

1. Сформировать список, состоящий из строчных символов русского алфавита. Определите среднее значение кодов этих символов в кодировке UNICODE. Ответ округлить до меньшего целого.

```
alphChar = CharacterRange["a", "я"]
charCode = ToCharacterCode[StringJoin[alphChar]]
charSum = Sum[charCode[[i]], {i, 1, Length[charCode]}]
charSrednee = N[charSum / Length[charCode]]
```

2. Найти два простых числа, ближайших к 7920, меньших 7920. Определить произведение этих чисел по модулю 134

```
In[22]:= chislo = 7920
prost1 = NextPrime[chislo, -1]
prost2 = NextPrime[chislo, -2]
proizvedenie = Mod[prost1*prost2, 134]
```

3. Переведите число 4D73F3, представленное в шестнадцатеричной форме, в десятичную и определите число разрядов.

```
In[40]:= chislo3 = BaseForm[16^4D73F3, 10]
IntegerLength[chislo3]
```

4. Создайте два списка чисел: Первый в диапазоне от 1 до 54, второй - от 200 до 1365, затем объедините оба списка. В отсортированном объединенном списке при начальном значении генератора псевдослучайных чисел, равном 418463498, было выбрано два случайных значения. Определите номера позиций найденных элементов в отсортированном списке и найдите произведение номеров позиции.

```
In[166]:= list41 = Range[1, 54]
list42 = Range[200, 1365]
list4 = Union[list41, list42]
list4 = Sort[list4]
SeedRandom[418 463 498]
random1 = RandomChoice[list4, 1]
random2 = RandomChoice[list4, 1]
listNum1 = Position[list4, Min[random1]]
listNum2 = Position[list4, Min[random2]]
proizv4 = listNum1*listNum2
```

5. Импортировать файл textrpl.txt из папки ФО К331. Исключить из текста все символы, кроме строчных и прописных букв русского алфавита. Преобразовать полученный список в матрицу размерностью 6x157. В столбцах с 53 по 102 провести замену строчных букв на прописные. Определить сумму кодов элементов матрицы.

text = "И как этот же самый лес хорош поздней осенью , когда прилетают вальдшнепы ! Они не держатся в самой глуши : их надобно искать вдоль опушки . Ветра нет, и нет ни солнца , ни света, ни тени, ни движения , ни шума; в мягком воздухе разлит осенний запах, подобный запаху вина; тонкий туман стоит вдали над желтыми полями . Сквозь обнаженные , бурые сучья деревьев мирно белеет неподвижное небо; кое-где на липах висят последние золотые листья . Сырая земля упруга под ногами ; высокие сухие былинки не шевелятся ; длинные нити блестят на побледневшей траве. Спокойно дышит грудь, а на душу находит странная тревога . Идешь вдоль опушки , глядишь за собакой , а между тем любимые образы , любимые лица, мертвые и живые , приходят на память , давным-давно заснувшие впечатления неожиданно просыпаются ; воображение реет и носится , как птица, и всё так ясно движется и стоит перед глазами . Сердце то вдруг задрожит и забьется , страстно бросится вперед , то безвозвратно потонет в воспоминаниях . Вся жизнь разворачивается легко и быстро , как свиток ; всем своим прошедшим , всеми чувствами , силами , всею своею душою владеет человек . И ничего кругом ему не мешает – ни солнца нет, ни ветра, ни шуму"

```
alpha5 = CharacterRange["A", "я"]
textForMatr = StringCases[text, alpha5]
matr = Partition[textForMatr, 157]
matr1 = matr
For[i = 1, i ≤ 6, i++,
For[j = 53, j ≤ 102, j++, matr1[[i, j]] = ToUpperCase[matr1[[i, j]]]]
]
matr1 // TableForm
textmatr = StringJoin[matr1]
kodmatr = ToCharacterCode[textmatr]
sum = 0
For[i = 1, i ≤ 942, i++, sum = sum + kodmatr[[i]]
Sum
Total[kodmatr]
```

