



# Открытая лекция Новая форма сдачи ЕГЭ по информатике и ИКТ

ст. препод. каф. ПОВТиАС  
Голубничий Артем Александрович  
[artem@golubnichij.ru](mailto:artem@golubnichij.ru)

Абакан, 2020

**2** Логическая функция  $F$  задаётся выражением:

$$(x \rightarrow y) \wedge (x \vee \neg z) \wedge (x \equiv \neg w).$$

Дан частично заполненный фрагмент, содержащий **неповторяющиеся** строки таблицы истинности функции  $F$ .

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

Переменная 1	Переменная 2	Переменная 3	Переменная 4	Функция
???	???	???	???	$F$
1	1		1	1
	1			1
	1		1	1

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.

```
def f(w, x, y, z):
    return (not x or y) and (x or not z) and (x == (not w))
print('w x y z')
for w in 0, 1:
    for x in 0, 1:
        for y in 0, 1:
            for z in 0, 1:
                if f(w, x, y, z) == 1:
                    print(w, x, y, z)
```

---

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

5

Алгоритм получает на вход натуральное число  $N$  и строит по нему новое число  $R$  следующим образом:

1. Строится двоичная запись числа  $N$ .
2. Складываются все цифры полученной двоичной записи. В конец записи (справа) дописывается остаток от деления суммы на 2.
3. Предыдущий пункт повторяется для записи с добавленной цифрой.
4. Результат переводится в десятичную систему.

*Пример.* Дано число  $N = 13$ . Алгоритм работает следующим образом:

1. Двоичная запись числа  $N$ : 1101.
2. Сумма цифр двоичной записи 3, остаток от деления на 2 равен 1, новая запись 11011.
3. Сумма цифр полученной записи 4, остаток от деления на 2 равен 0, новая запись 110110.
4. Результат работы алгоритма  $R = 54$ .

При каком наименьшем числе  $N$  в результате работы алгоритма получится  $R > 154$ ? В ответе запишите это число в десятичной системе счисления.

```
for n in range(64):
    r = [int(bin(n)[2:])]
    r += [sum(r) % 2]
    r += [sum(r) % 2]
    r = [str(i) for i in r]
    if int(''.join(r), 2) > 154:
        print(n)
        break
```

---

**6**

Определите, при каком наименьшем введённом значении переменной  $s$  программа выведет число 128. Для Вашего удобства программа представлена на четырёх языках программирования.

<b>Паскаль</b>	<b>Python</b>
<pre>var s, n: integer; begin   readln(s);   n := 2;   while s &lt; 37 do begin     s := s + 3;     n := n * 2   end;   writeln(n) end.</pre>	<pre>s = int(input()) n = 2 while s &lt; 37:   s = s + 3   n = n * 2 print(n)</pre>

```
for i in range(1, 100):
    s = i
    n = 2
    while s < 37:
        s = s + 3
        n = n * 2
    if n == 128:
        print(i)
# 1 - минимальное
# последнее - максимальное
# Общее количество считается по выводу
```

---

19  
20  
21

8

Игорь составляет 8-буквенные коды из букв И, Г, О, Р, Ъ. Буквы О и Ъ должны встречаться в коде ровно по одному разу, при этом буква Ъ не может стоять на первом месте. Остальные допустимые буквы могут встречаться произвольное количество раз или не встречаться совсем. Сколько различных кодов может составить Игорь?



```
from itertools import product
#возможно понадобится permutation (для перестановок )
words = [''.join(i) for i in list(product("ИГОРЬ",
repeat=8))]
k = 0
for word in words:
    if word[0] != "Б" and word.count("О") == 1 and
word.count("Б")==1:
        k += 1
print(k)
```

---

35721

Какая строка получится в результате применения приведённой ниже программы к строке, состоящей из 101 единицы?

НАЧАЛО

ПОКА **нашлось** (111)

**заменить** (111, 22)

**заменить** (222, 11)

КОНЕЦ ПОКА

КОНЕЦ

```
s = "1" * 101
while s.find("111") >= 0:
    s = s.replace("111", "22", 1)
    s = s.replace("222", "11", 1)
print(s)
```

---

112

14

Значение выражения  $81^{15} + 3^{22} - 27$  записали в системе счисления с основанием 9. Сколько цифр 8 содержится в этой записи?

```
num = 81**15 + 3**22 - 27
base = 9
res = ""
while num > 0:
    res = str(num % base) + res
    num = num // base
print(res.count("8"))
```

-----

9

**16**

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

$$F(1) = 1;$$

$$F(n) = n + F(n-2), \text{ если } n > 1 \text{ и при этом } n \text{ нечётно};$$

$$F(n) = n \times F(n-1), \text{ если } n \text{ чётно.}$$

Чему равно значение функции  $F(40)$ ?



```
def F(n):  
    if n == 1:  
        return 1  
    if n % 2 == 1:  
        return n + F(n - 2)  
    else:  
        return n * F(n - 1)  
print(F(40))
```

---

16000

Ниже на четырёх языках программирования записана программа, которая вводит натуральное число  $x$ , выполняет преобразования, а затем выводит одно число. Укажите **наименьшее** возможное значение  $x$ , при вводе которого программа выведет число 40.

Python	Паскаль
<pre>x = int(input()) a = 1 while x &gt; 0:     a *= x % 7     x = x // 7 print(a)</pre>	<pre>var x, a: integer; begin     readln(x);     a := 1;     while x &gt; 0 do begin         a := a * (x mod 7);         x := x div 7     end;     writeln(a) end.</pre>



```
for i in range(1000):  
    x = i  
    a = 1  
    while x > 0:  
        a *= x % 7  
        x = x // 7  
    if a == 40:  
        print(i)  
        break # для наименьшего
```

---

23

Исполнитель преобразует число на экране.

У исполнителя есть две команды, которым присвоены номера:

**1. Прибавить 1**

**2. Умножить на 3**

Первая команда увеличивает число на экране на 1, вторая умножает его на 3.

Программа для исполнителя – это последовательность команд.

Сколько существует программ, которые преобразуют исходное число 1 в число 65, и при этом траектория вычислений содержит число 20?

Траектория вычислений – это последовательность результатов выполнения всех команд программы. Например, для программы 212 при исходном числе 4 траектория будет состоять из чисел 12, 13, 39.

```
f = [0] * 66
f[1] = 1
for n in range(2, 21):
    f[n] = f[n-1]
    if n % 3 == 0 and n // 3 >= 1:
        f[n] += f[n // 3]

for n in range(21, 66):
    f[n] = f[n-1]
    if n % 3 == 0 and n // 3 >= 20:
        f[n] += f[n // 3]
print(f[65])
```

---