

# Шаблоны кодов ЕГЭ

ЕГЭ 2025

# Задание 2

Миша заполнял таблицу истинности логической функции  $F = ((w \rightarrow y) \rightarrow x) \vee \neg z$ , но успел заполнить лишь фрагмент из трёх различных её строк, даже не указав, какому столбцу таблицы соответствует каждая из переменных w, x, y, z.

				F
		1		0
	0			0
	1	0	0	0

Определите, какому столбцу таблицы соответствует каждая из переменных w, x, y, z.

В ответе напишите буквы w, x, y, z в том порядке, в котором идут соответствующие им столбцы (сначала буква, соответствующая первому столбцу; затем буква, соответствующая второму столбцу, и т.д.). Буквы в ответе пишите подряд, никаких разделителей между буквами ставить не нужно.

```
print('x', 'y', 'w', 'z')
for x in range(2):
    for y in range(2):
        for w in range(2):
            for z in range(2):
                if (((w <= y) <= x) or (not(z))) == 0:
                    print(x, y, z, w)
```

x	y	w	z
0	0	1	0
0	1	1	0
0	1	1	1

Путём перестановки строк и столбцов  
получаем итоговый ответ

# Задание 5

На вход алгоритма подаётся натуральное число  $N$ . Алгоритм строит по нему новое число  $R$  следующим образом.

1. Строится двоичная запись числа  $N$ .

2. Далее эта запись обрабатывается по следующему правилу:

а) если число чётное, то к двоичной записи числа слева дописывается 10;

б) если число нечётное, то к двоичной записи числа слева дописывается 1 и справа дописывается 01.

Полученная таким образом запись является двоичной записью искомого числа  $R$ .

3. Результат переводится в десятичную систему и выводится на экран.

Например, для исходного числа  $4_{10} = 100_2$  результатом является число  $20_{10} = 10100_2$ , а для исходного числа  $5_{10} = 101_2$  это число  $53_{10} = 110101_2$ .

Укажите максимальное число  $R$ , которое может быть результатом работы данного алгоритма, при условии, что  $N$  не больше 12. В ответе запишите это число в десятичной системе счисления.

```
def r(n):  
    N = bin(n)[2:]  
    if n % 2 == 0:  
        N = '10' + N  
    else:  
        N = '1' + N + '01'  
    return int(N, 2)  
  
k = []  
for n in range(1, 13):  
    k.append(r(n))  
print(max(k))
```

# Задание 5

(Л. Шастин) На вход алгоритма подаётся натуральное число  $N$ . Алгоритм строит по нему новое число  $R$  следующим образом.

1. Строится семеричная запись числа  $N$ .

2. Далее эта запись обрабатывается по следующему правилу:

а) если количество двоек в этой записи чётно, то к ней дописываются 3 пятёрки.

б) иначе, если количество двоек в этой записи нечётно, то слева к этой записи дописывается 1 единица.

Полученная таким образом запись является семеричной записью искомого числа  $R$ .

3. Результат переводится в десятичную систему и выводится на экран.

Например, для исходного числа  $11 = 14_7$  результатом является число  $14555_7 = 4058$ , а для исходного числа  $14 = 20_7$  это число  $120_7 = 63$ .

Укажите максимальное число  $N$ , после обработки которого с помощью этого алгоритма получается число  $R$ , меньшее 3799.

```
def sm(n): #функция перевода в 7СС
    s = ''
    while n > 0:
        s = str(n % 7) + s
        n //= 7
    return s

1 usage
def r(n):
    N = sm(n)
    if N.count('2') % 2 == 0:
        N = N + '555'
    else:
        N = '1' + N
    return int(N, 7)
for n in range(1, 10001):
    if r(n) < 3799:
        print(n)
```

# Задание 8

Определите количество 12-ричных пятизначных чисел, в записи которых ровно одна цифра 7 и не более трёх цифр с числовым значением, превышающим 8.

```
from itertools import *
s = '0123456789ab' #алфавит 12СС
k = 0
for x in product(s, repeat = 5):
    x = ''.join(x)
    if x.count('7') == 1 and x[0] != '0' and \
        (x.count('9') + x.count('a') + x.count('b')) <= 3:
        k += 1
print(k)
```



# Задание 8

М. Ишимов

(М. Ишимов) Все пятибуквенные слова, составленные из букв Л, О, Г, А, Р, И, Ф, М, записаны в алфавитном порядке и пронумерованы.