6. Известно, что начальное состояние генератора случайных чисел установлено как 24059307. Было сформировано 32 случайных **простых** чисел в диапазоне 5643 - 39501. Определите сумму последних 5-ти (значит в цикле вычитаем 4) сформированных чисел.

```
In[21]:= SeedRandom[24 059 307]
list6 = RandomPrime[{5643, 39 501}, 32]
sum6 = 0
For[i6 = Length[list6] - 4, i6 ≤ Length[list6], i6++, sum6 = sum6 + list6[[i6]]
sum6
```

7. Сформировать множество целых чисел из интервала [56,119] и множество простых чисел из интервала [82,127]. Найти произведение элементов, принадлежащих обоим множествам. Если множества не пересекаются, ввести ответ: 0

```
In[70]:= list71 = Range[56, 119]
min72 = 82
max72 = 127
If[PrimeQ[min72], t1 = PrimePi[min72], t1 = PrimePi[min72] + 1]
t2 = PrimePi[max72]
list72 = Table[Prime[x], {x, t1, t2}]
listM = Intersection[list71, list72]
lenList = Length[listM]
mult = 1
For[j = 1, j ≤ lenList, j++, mult *= listM[[j]]]
mult
```

8. Определить количество простых чисел, меньших или равных 5373

```
In[81]:= r81 = 0
r82 = 5373
listR = Table[s, {s, r81, r82}]
If[PrimeQ[r81], beforeR81 = PrimePi[r81] - 1, beforeR81 = PrimePi[r81]]
beforeR82 = PrimePi[r82]
cont = beforeR82 - beforeR81
```

9. Получить список из первых 1849 нечетных чисел. Сформировать из списка квадратную матрицу. Провести последовательно следующие операции: поменять местами строки 5 и 42; циклически сдвинуть 14-ю строку на 63 позиций вправо. Поменять местами столбцы 18 и 40; циклически сдвинуть 19-й столбец на 61 позиций вниз. Рассчитать сумму элементов главной диагонали.

```
In[118]:= list9 = {}
For[i9 = 0, i9 ≤ 1849, i9++, If[Mod[i9, 2] == 1, AppendTo[list9, i9]]]
Length[list9]
i9 = 0
While[Length[list9] ≤ 1848,
If[Mod[i9, 2] == 1, AppendTo[list9, i9]] ×
i9++]
```

```
Out[118]=
{}

```

```
Out[120]=
925

```

```
Out[121]=
0

```

```
In[123]:= Length[list9]
```

```
Out[123]=
```

```
1849
```

9. Получить список из первых 1849 нечетных чисел. Сформировать из списка квадратную матрицу. Провести последовательно следующие операции: поменять местами строки 5 и 42; циклически сдвинуть 14-ю строку на 63 позиций вправо. Поменять местами столбцы 18 и 40; циклически сдвинуть 19-й столбец на 61 позиций вниз. Рассчитать сумму элементов главной диагонали.

```
In[1]:= list9 = {}  
lengthList9 = 1849  
i9 = 0  
While [Length[list9] ≤ lengthList9 - 1,  
If [Mod[i9, 2] == 1, AppendTo[list9, i9]] ×  
i9++]  
list9 = Partition[list9, Sqrt[lengthList9]]  
Dimensions[list9]  
  
Поменять местами строки 5 (swapistr91) и 42 (swapistr92):
```

```
In[7]:= swapistr91 = 5  
swapistr92 = 42  
row1 = list9[[swapistr91]]  
list9[[swapistr91]] = list9[[swapistr92]]  
list9[[swapistr92]] = row1  
  
Циклически сдвинуть 14-ю строку (sdvigstr91) на 63 позиций вправо (goRight9):
```

```
In[12]:= sdvigstr91 = 14  
goRight9 = 63  
sdvigRow91 = list9[[14]]  
sdvigRow91 = RotateRight[sdvigRow91, goRight9]  
list9[[14]] = sdvigRow91
```

