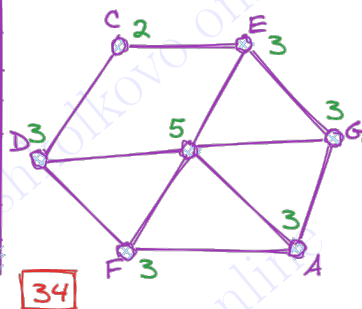


На рисунке схема дорог Н-ского района изображена в виде графа. Так как таблицу и схему рисовали независимо друг от друга, нумерация населённых пунктов в таблице никак не связана с буквенными обозначениями на графе. Определите, какие номера в таблице соответствуют населённым пунктам F и G . В ответе запишите 2 числа в порядке возрастания.

		π_1	π_2	π_3	π_4	π_5	π_6	π_7
B	π_1		*	*	*	*		*
A	π_2	*		*	*			
F/G	π_3	*	*					*
F/G	π_4	*	*			*		
E/D	π_5	*			*		*	
C	π_6					*	*	*
E/D	π_7	*		*			*	*



Алекс заполнил таблицу истинности логической функции F

$$(x \vee y) \wedge \overline{(y \equiv z)} \wedge \overline{w}$$

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

???	???	???	???	F
?	1	?	1	1
0	0	1	?	1
0	?	1	1	1

Решение

```
def f(x,y,z,w):
    return (x or y) and (not (y == z)) and (not w)

print("x y z w f")
for x in range(2):
    for y in range(2):
        for z in range(2):
            for w in range(2):
                if f(x,y,z,w) == 1:
                    print(x, y, z, w, f(x, y, z, w))
```

Проанализируем вывод нашей программы:

```
x y z w f
0 1 0 0 True
1 0 1 0 True
1 1 0 0 True
```

Первый столбик у нас это точно w , потому что все нули мы можем поставить только в первый. Далее мы можем сказать, что второй столбец — это z , ведь только там нет двух единиц, и может быть два нуля. Посмотрим на вторую строку, у нас стоит одна единица, два нуля и неизвестность. Только в данной строке мы можем сделать из нее три нуля и одну единицу, при этом данная единица будет принадлежать — y . Поэтому третий столбец — это y . И четвертый разумеется — это x .

Задача 3

Стандарт, вроде ВПР не требуется

Задача 4

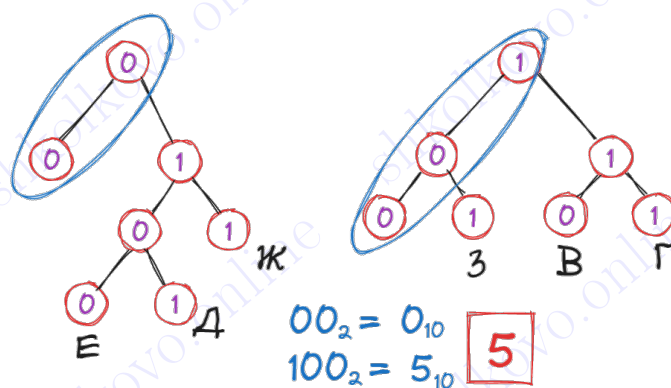
По каналу связи передаются сообщения, содержащие только восемь букв А, Б, В, Г, Д, Е, Ж и З. Для передачи используется двоичный код, удовлетворяющий условию Фано. Кодовые слова для некоторых букв известны:

В	110
Г	111
Д	0101
Е	0100
Ж	011
З	101

Какое наименьшее количество двоичных знаков потребуется для кодирования оставшихся букв? В ответе запишите суммарную длину кодовых слов для букв: А, Б

Примечание: Условие Фано означает, что ни одно кодовое слово не является началом другого слова. Это обеспечивает возможность однозначной расшифровки закодированных сообщений.

Решение



Задача 5

На вход алгоритма подается натуральное число N . Алгоритм строит по нему новое число R следующим образом:

- 1) Строится троичная запись числа N

2) Если N не кратно 3, то остаток от деления на 3 умножается на 5, переводится в троичную запись и дописывается в конец числа

3) Результат R переводится в десятичную систему счисления и выводится на экран
Укажите минимальное число N , после обработки которого автомат получается число, большее 146.