Вот начало списка:

1. ААААА
2. ААААГ
3. ААААИ
4. ААААЛ
5. ААААМ
6. ААААО
7. ААААР

...

Определите в этом списке количество слов с чётными номерами, которые не начинаются на ЛМ и при этом содержат в своей записи не менее двух букв И.

```
from itertools import *
s = 'логарифм'
k = 0
l = 0
for x in product(sorted(s), repeat = 5): #делаем sorted, чтобы был алфавитный порядок
    x = ''.join(x)
    k += 1
    if x[:2] != 'лм' and x.count('и') >= 2:
        if k % 2 == 0:
            l += 1
print(l)
```

# Задание 9

Откройте файл электронной таблицы, содержащей в каждой строке шесть натуральных чисел. Определите количество строк таблицы, содержащих числа, для которых выполнены оба условия:

- в строке только одно число повторяется трижды, остальные числа различны;
- квадрат суммы всех повторяющихся чисел строки больше квадрата суммы всех её неповторяющихся чисел.

В ответе запишите только число.

```
f = open('09.txt')
k = 0
for i in f:
    a = [int(x) for x in i.split()]
    a1 = [x for x in a if a.count(x) == 1]
    a3 = [x for x in a if a.count(x) == 3]
    if len(a3) == 3 and len(a1) == 3:
        if sum(a3) ** 2 > sum(a1) ** 2:
            k += 1
print(k)
```

# Задание 9

(Д. Бахтиев) Откройте файл электронной таблицы, содержащей в каждой строке пять натуральных чисел. Определите количество строк таблицы, содержащих числа, для которых выполнены оба условия:

- удвоенная сумма двух наибольших чисел в строке больше утроенной суммы остальных чисел
- не менее двух чисел в строке оканчиваются цифрой 5

В ответе запишите только число.

```
f = open('09_1.txt')
k = 0
for i in f:
    a = [int(x) for x in i.split()]
    a.sort()
    k1 = 0
    for l in a:
        if l % 10 == 5:
            k1 += 1
    if (a[-1] + a[-2]) * 2 > 3 * (sum(a[:3])):
        if k1 >= 2:
            k += 1
print(k)
```



# Задание 12

Исполнитель Редактор получает на вход строку цифр и преобразовывает её. Редактор может выполнять две команды, в обеих командах  $v$  и  $w$  обозначают цепочки цифр.

А) заменить ( $v, w$ ).

Эта команда заменяет в строке первое слева вхождение цепочки  $v$  на цепочку  $w$ . Например, выполнение команды заменить (111, 27) преобразует строку 05111150 в строку 0527150. Если в строке нет вхождений цепочки  $v$ , то выполнение команды заменить ( $v, w$ ) не меняет эту строку.

Б) нашлось ( $v$ ).

Эта команда проверяет, встречается ли цепочка  $v$  в строке исполнителя Редактор. Если она встречается, то команда возвращает логическое значение «истина», в противном случае возвращает значение «ложь». Строка исполнителя при этом не изменяется.

Какая строка получится в результате применения приведённой ниже программы к строке, состоящей из 81 идущей подряд цифре 1? В ответе запишите полученную строку.

НАЧАЛО

ПОКА нашлось (11111) ИЛИ нашлось (888)

    ЕСЛИ нашлось (11111)

        ТО заменить (11111, 88)

        ИНАЧЕ заменить (888, 8)

    КОНЕЦ ЕСЛИ

КОНЕЦ ПОКА

КОНЕЦ

```
s = 81 * '1'
while '11111' in s or '888' in s:
    if '11111' in s:
        s = s.replace('11111', '88', 1)
    else:
        s = s.replace('888', '8', 1)
print(s)
```

# Задание 12

Исполнитель Редактор получает на вход строку цифр и преобразовывает её. Редактор может выполнять две команды, в обеих командах  $v$  и  $w$  обозначают цепочки цифр.