Поменять местами столбцы 18 (swapStolb91) и 40 (swapStolb92). Для этого транспонируем копию матрицы, обработаем и поставим обратно:

```
In[17]:= swapStolb91 = 18  
swapStolb92 = 40  
list9Copy = list9  
list9Copy = Transpose[list9Copy]  
swapRow91 = list9Copy[[swapStolb91]]  
list9Copy[[swapStolb91]] = list9Copy[[swapStolb92]]  
list9Copy[[swapStolb92]] = swapRow91  
list9Copy = Transpose[list9Copy]  
list9 = list9Copy
```

Циклически сдвинуть 19-й столбец (sdvigStolb91) на 61 позиций вниз (goDown9):

```

In[26]:= sdvigStolb91 = 19
          goDown9 = 61
          list9Copy = list9
          list9Copy = Transpose[list9Copy]
          sdvigRow92 = list9Copy[[sdvigStolb91]]
          sdvigRow92 = RotateRight[sdvigRow92, goDown9]
          list9Copy[[sdvigStolb91]] = sdvigRow92
          list9Copy = Transpose[list9Copy]
          list9 = list9Copy

```

Рассчитать **сумму** элементов главной диагонали:

```

sum91 = 0
For[i9 = 1, i9 ≤ Sqrt[lengthList9], i9++, sum91 = sum91 + list9[[i9, i9]]]
sum91

```

10. Определить число простых чисел, содержащихся в интервале [281,6730].

```

In[50]:= chislo101 = 281
          chislo102 = 6730
          If[PrimeQ[chislo101], t1 = PrimePi[chislo101], t1 = PrimePi[chislo101] + 1]
          t2 = PrimePi[chislo102]
          list102 = Table[Prime[x], {x, t1, t2}]
          Length[list102]

```

Контрольная работа №2

1. При стартовом значении генератора случайных чисел равном 37 (rand1) сформировать последовательность, состоящую из 811 (count1) случайных целых чисел, лежащих в диапазоне [139, 839]([a1,b1]). Найти произведение элементов последовательности, принадлежащих подмножеству, содержащему **тройную** коллизию (colliz1). В поле для ответа ввести количество разрядов для двоичного представления полученного произведения.

```

In[56]:= rand1 = 37
count1 = 811
a1 = 139
b1 = 839
colliz1 = 3
SeedRandom[rand1]
list1 = RandomChoice[Range[a1, b1], count1]
mul1 = 1
For[i1 = a1, i1 ≤ b1, i1++, If[Count[list1, i1] == colliz1+1, mul1 = mul1 * i1]]
Length[IntegerDigits[mul1, 2]]

```

2. Определить количество n во множестве, если при 70 (count2) экспериментах извлечения, коллизия возникает с вероятностью 0.3 (chance2). Ответ округлить до ближайшего большего целого.

```

In[66]:= count2 = 70
chance2 = 0.3
Solve[1 - Exp[(- count2 * (count2 - 1)) / (2 * n)] == chance2, n]

```

3. В поле целых чисел определить сумму элементов приведенной системы вычетов по модулю 3841 (r).

```

In[4]:= r = 3841
VichF[r_] :=
Module[{PolnVich = {},
PrivVich = {}, i},
Print["Полная система вычетов :"];
PolnVich = Range[0, r - 1];
Print[PolnVich];
Print["Длина :"] * Print[Length[PolnVich]];
PrivVich = {};
For[i = 0,
i ≤ Length[PolnVich],
i++,
If[GCD[PolnVich[[i]], r] == 1,
AppendTo[PrivVich, PolnVich[[i]]]];
Print["Приведенная система вычетов :"];
Print[PrivVich];
Print["Длина :"]; Print[Length[PrivVich]];
Print["Длина (Phi):"]; Print[EulerPhi[r]];
Print["Ответ:"]; Print[Total[PrivVich]]
]
VichF[r]

```

4. Скачайте с сетевого диска (ftp-сервера) файл Text-081.txt, расположенный в папке Texts и определите энтропию сообщения, содержащегося в нем. Ответ представить в битах, с 7

