

Шаблоны для решения заданий ЕГЭ по информатике 2023

Задание 2

Миша заполнял таблицу истинности логической функции F

$$(w \rightarrow z) \wedge ((y \rightarrow x) \equiv (z \rightarrow y)),$$

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	1
	0	1		1
1	0	0	1	1

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.

Логическая операция	Представление в Питоне
Отрицание $\neg$	not()
Логическое умножение $\wedge$	and
Логическое сложение $\vee$	or
Следование $A \rightarrow B$	A <=B
Тождественность $\equiv$	==

Решение:

Решать задачу будем с помощью шаблона на языке Python (Питон).

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

В задаче у нас 4 переменные, значит, формируем 4 вложенных цикла. В каждом цикле перебираем все возможные значения для конкретной переменной. Мы перебираем значения 0 и 1.

<i>x</i>	<i>w</i>	<i>z</i>	<i>y</i>	<i>F</i>
1	0	0	0	1
1	0	1	1	1
1	0	0	1	1

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

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 сумма цифр в двоичной записи числа чётная, то к этой записи справа дописывается 0, а затем два левых разряда заменяются на 10;
- б) если сумма цифр в двоичной записи числа нечётная, то к этой записи справа дописывается 1, а затем два левых разряда заменяются на 11.

Полученная таким образом запись является двоичной записью искомого числа R .

Например, для исходного числа $6_{10} = 110_2$ результатом является число $1000_2 = 8_{10}$, а для исходного числа $4_{10} = 100_2$ результатом является число $1101_2 = 13_{10}$.

Укажите минимальное число N , после обработки которого с помощью этого алгоритма получается число R , большее 40. В ответе запишите это число в десятичной системе счисления.

Внимательно читай , что нужно найти! В двоичной системе:

- четные числа оканчиваются на 0, нечетные – на 1;
- числа, которые делятся на 4, оканчиваются на 00, и т.д.; числа, которые делятся на 2^k , оканчиваются на k нулей

1. В программе перебираем натуральные числа от 1 до 1000 с помощью цикла **for**. Каждое число подставляем в описанный алгоритм, в надежде получить в результате число **r**, удовлетворяющие условию задачи.
2. С помощью функции **bin(n)** переводим число **n** в двоичный вид. Получаем результат в виде **строки n1**. **[2:]** отбрасывает символы в двух старших левых разрадах.
3. Чтобы найти сумму цифр получившейся двоичной записи, достаточно подсчитать количество единиц в строке **n1**. Это можно сделать, применив функцию **.count()** к строке **n1**.
4. Остаток от деления суммы цифр на 2 записываем с помощью - **%**.
5. Чтобы обратно превратить строку двоичной записи в десятичное число, используем функцию **int()**, указав параметр **2**.

5_демо 2023.py - C:/Users/classadmin/Desktop/5_демо 2023.py (3.4.3rc1)

File Edit Format Run Options Window Help

```
for n in range(1,1000):
    n1=bin(n)[2:]
    if n1.count('1')%2==0:
        n1=n1+'0'
        n1='10'+n1[2:]
    else:
        n1=n1+'1'
        n1='11'+n1[2:]

    r=int(n1,2)
    if r>40:
        print(n)
        break
```

Ответ: 16

Задание 8

Формула для вычисления числа перестановок с повторениями; для двух разных символов она выглядит так: $P(n_A, n_*) = \frac{(n_A + n_*)!}{n_A! n_*!}$

Здесь n_A – количество букв А, n_* – количество звёздочек и восклицательный знак обозначает факториал натурального числа.

Число не может начинаться с 0! 0,2,4,6,8 четные 1,3,5,7,9 нечетные

Петя составляет шестибуквенные слова перестановкой букв слова КАБАЛА. При этом он избегает слов с двумя подряд одинаковыми буквами. Сколько всего различных слов может составить Петя?

№8_3py.py - C:\Users\classadmin\Desktop\номер 8\№8_3py.py (3.8.0)

File Edit Format Run Options Window Help

```
import itertools
k=0
for i in set(itertools.permutations ('КАБАЛА')):
    s=''.join(i)
    if s.count('AA')==0:
        k=k+1

print(k)
```

Задание 8

Сергей составляет 6-буквенные коды из букв С, О, Л, О, В, Е, Й. Буква Й может использоваться в коде не более одного раза, при этом она не может стоять на первом месте, на последнем месте и рядом с буквой Е. Все остальные буквы могут встречаться произвольное количество раз или не встречаться совсем. Сколько различных кодов может составить Сергей?