А) **заменить** ( $v, w$ ). Эта команда заменяет в строке первое слева вхождение цепочки  $v$  на цепочку  $w$ . Например, выполнение команды **заменить** (111, 27) преобразует строку 05111150 в строку 0527150.

Если в строке нет вхождений цепочки  $v$ , то выполнение команды **заменить** ( $v, w$ ) не меняет эту строку.

Б) **нашлось** ( $v$ ).

Эта команда проверяет, встречается ли цепочка  $v$  в строке исполнителя Редактор. Если она встречается, то команда возвращает логическое значение «истина», в противном случае возвращает значение «ложь». Строка исполнителя при этом не изменяется.

Цикл **ПОКА** условие последовательность команд **КОНЕЦ ПОКА** выполняется, пока условие истинно.

В конструкции **ЕСЛИ** условие **ТО** команда1 **ИНАЧЕ** команда2 **КОНЕЦ ЕСЛИ** выполняется команда1 (если условие истинно) или команда2 (если условие ложно).

**Дана программа для Редактора:**

НАЧАЛО

ПОКА нашлось (19) ИЛИ нашлось (49) ИЛИ нашлось (999)

    ЕСЛИ нашлось (19)

        ТО заменить (19, 9)

    КОНЕЦ ЕСЛИ

    ЕСЛИ нашлось (49)

        ТО заменить (49, 91)

    КОНЕЦ ЕСЛИ

    ЕСЛИ нашлось (999)

        ТО заменить (999, 4)

    КОНЕЦ ЕСЛИ

КОНЕЦ ПОКА

КОНЕЦ

На вход приведённой выше программе поступает строка, начинающаяся с цифры «1», а затем содержащая  $n$  цифр «9» ( $3 < n < 10\,000$ ).

Определите **наибольшее** возможное значение суммы числовых значений цифр в строке, которая может быть результатом выполнения программы.

```
l = []
for n in range(4, 10000):
    s = '1' + n * '9'
    while '19' in s or '49' in s or '999' in s:
        if '19' in s:
            s = s.replace('19', '9', 1)
        if '49' in s:
            s = s.replace('49', '91', 1)
        if '999' in s:
            s = s.replace('999', '4', 1)
    l.append(s.count('1') + s.count('4') * 4 + s.count('9') * 9)
print(max(l))
```

# Задание 13

В терминологии сетей TCP/IP маской сети называют двоичное число, которое показывает, какая часть IP-адреса узла сети относится к адресу сети, а какая – к адресу узла в этой сети. Адрес сети получается в результате применения поразрядной конъюнкции к заданному адресу узла и маске сети. Сеть задана IP-адресом 172.16.168.0 и маской сети 255.255.248.0.

Сколько в этой сети IP-адресов, для которых количество единиц в двоичной записи IP-адреса не кратно 5?

В ответе укажите только число

```
from ipaddress import *
k = 0
net = ip_network('172.16.168.0/255.255.248.0')
for ip in net:
    if bin(int(ip))[2:].count('1') % 5 != 0:
        k += 1
print(k)
```

# Задание 14

Операнды арифметического выражения записаны в системе счисления с основанием 19.

$$98897x21_{19} + 2x923_{19}$$

В записи чисел переменной  $x$  обозначена неизвестная цифра из алфавита 19-ричной системы счисления. Определите наибольшее значение  $x$ , при котором значение данного арифметического выражения кратно 18.

Для найденного  $x$  вычислите частное от деления значения арифметического выражения на 18 и укажите его в ответе в десятичной системе счисления.