Решение

```
def per(x): #ф-ция перевода в ЗСС
    s = ''
    while x > 0:
        s = str(x % 3) + s
        x //= 3
    return s

for n in range(1, 1000):
    x = per(n) #переводим в ЗСС
    if n % 3 == 0: #проверка исходного числа на делимость на 3
        x += x[-2:] #дописываем две последние цифры
        # в троичной записи
    else:
        x += str(per(n%3 * 5)) #дописываем в конец переведенный
        # в ЗСС остаток от деления на 3, умноженный на 5
    r = int(x, 3) #переводим результат в 10СС
    if r >= 146:
        print(n)
        break
```

Задача 6

Исполнитель Черепаха действует на плоскости с декартовой системой координат. В начальный момент Черепаха находится в начале координат, её голова направлена вдоль положительного направления оси ординат, хвост опущен. При опущенном хвосте Черепаха оставляет на поле след в виде линии. В каждый конкретный момент известно положение исполнителя и направление его движения. У исполнителя существует три команды: **Вперёд** n (где n — целое число), вызывающая передвижение

Черепашки на n единиц в том направлении, куда указывает её голова, **Направо** m (где m — целое число), вызывающая изменение направления движения на m градусов по часовой стрелке, и **Налево** m (где m — целое число), вызывающая изменение направления движения на m градусов против часовой стрелки.

Запись **Повтори** k [**Команда1 Команда2 ... КомандаS**] означает, что последовательность из S команд повторится k раз.

Черепашке был дан для исполнения следующий алгоритм:

Повтори 2 [Вперёд 9 Направо 90 Вперёд 15 Направо 90]

Поднять хвост

Вперёд 12 Направо 90

Опустить хвост

Повтори 2 [Вперёд 6 Направо 90 Вперёд 12 Направо 90]

Определите, сколько точек с целочисленными координатами будут находиться внутри пересечения фигур, ограниченных заданными алгоритмом линиями, включая точки на границах этого пересечения.

Решение в Кумире:

использовать Черепашка

алг

нач

опустить хвост

нц 2 раз

вперед(9)

вправо(90)

вперед(15)

вправо(90)

кц

поднять хвост

вперед(12)

вправо(90)

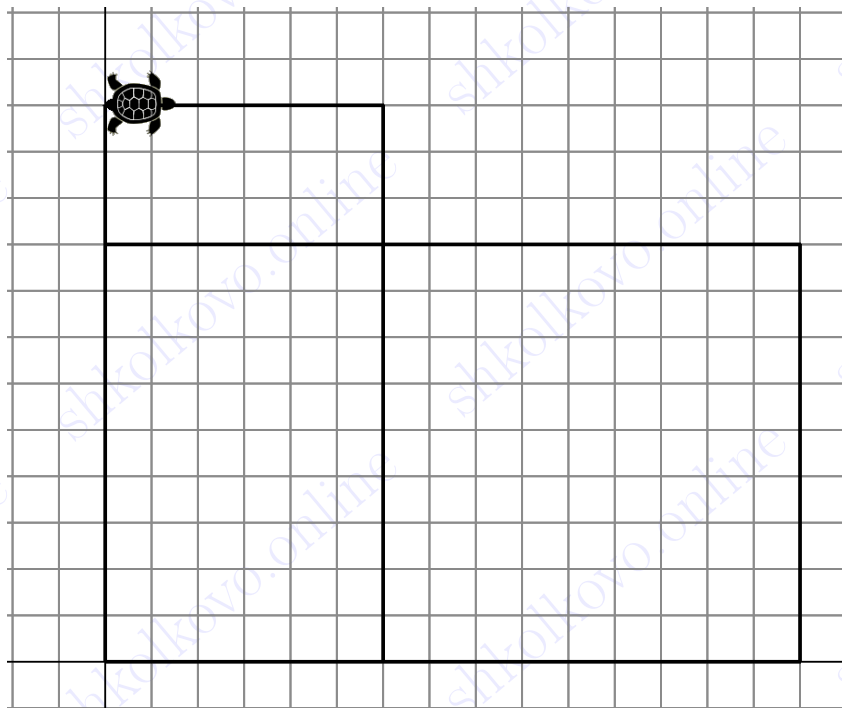
опустить хвост

нц 2 раз

вперед(6)

