1141, 1142, 1143, 1144, 1145, 1146, 1147, 1148, 1149, 1150, 1151,
1152, 1153, 1154, 1155, 1156, 1157, 1158, 1159, 1160, 1161, 1162, 1163, 1164}
```

Out[2465]=

```
667 482 789
```

Out[2466]=

```
RandomGeneratorState[ Method: ExtendedCA
                      State hash: 7223303020361197338 ]
```

Out[2467]=

```
{904, 284, 769, 753, 1028, 615, 948, 346, 1067, 451, 1134, 20, 849, 368, 354, 641, 418,
1109, 767, 679, 800, 412, 489, 1023, 398, 589, 1123, 842, 293, 1003, 803, 1145,
1151, 1136, 1083, 911, 622, 693, 281, 720, 1164, 497, 414, 415, 1129, 404, 401,
943, 637, 986, 511, 903, 316, 522, 488, 977, 885, 291, 465, 954, 588, 1068, 860,
287, 927, 878, 611, 413, 543, 996, 818, 568, 976, 923, 780, 799, 587, 905, 971,
1153, 764, 288, 285, 793, 337, 286, 508, 348, 603, 1030, 399, 691, 1016, 551, 577,
676, 804, 377, 480, 1085, 952, 1120, 953, 13, 397, 719, 673, 459, 941, 516, 499,
962, 680, 725, 606, 351, 1008, 960, 844, 260, 595, 639, 537, 478, 771, 888, 763,
870, 283, 578, 447, 978, 382, 778, 1069, 1156, 979, 379, 796, 585, 279, 1148, 1053,
648, 272, 289, 438, 610, 552, 635, 988, 916, 23, 1024, 949, 910, 1121, 340, 636,
907, 886, 14, 972, 319, 827, 746, 575, 524, 784, 408, 853, 496, 1057, 789, 782,
765, 934, 425, 1051, 813, 697, 608, 431, 290, 485, 384, 305, 601, 712, 968, 301,
625, 773, 718, 825, 776, 592, 312, 630, 494, 531, 576, 1143, 1161, 810, 616, 657,
707, 650, 433, 268, 863, 838, 1034, 812, 15, 985, 901, 1118, 742, 1157, 565, 994,
843, 741, 947, 982, 783, 569, 631, 779, 1125, 821, 722, 710, 730, 826, 1074, 262,
8, 446, 873, 1007, 598, 555, 389, 656, 411, 734, 599, 335, 774, 762, 893, 505, 364,
652, 304, 344, 1054, 756, 775, 513, 859, 274, 876, 1039, 961, 1066, 476, 1041, 739,
699, 899, 296, 824, 857, 538, 549, 609, 959, 896, 388, 990, 352, 1139, 701, 393,
1031, 666, 875, 755, 788, 989, 380, 1077, 571, 580, 1147, 302, 1104, 423, 579,
448, 332, 1040, 700, 750, 566, 805, 880, 674, 429, 612, 1122, 1115, 997, 938, 692,
331, 407, 664, 872, 391, 834, 434, 419, 24, 628, 541, 1107, 728, 403, 1094, 1010,
792, 963, 12, 559, 492, 416, 583, 501, 445, 946, 966, 466, 567, 683, 526, 495, 682,
584, 299, 1119, 665, 1000, 811, 326, 422, 928, 846, 392, 1060, 867, 591, 713, 581,
563, 1110, 942, 1154, 984, 534, 590, 1088, 366, 909, 751, 546, 267, 5, 359, 529,
704, 732, 1020, 17, 506, 772, 1100, 442, 306, 695, 951, 891, 527, 965, 806, 862,
604, 265, 957, 1138, 510, 689, 717, 594, 658, 517, 387, 6, 547, 390, 372, 504, 560,
1043, 9, 375, 832, 613, 329, 586, 1065, 845, 983, 678, 378, 308, 1080, 263, 654,
```

539, 409, 276, 2, 597, 837, 698, 1071, 900, 1097, 868, 929, 752, 545, 498, 1149, 519, 313, 1005, 309, 851, 914, 350, 833, 26, 11, 349, 482, 869, 292, 854, 441, 528, 920, 830, 28, 605, 21, 363, 696, 1105, 879, 1013, 259, 918, 381, 1063, 791, 318, 1137, 1064, 1061, 1035, 1076, 661, 1036, 1133, 421, 514, 864, 426, 607, 829, 515, 633, 822, 735, 694, 906, 858, 912, 303, 945, 618, 995, 685, 1026, 1042, 464, 400, 1058, 518, 898, 395, 530, 362, 1018, 617, 852, 22, 437, 973, 823, 861, 523, 282, 992, 1163, 1025, 394, 688, 1141, 264, 721, 370, 1050, 1108, 500, 336, 766, 1160, 1086, 839, 975, 278, 1112, 357, 10, 310, 670, 553, 295, 930, 980, 402, 908, 