знаками после запятой. Пример ввода 1.1111111
ДЕЛАТЬ ЭТО В ДЕСКТОПНОЙ ВЕРСИИ

```
text4 = ReadList["C:\\Users\\Admin\\OneDrive\\Desktop\\Text-081.txt", Byte]
Ntext4 = N[Entropy[2, text4], 8]
```

5. Установить генератор случайных чисел в начальное состояние с параметром, равным обратному элементу числа 1718 (chislo5) по модулю 157 (mod5). Получить список, состоящий из 100 (count5) случайных **строчных букв английского алфавита**. Инициализировать массив 10×10 с нулевыми начальными индексами элементами этого списка. Преобразовать элементы массива с индексами [7,3], [0,0], [0,7], [1,9], [4,8], [2,0], [7,3] в строку и ввести в поле ввода.

```
In[17]:= chislo5 = 1718
mod5 = 157
count5 = 100
randomInt = PowerMod[chislo5, -1, mod5]
SeedRandom[randomInt]
listAlph = CharacterRange["a", "z"]
randomStr5 = RandomChoice[listAlph, count5]
Array[array5, {10, 10}, 0]
masAlph = Partition[randomStr5, 10]
Do[Do[array5[i - 1, j - 1] = masAlph[[i, j]], {i, 10}], {j, 10}]
array5[7, 3]
array5[0, 0]
array5[0, 7]
array5[1, 9]
array5[4, 8]
array5[2, 0]
array5[7, 3]
```

6. В поле GF[163] (GF[pole6]) определить произведение обратного элемента по сложению для числа $a = 121$ и обратного элемента по умножению для числа $b = 101$.

```
In[39]:= pole6 = 163
a6 = 121
b6 = 101
na6 = pole6 - a6
nb6 = PowerMod[b6, -1, pole6]
otvet6 = Mod[na6 * nb6, pole6]
```

7. Определите обратный элемент числа 52632 (chislo7) в поле GF(609779) (GF[pole7]).


```
In[45]:= chislo7 = 52 632
pole7 = 609 779
otvet7 = PowerMod[chislo7, -1, pole7]
```

8. Определите количество положительных целых чисел, меньших 7637 (chislo8), которые взаимно просты с 7637.

```
In[50]:= chislo8 = 7637
For[i8 = 1; t8 = 0, i8 < chislo8, If[GCD[i8, chislo8] == 1, t8 = t8 + 1,]; i8++]
t8

Out[50]= 7637

Out[52]= 6540
```

Контрольная работа №3

1. Скачать с FTP - сервера и запустить файл break31.exe. Определить пароль любым доступным способом.

Алгоритм решения:

1. Скачать файл на рабочий стол своего реального компа
2. Запустить виртуальную машину.
3. Скопировать в виртуальную машину файл (с помощью флешки или через буфер обмена).
4. Открыть этот файл в программе WinHex.
5. Найти среди текстового описания содержимого этого файла место, где написано что-то типа "Password:"
6. Пароль состоит из 4 символов. Там могут содержаться латинские (они норм отображаются) и кириллица. Для латинских символов (они отображаются странными символами) нужно посмотреть hex описание и сопоставить номер этой буквы с таблицей ASCII.

2. Найти значение функции Эйлера для числа x , которое определяется из соотношения $a \cdot x + b = c \pmod{n}$, где $a = 35105$, $b = 34076$, $c = 13458$, $n = 12953$.

```
In[77]:= a2 = 35 105
b2 = 34 076
c2 = 13 458
n2 = 12 953
i2 = 1; While[Mod[Mod[a2, n2] * i2, n2] != Mod[c2 - b2, n2], i2++];
EulerPhi[i2]
```

3. Установить генератор случайных чисел в начальное состояние с параметром равным $2^{15} \pmod{71}$. Получить список из 10000 (count3) случайных простых чисел в диапазоне от 5000 (a3)