вправо(90)

вперед(12)
вправо(90)
кц
кон



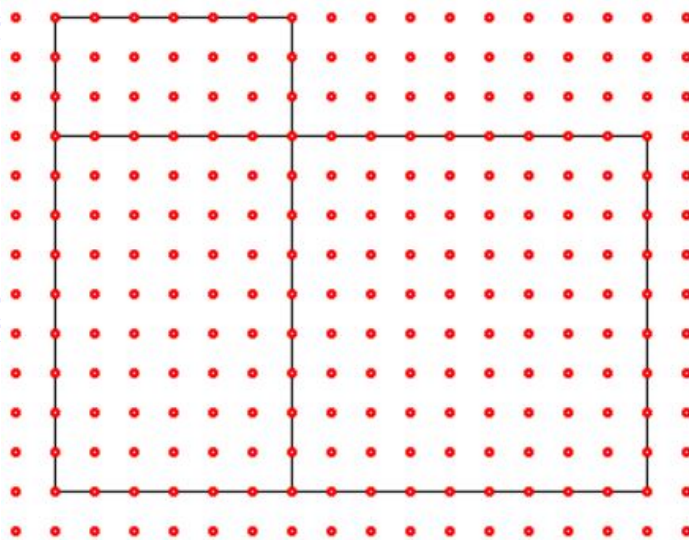
Решение в Питоне:

```
from turtle import *  
left(90)  
k = 20  
tracer(3)  
for i in range(2):  
    forward(9 * k)  
    right(90)  
    forward(15 * k)  
    right(90)  
pu()  
forward(12 * k)  
right(90)  
pd()  
for i in range(2):
```

```

forward(6*k)
right(90)
forward(12 * k )
right(90)
pu()
for x in range(-20, 20):
    for y in range(-20, 20):
        goto(x * k, y * k)
        dot(5, "red")
        dot(2, "white")
mainloop()

```



Задача 7

Сколько секунд потребуется модему, передающему сообщения со скоростью 28800 бит/с, чтобы передать цветное растровое изображение размером 1280×720 пикселей, при условии, что цвет каждого пикселя кодируется 4 Байтами.

Решение

$$\frac{I}{V} = \frac{k \cdot i}{V} = \frac{1280 \cdot 720 \cdot 32}{28800} = 1024 \text{ секунд.}$$

Задача 8

Игорь составляет таблицу кодовых слов для передачи сообщений, каждому сооб-

цению соответствует своё кодовое слово. В качестве кодовых слов Игорь использует пятибуквенные слова, в которых могут быть только буквы ПЯТНИЦА, причём буква Я появляется ровно 1 раз. Каждая из других допустимых букв может встречаться в кодовом слове любое количество раз или не встречаться совсем. На первом месте НЕ может стоять буква Н. Сколько различных кодовых слов может использовать Игорь?

Решение через itertools

```
from itertools import product
count = 0
for word in product('ПЯТНИЦА', repeat=5):
    word = "".join(word)
    if word[0] != "Н" and word.count("Я") == 1:
        count += 1
print(count)
```

Решение без itertools

```
count = 0
for a in 'ПЯТНИЦА':
    for b in 'ПЯТНИЦА':
        for c in 'ПЯТНИЦА':
            for d in 'ПЯТНИЦА':
                for e in 'ПЯТНИЦА':
                    word = a + b + c + d + e
                    if word[0] != "Н" and word.count("Я") == 1:
                        count += 1
print(count)
```

Задача 9

В файле находится таблица, которая содержит в каждой из строк по 7 натуральных чисел. Найдите количество таких строк, в которых два числа повторяются по 2 раза, а 3 других различны, среднее арифметическое неповторяющихся чисел больше среднего арифметического повторяющихся.

Задача 10

С помощью текстового редактора определите, сколько раз, не считая сносок, встречается слово «солдат» (со строчной буквы) в тексте повести Александра Куприна «Поединок». Однокоренные слова не следует учитывать. В ответе укажите только число.