Основание системы счисления указывать не нужно.

```
for x in '0123456789abcdefghi':  
    s = int(f'98897{x}21', 19) + int(f'2{x}923', 19)  
    if s % 18 == 0:  
        print(s // 18)
```

Значение арифметического выражения  $3 \cdot 3125^8 + 2 \cdot 625^7 - 4 \cdot 625^6 + 3 \cdot 125^5 - 2 \cdot 25^4 - 2025$  записали в системе счисления с основанием 25. Сколько значащих нулей содержится в этой записи?

```
k = 3 * 3125 ** 8 + 2 * 325 ** 7 - 4 * 625 ** 6 + 3 * 125 ** 5 + 2 * 25 ** 4 - 2025  
h = 0  
while k > 0:  
    if k % 25 == 0:  
        h += 1  
    k //= 25  
print(h)
```

# Задание 14

Значение арифметического выражения  $7^{110} + 7^{100} - x$ , где  $x$  – целое положительное число, не превышающее 2030, записали в 7-ричной системе счисления. Определите наибольшее значение  $x$ , при котором в 7-ричной записи числа, являющегося значением данного арифметического выражения, содержится ровно 71 нуль.

В ответе запишите число в десятичной системе счисления.

```
def sm(n):  
    s = ''  
    while n > 0:  
        s = str(n % 7) + s  
        n //= 7  
    return s  
  
for n in range(1, 2031):  
    k = sm(7 ** 170 + 7 ** 100 - n)  
    if k.count('0') == 71:  
        print(n)
```



# Задание 15

На числовой прямой даны два отрезка:  $P = [15; 40]$  и  $Q = [21; 63]$ . Укажите наименьшую возможную длину такого отрезка  $A$ , для которого логическое выражение  $(x \in P) \rightarrow (((x \in Q) \wedge \neg(x \in A)) \rightarrow \neg(x \in P))$  истинно (т.е. принимает значение 1) при любом значении переменной  $x$

```
p = list(range(15, 41))
q = list(range(21, 64))
a = []
for x in range(0, 1000):
    if not((x in p) <= (((x in q) and (x not in a)) <= (x not in p))):
        a.append(x)
print(len(a) - 1)
```

**(Л. Шастин)** Для какого наименьшего целого неотрицательного числа  $A$  формула

$$(x \leq 19) \vee (y < 2x + A - 50) \vee (y > 17)$$

тождественно истинна, т.е. принимает значение 1 при любых целых неотрицательных  $x$  и  $y$ ?

```
for a in range(1, 1001):
    if all(((x <= 19) or (y < 2 * x + a - 50) or (y > 17)) for x in range(1, 1001) \
        for y in range(1, 1001)):
        print(a)
```

# Задание 15

Обозначим через  $\text{ДЕЛ}(n, m)$  утверждение «натуральное число  $n$  делится без остатка на натуральное число  $m$ ». Для какого наибольшего натурального числа  $A$  логическое выражение  $\neg \text{ДЕЛ}(x, A) \rightarrow (\text{ДЕЛ}(x, 14) \rightarrow \neg \text{ДЕЛ}(x, 4))$  истинно (т.е. принимает значение 1) при любом целом положительном значении переменной  $x$ ?

```
for a in range(1, 1001):  
    if all(((x % a != 0) <= ((x % 14 == 0) <= (x % 4 != 0))) for x in range(1, 1001)):  
        print(a)
```

**(М. Попков)** Обозначим через  $m \& n$  поразрядную конъюнкцию неотрицательных целых чисел  $m$  и  $n$ . Определите наибольшее натуральное число  $A$ , такое что выражение  $(x \& A \neq 0) \rightarrow ((x \& 168 = 0) \rightarrow (x \& 69 \neq 0))$  тождественно истинно (то есть принимает значение 1 при любом натуральном значении переменной  $x$ )?

```
for a in range(1, 1001):  
    if all(((x & a != 0) <= ((x & 168 == 0) <= (x & 69 != 0))) for x in range(1, 1001)):  
        print(a)
```

# Задание 16

Алгоритм вычисления значения функции  $F(n)$ , где  $n$  – натуральное число, задан следующими соотношениями:

$F(n) = 1$  при  $n = 1$ ;

$F(n) = (n - 1) \times F(n - 1)$ , если  $n > 1$ .