№8_5.py - C:\Users\classadmin\Desktop\номер 8\№8_5.py (3.8.0)

File Edit Format Run Options Window Help

```
import itertools
k=0
for i in set(itertools.product ('СОЛОВЕЙ', repeat=6)):
    s=''.join(i)
    if s.count('Й')<=1 and s[0]!='Й' and s[5]!='Й' and s.count('ЙЕ')==0\
        and s.count('ЕЙ')==0 :
        k=k+1

print(k)
```

Задание 12

НАЧАЛО

ПОКА нашлось (21) ИЛИ нашлось (31) ИЛИ нашлось (23)

 ЕСЛИ нашлось (21)

 ТО заменить (21, 12)

 КОНЕЦ ЕСЛИ

 ЕСЛИ нашлось (31)

 ТО заменить (31, 13)

 КОНЕЦ ЕСЛИ

 ЕСЛИ нашлось (23)

 ТО заменить (23, 32)

 КОНЕЦ ЕСЛИ

КОНЕЦ ПОКА

КОНЕЦ

На вход приведённой ниже программе поступает строка, содержащая 50 цифр 1, 50 цифр 2 и 50 цифр 3, расположенных в произвольном порядке. Запишите без разделителей символы, которые имеют порядковые номера 10, 90 и 130 в получившейся строке.

Решение:

Здесь сказали, что начальный порядок цифр в строке неизвестен. А просят написать конкретные данные на выходе. Отсюда делаем вывод, что любой порядок в начале приводит к одному и тому же результату.

В строке заведём цифры по порядку: в начале единицы, потом двойки, потом тройки.

```
s = '1'*50 + '2'*50 + '3'*50

while ('21' in s) or ('31' in s) or ('23' in s):
    if '21' in s:
        s = s.replace('21', '12', 1)
    if '31' in s:
        s = s.replace('31', '13', 1)
    if '23' in s:
        s = s.replace('23', '32', 1)

print(s[9] + s[89] + s[129])
```

Чтобы обратиться к конкретному символу в строке, используем конструкцию **s[позиция символа]**. Но нумерация начинается с нуля.

В ответе запишем 132.

Ответ: 132

Задание 12

```
НАЧАЛО
  ПОКА нашлось (>1) ИЛИ нашлось (>2) ИЛИ нашлось (>3)
    ЕСЛИ нашлось (>1)
      ТО заменить (>1, 2>)
    КОНЕЦ ЕСЛИ
    ЕСЛИ нашлось (>2)
      ТО заменить (>2, 2>1)
    КОНЕЦ ЕСЛИ
    ЕСЛИ нашлось (>3)
      ТО заменить (>3, 1>2)
    КОНЕЦ ЕСЛИ
  КОНЕЦ ПОКА
КОНЕЦ
```

На вход приведённой ниже программе поступает строка, начинающаяся с символа «>», а затем содержащая 28 цифр 1, 18 цифр 2 и 35 цифр 3, расположенных в произвольном порядке. Определите сумму числовых значений цифр строки, получившейся в результате выполнения программы. Так, например, если результат работы программы представлял бы собой строку, состоящую из 50 цифр 4, то верным ответом было бы число 200.

Решение:

Здесь снова результат будет одним и тем же независимо от того, как будут расположены числа в начале.

```
s = '>' + '1'*28 + '2'*18 + '3'*35

while ('>1' in s) or ('>2' in s) or ('>3' in s):
    if '>1' in s:
        s = s.replace('>1', '22>', 1)
    if '>2' in s:
        s = s.replace('>2', '2>1', 1)
    if '>3' in s:
        s = s.replace('>3', '1>2', 1)

# Находим сумму цифр в строке
sm=0
for i in range(0, len(s)):
    if s[i]!='>':
        sm = sm + int(s[i])

print(sm)
```

Чтобы найти сумму цифр, мы пробегаем по всей строке с помощью цикла range. Если символ не равен ">", то превращаем его в число с помощью функции int() и суммируем в переменную sm.

Получается 465.

Ответ: 465

Задание №14 Значение арифметического выражения: $125 + 25^3 + 5^9$ – записали в системе счисления с основанием 5.