Задача 11

При регистрации в компьютерной системе каждому объекту присваивается идентификатор, состоящий из 65 символов и содержащий только десятичные цифры и символы из 2500-символьного специального алфавита. В базе данных для хранения сведений о каждом идентификаторе отведено одинаковое и минимальное целое число БАЙТ. При этом используется посимвольное кодирование идентификаторов, все символы кодируются одинаковым и минимально возможным количеством бит. Определите объём памяти в Кбайт, необходимый для хранения сведений о 16384 объектах.

В ответе запишите только целое число - количество Кбайт

Решение

Алфавит: 2500 символов, кодируем каждый символ 12 битами.

Один идентификатор кодируется $\frac{12 \cdot 65}{8} = 97.5$, но так как одинаковое и минимальное целое число БАЙТ, то округляем до 98

$$\frac{98 \cdot 16384}{1024} = 1568$$

Задача 12

Исполнитель Редактор получает на вход строку цифр и преобразовывает её. Редактор может выполнять две команды, в которых v и w обозначают последовательности 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 последовательности v , то выполнение команды заменить (v, w) не меняет эту строку.

Б) Нашлось (v) .

Эта команда проверяет, встречается ли последовательность 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.

ПОКА нашлось (52) ИЛИ нашлось (1122) ИЛИ нашлось (2222)

ЕСЛИ нашлось (52)

ТО заменить (52, 11)

КОНЕЦ ЕСЛИ

ЕСЛИ нашлось (2222)

ТО заменить (2222, 5)

КОНЕЦ ЕСЛИ

ЕСЛИ нашлось (1122)

ТО заменить (1122, 25)

КОНЕЦ ЕСЛИ

КОНЕЦ ПОКА

КОНЕЦ

На вход приведённой выше программе поступает строка, начинающаяся с цифры «5», а затем содержащая n цифр «2» ($3 < n < 10000$). Определите наименьшее значение n , при котором сумма цифр в строке, получившаяся в результате выполнения программы, равна 64.

Решение

```
for n in range(4,10000):
    s = '5' + '2' * n
    while ('52' in s) or ('1122' in s) or ('2222' in s):
        if ('52' in s):
            s = s.replace('52', '11', 1)
        if ('2222' in s):
            s = s.replace('2222', '5', 1)
        if ('1122' in s):
```

```

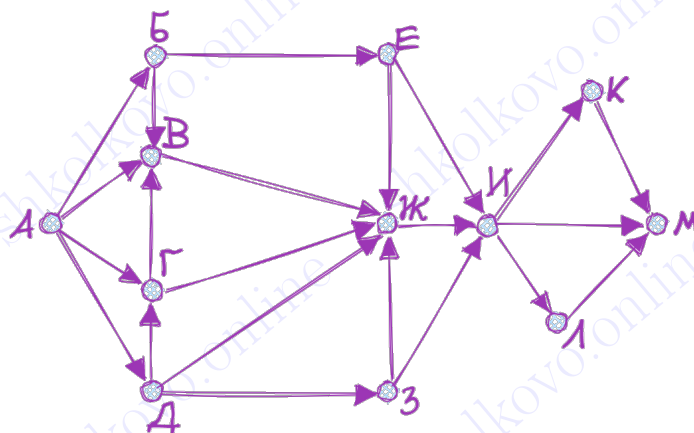
s = s.replace('1122', '25', 1)
r = s.count('1') * 1 + s.count('2') * 2 + s.count('5') * 5
if r == 64:
    print(n)
    break

```

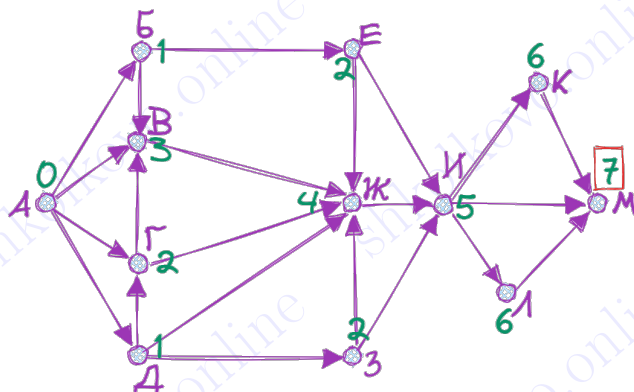
Ответ: 152

Задача 13

На рисунке представлена схема дорог, связывающих города А, Б, В, Г, Д, Е, Ж, З, И, К, Л, М. По каждой дороге можно двигаться только в одном направлении, указанном стрелкой. Какова длина самого длинного пути из города А в город М? Длиной пути считать количество дорог, составляющих этот путь.