Чему равно значение выражения  $(F(2024) + 2 \times F(2023)) / F(2022)$ ?

```
from sys import *
setrecursionlimit(3500)
4 usages
def f(n):
    if n == 1:
        return 1
    if n > 1:
        return (n - 1) * f(n - 1)
print((f(2024) + 2 * f(2023)) // f(2022))
```

# Задание 17

В файле содержится последовательность натуральных чисел.

Её элементы могут принимать целые значения от 1 до 100 000 включительно. Определите количество пар последовательности, в которых остаток от деления хотя бы одного из элементов на 16 равен минимальному элементу последовательности. В ответе запишите количество найденных пар, затем максимальную из сумм элементов таких пар. В данной задаче под парой подразумевается два идущих подряд элемента последовательности.

```
f = open('17_17873.txt')
s = [int(x) for x in f]
res = [] #список под ответ
for i in range(len(s) - 1):
    if s[i] % 16 == min(s) or s[i + 1] % 16 == min(s):
        res.append(s[i] + s[i + 1])
print(len(res), max(res))
```



# Задание 17

В файле содержится последовательность целых чисел. Элементы последовательности могут принимать целые значения от -100 000 до 100 000 включительно. Определите количество пар элементов последовательности, в которых только один из двух элементов оканчивается на 21 и является пятизначным числом, а сумма квадратов элементов пары не меньше квадрата максимального элемента последовательности, оканчивающегося на 21 и являющегося пятизначным числом. В ответе запишите количество найденных пар чисел, затем максимальную из сумм элементов таких пар.

В данной задаче под парой подразумевается два идущих подряд элемента последовательности.

```
f = open('17_16383.txt')
s = [int(x) for x in f]
res = [] #список под ответы
m = [] #список под макс элемент
for i in range(len(s)):
    if (abs(s[i]) % 100 == 21) and (len(str(abs(s[i]))) == 5):
        m.append(s[i])
for i in range(len(s) - 1):
    if ((abs(s[i]) % 100 == 21 and len(str(abs(s[i]))) == 5) and (abs(s[i + 1]) % 100 != 21 and len(str(abs(s[i + 1]))) != 5)) or \
        ((abs(s[i]) % 100 != 21 and len(str(abs(s[i]))) != 5) and (abs(s[i + 1]) % 100 == 21 and len(str(abs(s[i + 1]))) == 5)):
        if (s[i] ** 2 + s[i + 1] ** 2) >= max(m) ** 2:
            res.append(s[i] + s[i + 1])
print(len(res), max(res))
```



# Задание 19

Два игрока, Петя и Ваня, играют в следующую игру. Перед игроками лежит куча камней. Игроки ходят по очереди, первый ход делает Петя. За один ход игрок может: убрать из кучи два камня или убрать из кучи пять камней или уменьшить количество камней в куче в три раза (количество камней, полученное при делении, округляется до меньшего). Например, из кучи в 20 камней за один ход можно получить кучу из 18, 15 или 6 камней. Игра завершается, когда количество камней в куче становится не более 19.

Победителем считается игрок, сделавший последний ход, то есть первым получивший кучу, в которой будет 19 или меньше камней. В начальный момент в куче было  $S$  камней,  $S \geq 20$ .

Будем говорить, что игрок имеет выигрышную стратегию, если он может выиграть при любых ходах противника.

Укажите минимальное значение  $S$ , при котором Петя не может выиграть за один ход, но при любом ходе Пети Ваня может выиграть своим первым ходом

```
def f(a, n):
    if a <= 19 or n > 2: #условие для завершения игры
        return n == 2
    if n % 2 == 0: #4 и 5 строка убираются, если в задаче неудачный ход
        return all([f(a - 2, n + 1), f(a - 5, n + 1), f(a // 3, n + 1)])
    return any([f(a - 2, n + 1), f(a - 5, n + 1), f(a // 3, n + 1)])
for a in range(20, 1000):
    if f(a, 0):
        print(a)
```

# Задание 20

Для игры, описанной в задании 19, найдите два наименьших значения  $S$ , при которых у Пети есть выигрышная стратегия, причём одновременно выполняются два условия:

- Петя не может выиграть за один ход;
- Петя может выиграть своим вторым ходом независимо от того, как будет ходить Ваня.