#Количество 0 ?

```
n = 125 + 25 ** 3 + 5 ** 9
```

```
k0 = 0
```

```
while n > 0:
```

```
    if n % 5 == 0:
```

```
        k0 += 1
```

```
    n = n // 5
```

```
print(k0)
```

#Для суммы цифр

```
n = 125 + 25 ** 3 + 5 ** 9
```

```
s = 0
```

```
while n > 0:
```

```
    s += n % 5
```

```
    n = n // 5
```

```
print(s)
```

Задание 14

Задача(Новый тип, закрепление)

(Богданов) Операнды арифметического выражения записаны в системе счисления с основанием 17:

$$9759x_{17} + 3x108_{17}$$

В записи чисел переменной x обозначена неизвестная цифра из алфавита 17-ричной системы счисления. Определите наименьшее значение x , при котором значение данного арифметического выражения кратно 11. Для найденного значения x вычислите частное от деления значения арифметического выражения на 11 и укажите его в ответе в десятичной системе счисления. Основание системы счисления в ответе указывать не нужно.

Решение:

Решим задание с помощью предыдущего шаблона на языке Python.

```
for x in range(0, 17):
    a=x*17**0 + 9*17**1 + 5*17**2 + 7*17**3 + 9*17**4
    b=8*17**0 + 0*17**1 + 1*17**2 + x*17**3 + 3*17**4
    if (a+b)%11==0:
        print(x, (a+b)//11)
```

Ответ: 95306

Задание 14

Задача (Новый тип, две переменные)

(В. Шубинкин) Числа M и N записаны в системах счисления с основаниями 15 и 13 соответственно.

$$M = 2y23x5_{15}, N = 67x9y_{13}$$

В записи чисел переменными x и y обозначены допустимые в данных системах счисления неизвестные цифры. Определите наименьшее значение натурального числа A , при котором существуют такие x, y , что $M + A$ кратно N .

Решение:

Принцип решения данной задачи похож на решение [15 задания](#) из ЕГЭ по информатике.

```
for A in range(1, 5000):
    for x in range(0, 13):
        for y in range(0, 13):
            M=5*15**0 + x*15**1 + 3*15**2 + 2*15**3 + y*15**4 + 2*15**5
            N=y*13**0 + 9*13**1 + x*13**2 + 7*13**3 + 6*13**4
            if (M+A)%N==0:
                print(A)
```

Нужно найти A , значит, начинаем перебирать A . Идём от 1, т.к. речь идёт о натуральных числах. Перебираем x и y . Они могут принимать значения из алфавита в 13-ой системе. Берём меньшую, т.к. эти переменные и в первом числе, и во втором одинаковые.

Если выполняется условие задачи, то нам интересно такое A при котором это произошло.

В этой задаче A получается достаточно большим, поэтому перебираем эту переменную до 5000.

Ответ: 1535

Задание 15

Читай внимательно, ЦЕЛЫЕ НЕОТРИЦАТЕЛЬНЫЕ ИЛИ ПОЛОЖИТЕЛЬНЫЕ ЗНАЧЕНИЯ!

ДЕЛ (Для какого наибольшего натурального числа A формула $(\neg \text{ДЕЛ}(x, A) \wedge \text{ДЕЛ}(x, 21)) \rightarrow \text{ДЕЛ}(x, 14)$ тождественно истинна (то есть принимает значение 1 при любом натуральном значении переменной x)?)

```
for A in range(1,100):  
    flag = 0  
    for x in range(1,1000):  
        if (((x % A != 0) and (x % 21 == 0)) <= (x % 14 == 0)) == 0:  
            flag = 1  
    if flag == 0:  
        print(A)
```

X и Y

```
for A in range(1,200):  
    flag = 0  
    for x in range(1,100):  
        for y in range(1,100):  
            if ((y*y <= A) <= (y <= 10)) and ((x <= 9) <= (x * x < A)) == 0:  
                flag = 1  
    if flag == 0:  
        print(A)
```

Побитовая конъюнкция

```
for A in range(0,100):  
    flag = 0  
    for x in range(0,1000):  
        if ((x & 49 == 0) <= ((x & 28 != 0) <= (x & A != 0))) == 0:  
            flag = 1  
            break  
    if flag == 0:  
        print(A)
```


Обозначим через $\text{ДЕЛ}(n, m)$ утверждение «натуральное число n делится без остатка на натуральное число m ». Для какого **наименьшего** натурального числа A формула

$$(\text{ДЕЛ}(x, 2) \rightarrow \neg \text{ДЕЛ}(x, 3)) \vee (x + A \geq 100)$$

тождественно истинна (т. е. принимает значение 1) при любом натуральном значении переменной x ?