629, 1073, 314, 1082, 356, 558, 432, 638, 570, 1091, 562, 386, 820, 467, 334, 338, 743, 1075, 471, 874, 760, 1101, 417, 850, 847, 1001, 1055, 706, 1002, 420, 270, 29, 1059, 321, 325, 819, 436, 1106, 1144, 536, 1095, 428, 809, 690, 675, 1038, 535, 311, 1159, 647, 300, 1012, 371, 455, 1015, 1004, 1111, 458, 1084, 1062, 1116, 733, 491, 1096, 915, 865, 475, 269, 808, 667, 1033, 1029, 507, 1014, 512, 877, 1056, 320, 643, 481, 621, 1052, 919, 450, 396, 1046, 921, 723, 958, 1079, 640, 452, 509, 970, 646, 430, 1009, 16, 709, 708, 748, 931, 740, 757, 887, 1037, 883, 280, 672, 1152, 1087, 347, 1017, 889, 933, 770, 925, 564, 840, 479, 936, 1158, 663, 277, 457, 1047, 660, 1049, 991, 1114, 644, 275, 703, 855, 361, 856, 544, 729, 614, 502, 687, 582, 477, 327, 557, 503, 297, 627, 271, 902, 1098, 1155, 1021, 1093, 754, 490, 358, 624, 341, 998, 881, 472, 554, 324, 1078, 897, 427, 440, 744, 999, 736, 747, 669, 884, 1081, 1048, 761, 967, 376, 330, 406, 1127, 626, 866, 1135, 950, 469, 30, 790, 1132, 715, 540, 343, 649, 828, 1044, 369, 435, 345, 493, 1128, 939, 836, 548, 273, 1113, 651, 561, 339, 439, 1162, 454, 758, 955, 645, 882, 1150, 463, 342, 731, 484, 317, 550, 890, 1092, 1124, 461, 355, 405, 533, 1131, 383, 993, 684, 716, 1099, 787, 1045, 777, 19, 738, 600, 835, 18, 964, 1090, 974, 759, 926, 935, 807, 802, 328, 294, 987, 814, 532, 486, 727, 333, 981, 1102, 892, 969, 473, 749, 542, 702, 1, 266, 1070, 815, 323, 786, 841, 917, 470, 449, 424, 801, 460, 794, 487, 1072, 634, 298, 798, 596, 1142, 711, 620, 1032, 468, 831, 453, 1022, 1126, 474, 261, 681, 3, 724, 671, 483, 642, 768, 668, 659, 525, 619, 365, 1103, 795, 4, 1130, 1027, 940, 632, 374, 574, 871, 1011, 894, 385, 7, 922, 307, 817, 322, 410, 797, 924, 655, 1140, 623, 944, 816, 745, 1006, 653, 373, 602, 956, 444, 1019, 520, 662, 462, 1089, 556, 705, 726, 572, 714, 443, 25, 367, 456, 27, 913, 573, 895, 781, 677, 785, 937, 737, 848, 315, 1117, 521, 1146, 593, 360, 932, 353, 686}

Out[2468]=

{{828}}

Out[2469]=

{{41}}

Out[2470]=

{{33 948}}

19. Простые случайные числа (count штук),

лежащие в диапазоне от min до max, найти сумму последних n

In[2438]:=

min = 6678**max = 73 458****count = 74****n = 5****sum = 0****start = 111 432 590****SeedRandom[start]****listRandom = RandomPrime[{min, max}, count]****Do[sum = sum + listRandom[[i]], {i, Length[listRandom] - n + 1, Length[listRandom]}]****sum**

Out[2438]=

6678

Out[2439]=

73 458

Out[2440]=

74

Out[2441]=

5

Out[2442]=

0

Out[2443]=

111 432 590

Out[2444]=

RandomGeneratorState[

Method: ExtendedCA
State hash: -7 880 127 310 187 003 425

]

Out[2445]=

{7949, 13 259, 51 131, 65 719, 22 277, 19 979, 44 537, 36 313, 49 253, 7507, 32 633, 53 051, 41 149, 51 817, 57 427, 65 881, 40 903, 6703, 18 553, 71 821, 19 301, 51 199, 20 117, 7723, 8431, 57 809, 7211, 48 589, 39 133, 7789, 36 559, 38 639, 12 799, 51 287, 65 699, 54 833, 8669, 44 207, 65 393, 16 339, 9341, 17 921, 44 579, 11 261, 66 071, 51 157, 28 759, 19 289, 34 613, 52 267, 27 397, 33 721, 54 331, 58 099, 16 829, 28 349, 10 531, 52 391, 71 399, 30 941, 33 721, 62 627, 18 859, 39 659, 9421, 30 467, 34 369, 23 899, 70 379, 8089, 32 089, 53 887, 10 567, 36 217}

Out[2447]=

140 849