до 16000 (b3). Найти произведение двух простых чисел, которые встречаются в списке с максимальной (применять функцию Max[]) и минимальной (применять функцию Min[]) частотой. В случае наличия чисел с одинаковыми частотами выбирать первые в списке.

```
In[1]:= randomInt3 = PowerMod[2, 15, 71]
SeedRandom[randomInt3]
count3 = 10 000
a3 = 5000
b3 = 16 000
list3 = RandomPrime[{a3, b3}, count3]
listFREQ = Tally[list3]
listSORT = Sort[listFREQ[[All, 2]], Greater]

In[9]:= i3 = 1
While[listFREQ[[i3, 2]] != listSORT[[1]], i3++];
max3 = listFREQ[[i3, 1]]
u3 = 1
While[listFREQ[[u3, 2]] != listSORT[[Length[listSORT]]], u3++];
min3 = listFREQ[[u3, 1]]
otvet3 = min3 * max3
```

4. Определить энтропию сектора с номером 1660 виртуального флоппи-диска flptest.flp с точностью 5 знаков после запятой. Для округления результата применить функцию N[.]. Пример ввода 5.55555

Алгоритм решения:

1. Заходим в WinHex на виртуальной машине.
2. Выбираем флоппи-диск (Инструменты -> Открыть диск -> Floppy Disk 0). Предварительно нужно его установить в настройках самой машины (Player -> Removable devices -> Floppy -> Connect (или setting если надо указать путь к флоппи диску)).
3. Переходим к сектору (Позиция -> Go To Sector и вводим номер сектора из задания). Можно найти по смещению. Для этого номер сектора умножить на 512 и перейти к смещению
4. Выделяем текст в секторе (сектор выделяется линиями) от начала линии и до конца. Затем переходим в Правка -> Copy Block -> В новый файл. Сохраняем файл на рабочем столе виртуальной машины.
5. Открываем Converter с рабочего стола. Выбираем для преобразования только что сохранённый файл. Преобразуем в файл типа .dat и сохраняем.
6. Копируем файл на рабочий стол реального компьютера (через флешку или буфер обмена).
7. Заходим в математику (у меня десктопная) и пишем туда следующее:

```
myfile = ReadList["Путь к своему файлу", Number]
nByte = N[Entropy[2, myfile], 8]
```

5. Архив текстового файла archive-127.zip защищен паролем из 4-х 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 65.5, а полуразность кода символа и кода позиции символа в пароле равна 14.5. Исключить пробелы и подсчитать число символов в тексте.

Алгоритм решения:

1. Вычисляем систему уравнений:

$$(x+y)/2 = 65.5$$

$$(x-y)/2 = 14.5$$

где x - код символа, y - код позиции символа

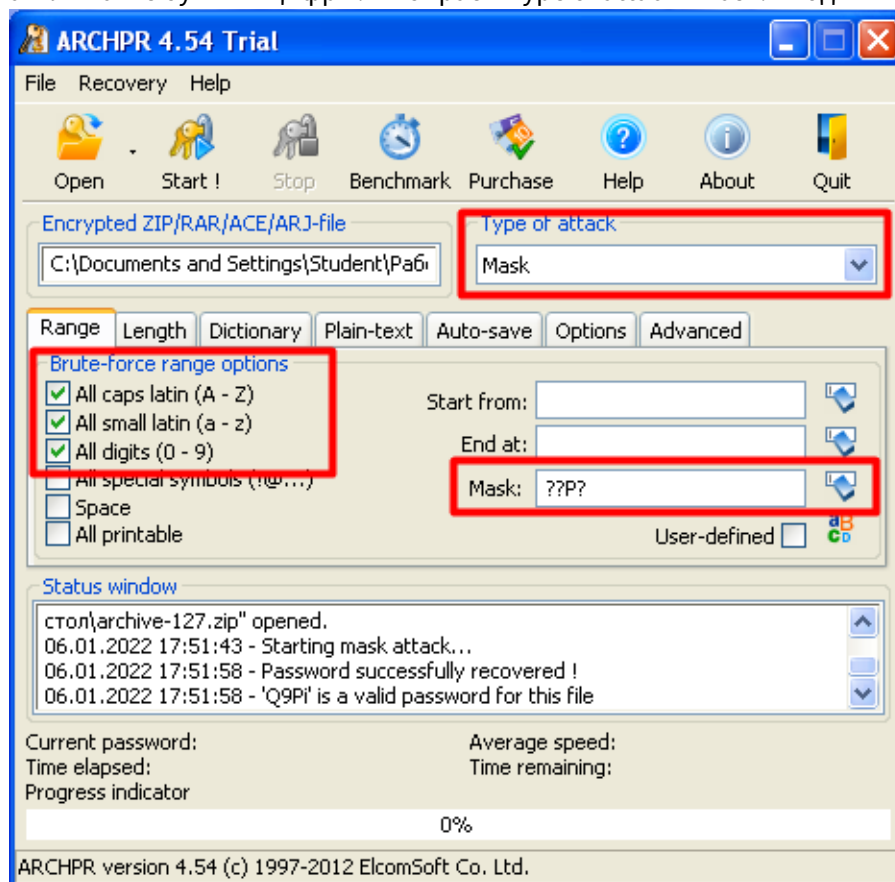
В моём случае $x = 80$, $y = 512$

2. Сопоставляем коды по таблице ASCII. Код 80 соответствует букве P, а код 51 соответствует цифре 3.

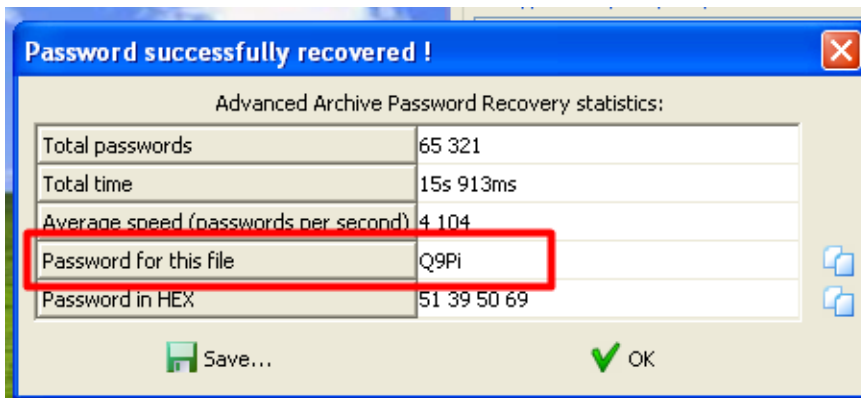
Получаем, что в пароле на третьей позиции стоит буква P. Получили маску ??P?

3. Открываем виртуальную машину. Перетаскиваем в неё архив (через флешку или буфер обмена).

4. Открываем Advanced Archive Password Recovery. Выбираем там заглавные и строчные английские буквы и цифры. Выбираем Type of attack - Mask. Вводим маску.



5. Открываем там архив и запускаем поиск пароля (start recovery). Получаем пароль:



6. Открываем архив и распаковываем файл.

7. Заходим в вольфрам и пишем (можно просто скопировать текст в переменную):

```
text1 = ReadString["C:\\Users\\Admin\\OneDrive\\Desktop\\Text-127.txt"]
s1 = StringReplace[text1, {" " → ""}]
StringLength[s1]
```

6. Определить ожидаемое время раскрытия пароля длиной 9 символов (S6) и содержащего следующие наборы: {цифры, прописные русские, строчные латинские}, если скорость перебора пароля (пароль в секунду) равна обратному элементу числа 2971 (a6) по модулю 547 (pole6). Ответ вводить как целое число суток.

```
In[26]:= list61 = CharacterRange["0", "9"]
list62 = CharacterRange["a", "z"]
list63 = CharacterRange["А", "Я"]
S6 = 9
a6 = 2971
pole6 = 547
R6 = PowerMod[a6, -1, pole6]
list6 = Union[list61, list62, list63]
A6 = Length[list6]
tS = IntegerPart[(1/2) * (A6 ^ S6 * 1 / R6) / 60 / 60 / 24]
```