Источник: Демонстрационная версия ЕГЭ–2023 по информатике

15_del.py - C:/Users/classadmin/Desktop/15_del.py (3.4.3rc1)

File Edit Format Run Options Window Help

```
for a in range (1, 500):  
    flag=0  
    for x in range (1, 500):  
        if ( ( (x%2==0) <= (x%3!=0) ) or (x+a>=100) ) ==0:  
            flag=1  
            break  
    if flag==0:  
        print(a)  
        break
```

15

Обозначим через $\text{ТРЕУГ}(n, m, k)$ утверждение «существует невырожденный треугольник с длинами сторон n, m и k ».

Для какого наибольшего натурального числа A формула

$$\neg((\text{ТРЕУГ}(x, 11, 18) \equiv (\neg(\text{МАКС}(x, 5) > 15))) \wedge \text{ТРЕУГ}(x, A, 5))$$

тождественно истинна (т. е. принимает значение 1) при любом натуральном значении переменной x ?

Примечание. $\text{МАКС}(a, b) = a$, если $a > b$ и $\text{МАКС}(a, b) = b$, если $a \leq b$.

15treug.py - C:\Users\classadmin\Desktop\15treug.py (3.4.3rc1)

File Edit Format Run Options Window Help

```
def treug(n, m, k):  
    return (n+m>k) and (m+k>n) and (k+n>m)  
for a in range(1, 500):  
    flag=0  
    for x in range(1, 500):  
        if not ((treug(x, 11, 18) == (not (max(x, 5) > 15))) and treug(x, a, 5)) == False:  
            flag=1  
            break  
    if flag==0:  
        print(a)
```

Задание 16

Задача (Стандартная)

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

$F(n) = 1$ при $n = 1$;

$F(n) = n + F(n - 1)$, если n – чётно,

$F(n) = 3 \times F(n - 2)$, если $n > 1$ и при этом n – нечётно.

Чему равно значение функции $F(25)$?

Решение:

Напишем программу для решения данной задачи. В начале опишем все правила, которые даны в условии задачи для функции. В основной части программы запустим эту функцию.

```
# Сама функция
def F(n):
    if n==1: return 1
    if n%2==0: return n+F(n-1)
    if n>1 and n%2!=0: return 3*F(n-2)

# Основная часть программы
print(F(25))
```

После запуска рекурсивной функции программа выведет ответ **531441**.

Выражение $n\%2 \neq 0$ (остаток от деления на "2" не равен нулю) обозначает нечётное число. Выражение $n\%2==0$ обозначает чётное число.

Ответ: 531441

Задание 16

Задача (Новое веяние)

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

$$F(n) = 2, \text{ если } n = 1,$$
$$F(n) = 2 \cdot F(n - 1), \text{ если } n > 1.$$

Чему равно значение выражения $F(1900)/2^{1890}$?

Решение:

 *16_новая.py - C:/Users/classadmin/Desktop/16_новая.py (3.8.0)*

File Edit Format Run Options Window Help

```
f={1:1}
for n in range(2,1901):
    f[n]=2*f[n-1]
print(f[1900]//2**1890)
```

 Python 3.8.0 Shell

File Edit Shell Debug Options Window

```
Python 3.8.0 (tags/bpo-38000:2020-01-20,
tel)] on win32
Type "help", "copyright()", "credits()" or "quit()" for
>>>
=====
512
>>> |
```

В файле содержится последовательность натуральных чисел. Элементы последовательности могут принимать целые значения от 1 до 100 000 включительно. Определите количество пар последовательности, в которых только один из элементов оканчивается на 12, а квадрат суммы элементов пары меньше квадрата максимального элемента последовательности, оканчивающегося на 12. 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.txt')
a=[int(i) for i in f]
k=0
ml=0
ms=0
for i in range (len(a)):
    if a[i]%100==12:
        ml=max (a[i],ml)
for i in range (len(a)-1):
    if (a[i]%100==12 and a[i+1]%100!=12) or (a[i+1]%100==12 and a[i]%100!=12):
        if (a[i]+a[i+1])**2<ml**2:
            k+=1
            ms=max(ms,a[i]+a[i+1])
print (k,ms)
```

Задание 19 (Демо 2022)

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 добавить в кучу один камень или увеличить количество камней в куче в два раза. Для того чтобы делать ходы, у каждого игрока есть неограниченное количество камней. Игра завершается в тот момент, когда количество камней в куче становится не менее 29. Победителем считается игрок, сделавший последний ход, т.е. первым получивший кучу, в которой будет 29 или больше камней. В начальный момент в куче было S камней, $1 \leq S \leq 28$. Будем говорить, что игрок имеет выигрышную стратегию, если он может выиграть при любых ходах противника. Описать стратегию игрока – значит описать, какой ход он должен сделать в любой ситуации, которая ему может встретиться при различной игре противника. В описание выигрышной стратегии не следует включать ходы играющего по этой стратегии игрока, не являющиеся для него безусловно выигрышными, т.е. не являющиеся выигрышными независимо от игры противника. Укажите такое значение S , при котором Петя не может выиграть за один ход, но при любом ходе Пети Ваня может выиграть своим первым ходом.

Введём параметр **p**, который будет олицетворять позицию игры (ход).

	Начальная позиция	Ход Пети	Ход Вани	Ход Пети	Ход Вани	Ход Пети
p	1	2	3	4	5	6