Решение



Задача 14

Операнды арифметического выражения записаны в системе счисления с основанием 22

$$98x79641_{22} + 25x43_{22} + 63x5_{22}$$

В записи чисел переменной x обозначена неизвестная цифра алфавита 15-ричной системы счисления. Определите наименьшее значение x , при котором значение данного арифметического выражения кратно 21. Для найденного значения x вычислите частно от деления значения арифметического выражения на 21 и укажите его в ответе в десятичной системе счисления.

Основание системы счисления указывать не нужно.

Решение

```
for x in '0123456789ABCDE':
    if (int('98' + x + '79641', 22) + int('25' + x + '43', 22) +
        int('63' + x + '5', 22)) % 21 == 0:
        print((int('98' + x + '79641', 22) + int('25' + x + '43', 22) +
            int('63' + x + '5', 22)) // 21)
        break
```

Задача 15

Для какого наибольшего целого неотрицательного числа A выражение

$$(y + 3x > A) \vee (x < 20) \vee (y < 50)$$

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

Решение

```
def f(x, y, A):
    return (y + 3*x > A) or (x < 20) or (y < 50)

for a in range(1000):
    if all(f(x, y, a) == 1 for x in range(100) for y in range(100)):
```

```
print(a)
```

Задача 16

Задан алгоритм вычисления функции $F(n)$, где n - натуральное число:

$F(n) = 1$, при $n = 1$;

$F(n) = n - 2 + F(n - 1)$, если $n > 1$

Чему равно значение функции $F(2023) - F(2021)$?

Решение

```
from functools import lru_cache
```

```
@lru_cache(None)
```

```
def f(n):
```

```
    if n == 1:
```

```
        return 1
```

```
    elif n > 1:
```

```
        return n - 2 + f(n-1)
```

```
for i in range(1, 2023):
```

```
    f(i)
```

```
print(f(2023) - f(2021))
```

Задача 17

В файле содержится последовательность целых чисел. Элементы последовательности могут принимать целые значения от -1000000 до 1000000 включительно. Определите количество троек элементов в которых только одно число трехзначное, и сумма элементов тройки меньше максимального числа последовательности оканчивающегося на 17. В ответе запишите два числа: сначала количество найденных троек, а затем минимальную из сумм таких троек. В данной задаче под тройкой подразумевается три идущих подряд элемента последовательности.

Решение

```

f = open('17.txt')
a = list(map(int, f.readlines()))
m = max([c for c in a if str(c)[-2:] == '17'])
n = len(a)
k = 3
count = 0
ans = 10 ** 20
for i in range(n - (k - 1)):
    x = a[i : i + k]
    l = len([c for c in x if 99 < abs(c) < 1000])
    if l == 1 and sum(x) < m:
        count += 1
        ans = min(sum(x), ans)
print(count, ans)

```

Задача 18

Квадрат разлинован на $N \times N$ клеток ($1 < N < 17$). Исполнитель Робот может перемещаться по клеткам, выполняя за одно перемещение одну из двух команд: вправо или вниз. По команде вправо Робот перемещается в соседнюю правую клетку, по команде вниз — в соседнюю нижнюю. При попытке выхода за границу квадрата Робот разрушается. Между соседними клетками квадрата могут быть внутренние стены. Сквозь стену Робот пройти не может. Перед каждым запуском Робота в каждой клетке квадрата лежит монета достоинством от 1 до 100. Посетив клетку, Робот забирает монету с собой; это также относится к начальной и конечной клетке маршрута Робота.