Найденные значения запишите в ответе в порядке возрастания.

```
def f(a, n):  
    if a <= 19 or n > 3: #условие для завершения игры  
        return n == 3  
    if n % 2 == 1: |  
        return all([f(a - 2, n + 1), f(a - 5, n + 1), f(a // 3, n + 1)])  
        return any([f(a - 2, n + 1), f(a - 5, n + 1), f(a // 3, n + 1)])  
for a in range(20, 1000):  
    if f(a, 0):  
        print(a)
```

# Задание 21

Для игры, описанной в задании 19, найдите минимальное значение  $S$ , при котором одновременно выполняются два условия:

- у Вани есть выигрышная стратегия, позволяющая ему выиграть первым или вторым ходом при любой игре Пети;
- у Вани нет стратегии, которая позволит ему гарантированно выиграть первым ходом.

```
def f(a, n):  
    if a <= 19 or n > 4: #условие для завершения игры  
        return n == 2 or n == 4  
    if n % 2 == 0:  
         return all([f(a - 2, n + 1), f(a - 5, n + 1), f(a // 3, n + 1)])  
    return any([f(a - 2, n + 1), f(a - 5, n + 1), f(a // 3, n + 1)])  
for a in range(20, 1000):  
    if f(a, 0):  
        print(a)
```

# Задание 23

Исполнитель преобразует число на экране. У исполнителя есть две команды, которые обозначены латинскими буквами:

**A. Вычти 2**

**B. Найди целую часть от деления на 2**

Программа для исполнителя – это последовательность команд.

Сколько существует программ, для которых при исходном числе 38 результатом является число 2 и при этом траектория вычислений содержит число 16?

Траектория вычислений программы – это последовательность результатов выполнения всех команд программы. Например, для программы **ABB** при исходном числе 13 траектория состоит из чисел 11, 5, 2.

```
def f(a, b):  
    if a < b:  
        return 0  
    if a == b:  
        return 1  
    if a > b:  
        return f(a - 2, b) + f(a // 2, b)  
print(f(38, 16) * f(16, 2))
```

# Задание 25

Назовём маской числа последовательность цифр, в которой также могут встречаться следующие символы:

- символ «?» означает ровно одну произвольную цифру;
- символ «\*» означает любую последовательность цифр произвольной длины; в том числе «\*» может задавать и пустую последовательность.

Например, маске 123\*4?5 соответствуют числа 123405 и 12300405.

Среди натуральных чисел, не превышающих  $10^{10}$ , найдите все числа, соответствующие маске 3?12?14\*5, делящиеся на 1917 без остатка.

В ответе запишите в первом столбце таблицы все найденные числа в порядке возрастания, а во втором столбце – соответствующие им результаты деления этих чисел на 1917.

Количество строк в таблице для ответа избыточно.

```
from fnmatch import *
for x in range(1917, 10 ** 10 + 1, 1917):
    if fnmatch(str(x), '3?12?14*5'):
        print(x, x // 1917)
```



# Задание 25

Пусть  $M$  – сумма минимального и максимального натуральных делителей целого числа, не считая единицы и самого числа. Если таких делителей у числа нет, то считаем значение  $M$  равным нулю.

Напишите программу, которая перебирает целые числа, большие 800 000, в порядке возрастания и ищет среди них такие, для которых

$M$  оканчивается на 4. В ответе запишите в первом столбце таблицы первые пять найденных чисел в порядке возрастания, а во втором столбце – соответствующие им значения  $M$ .

Например, для числа 20  $M = 2 + 10 = 12$ .

Количество строк в таблице для ответа избыточно

```
def div(n):  
    d = set()  
    for i in range(2, int(n ** 0.5) + 1):  
        if n % i == 0:  
            d |= {i, n // i}  
    return sorted(d)  
for n in range(800001, 800100):  
    d = div(n)  
    if len(d) > 0 and (max(d) + min(d)) % 10 == 4:  
        print(n, max(d) + min(d))
```