```
def F(x, p):  
    if x>=29 and p==3: return True  
    if x<29 and p==3: return False  
    if x>=29: return False  
  
    if p%2==1:  
        return F(x+1, p+1) and F(x*2, p+1)  
    else:  
        return F(x+1, p+1) or F(x*2, p+1)  
  
for s in range(1, 29):  
    if F(s, 1):  
        print(s)
```

Заводим функцию F. Т.к. у нас одна куча, то она принимает параметры: **x** - количество камней в куче, **p**-позиция игры.

Задание 20 (Демо 2022)

Для игры, описанной в задании 19, найдите два таких 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.

Решение:

Задача точно такая же, как и в 19 задании, только теперь обязательно должен побежать Петя на своём втором ходу ($p=4$), при любой игре Вани.

Пишем тот же шаблон, немного отредактировав его.

```
def F(x, p):
    if x >= 29 and p == 4: return True
    if x < 29 and p == 4: return False
    if x >= 29: return False

    if p % 2 == 0:
        return F(x+1, p+1) and F(x*2, p+1)
    else:
        return F(x+1, p+1) or F(x*2, p+1)

for s in range(1, 29):
    if F(s, 1):
        print(s)
```

Задание 21 (Демо 2022)

Для игры, описанной в задании 19, найдите 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.

Если найдено несколько значений S , в ответе запишите минимальное из них.

Решение:

Опять используем прошлый шаблон, но немного модернизируем.

```
def F(x, p):
    if x>=29 and (p==3 or p==5): return True
    if x<29 and p==5: return False
    if x>=29: return False

    if p%2==1:
        return F(x+1, p+1) and F(x*2, p+1)
    else:
        return F(x+1, p+1) or F(x*2, p+1)

def F1(x, p):
    if x>=29 and p==3: return True
    if x<29 and p==3: return False
    if x>=29: return False

    if p%2==1:
        return F1(x+1, p+1) and F1(x*2, p+1)
    else:
        return F1(x+1, p+1) or F1(x*2, p+1)

for s in range(1, 29):
    if F(s, 1):
        print(s)

print()

for s in range(1, 29):
    if F1(s, 1):
        print(s)
```

Здесь Ваня должен выигрывать либо на первом своём ходе ($p=3$), либо на втором своём ходе ($p=5$).

Задание 19 (Демо 2021)

Два игрока, Петя и Ваня, играют в следующую игру. Перед игроками лежат две кучи камней. Игроки ходят по очереди, первый ход делает Петя. За один ход игрок может добавить в одну из куч (по своему выбору) один камень или увеличить количество камней в куче в два раза. Например, пусть в одной куче 10 камней, а в другой 5 камней; такую позицию в игре будем обозначать $(10, 5)$. Тогда за один ход можно получить любую из четырёх позиций: $(11, 5)$, $(20, 5)$, $(10, 6)$, $(10, 10)$. Для того чтобы делать ходы, у каждого игрока есть неограниченное количество камней.

Игра завершается в тот момент, когда суммарное количество камней в кучах становится не менее 77. Победителем считается игрок, сделавший последний ход, т.е. первым получивший такую позицию, при которой в кучах будет 77 или больше камней.

В начальный момент в первой куче было семь камней, во второй куче – S камней; $1 \leq S \leq 69$.

Будем говорить, что игрок имеет выигрышную стратегию, если он может выиграть при любых ходах противника. Описать стратегию игрока – значит описать, какой ход он должен сделать в любой ситуации, которая ему может встретиться при различной игре противника. В описание выигрышной стратегии не следует включать ходы играющего по этой стратегии игрока, не являющиеся для него безусловно выигрышными, т.е. не являющиеся выигрышными независимо от игры противника.

Известно, что Ваня выиграл своим первым ходом после неудачного первого хода Пети. Укажите минимальное значение S , когда такая ситуация возможна.

```
def F(x, y, p):  
    if x + y >= 77 and p==3: return True  
    if x + y < 77 and p==3: return False  
  
    return F(x+1, y, p+1) or F(x*2, y, p+1) or F(x, y+1, p+1) or F(x, y*2, p+1)  
  
for s in range(1, 70):  
    if F(s, 7, 1):  
        print(s)
```

Заводим функцию `F`. Она принимает параметры: `x` - количество камней в одной куче, `y` - в другой, `p`-позиция игры.

Задание 20 (Демо 2021)

Для игры, описанной в предыдущем задании, найдите два таких 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.

Решение:

Легко переделать из прошлой задачи.

```
def F(x, y, p):
    if x + y >= 77 and p==4: return True
    if x + y < 77 and p==4: return False
    if x + y >= 77: return False

    if p%2==0:
        return F(x+1, y, p+1) and F(x*2, y, p+1) and F(x, y+1, p+1) and F(x, y*2, p+1)
    else:
        return F(x+1, y, p+1) or F(x*2, y, p+1) or F(x, y+1, p+1) or F(x, y*2, p+1)

for s in range(1, 70):
    if F(s, 7, 1):
        print(s)
```

Задание 21 (Демо 2021)

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.

Решение:

Опять используем прошлый шаблон, но немного модернизируем.

```
def F(x, y, p):
    if x + y >= 77 and (p==3 or p==5): return True
    if x + y < 77 and p==5: return False
    if x + y >= 77: return False

    if p%2==1:
        return F(x+1, y, p+1) and F(x*2, y, p+1) and F(x, y+1, p+1) and F(x, y*2, p+1)
    else:
        return F(x+1, y, p+1) or F(x*2, y, p+1) or F(x, y+1, p+1) or F(x, y*2, p+1)

def F1(x, y, p):
    if x + y >= 77 and p==3: return True
    if x + y < 77 and p==3: return False
    if x + y >= 77: return False

    if p%2==1:
        return F1(x+1, y, p+1) and F1(x*2, y, p+1) and F1(x, y+1, p+1) and F1(x, y*2, p+1)
    else:
        return F1(x+1, y, p+1) or F1(x*2, y, p+1) or F1(x, y+1, p+1) or F1(x, y*2, p+1)

for s in range(1, 70):
    if F(s, 7, 1):
        print(s)

print()

for s in range(1, 70):
    if F1(s, 7, 1):
        print(s)
```

Задание 23

Задача (с обязательным узлом, закрепление)

Исполнитель Май17 преобразует число на экране.

У исполнителя есть две команды, которым присвоены номера:

1. Прибавить 1

2. Прибавить 3

Первая команда увеличивает число на экране на 1, вторая увеличивает его на 3. Программа для исполнителя Май17 — это последовательность команд.

Сколько существует программ, для которых при исходном числе 1 результатом является число 17 и при этом траектория вычислений содержит число 9?

Траектория вычислений программы — это последовательность результатов выполнения всех команд программы. Например, для программы 12 при исходном числе 7 траектория будет состоять из чисел 8, 11, 12.

```
def F(x, y):  
    if x == y: return 1  
  
    if x > y: return 0  
  
    if x < y: return F(x+1, y) + F(x+3, y)  
  
print(F(1, 9)*F(9, 17))
```

Задание 23

Задача (с избегаемым узлом)

Исполнитель НечетМ преобразует число на экране. У исполнителя НечетМ две команды, которым присвоены номера:

1. прибавь 1
2. сделай нечётное

Первая из этих команд увеличивает число x на экране на 1, вторая переводит число x в число $2x+1$. Например, вторая команда переводит число 10 в число 21. Программа для исполнителя НечетМ — это последовательность команд. Сколько существует таких программ, которые число 1 преобразуют в число 25, причём траектория вычислений не содержит число 24? Траектория вычислений программы — это последовательность результатов выполнения всех команд программы. Например, для программы 121 при исходном числе 7 траектория будет состоять из чисел 8, 17, 18.

```
def F(x, y):  
    if x == y: return 1  
  
    if x > y or x==24: return 0  
  
    if x < y: return F(x+1, y) + F(x*2+1, y)  
  
print(F(1, 25))
```