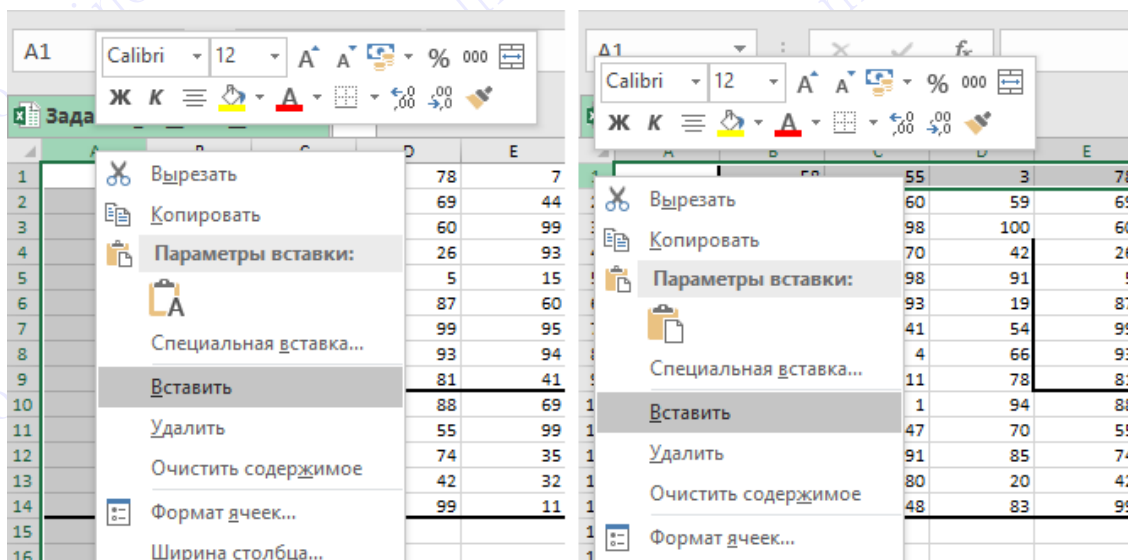
Откройте файл. Определите максимальную денежную сумму, которую может собрать Робот, начиная в верхнем левом углу. В ответ запишите одно число — максимальную сумму, которую может собрать Робот.

Исходные данные представляют собой электронную таблицу размером $N \times N$, каждая ячейка которой соответствует клетке квадрата.

Решение

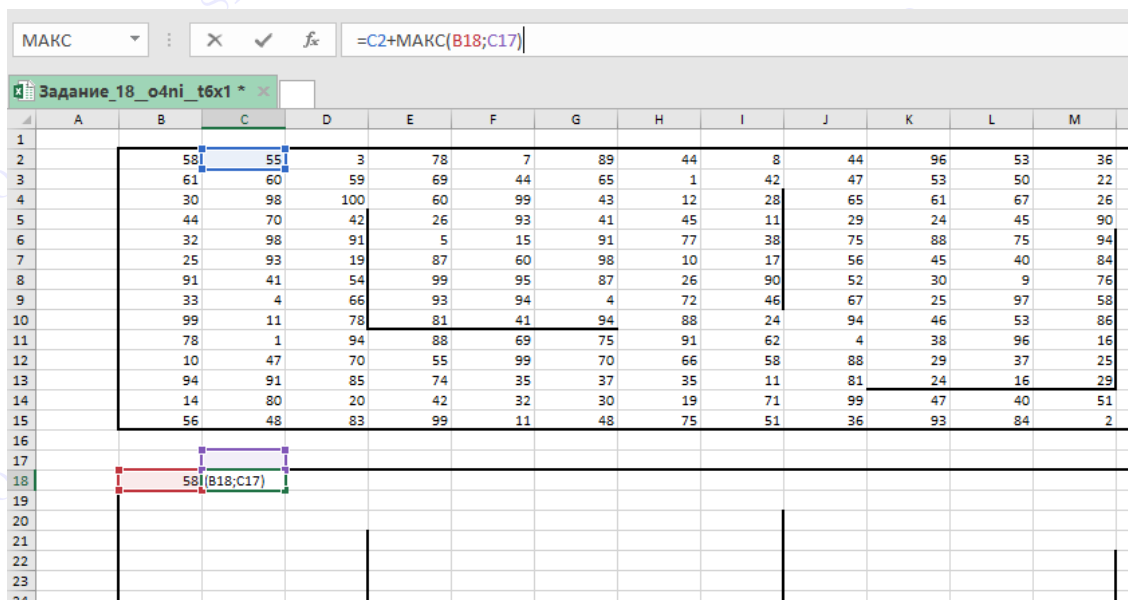
Добавляем пустой столбец (нажимаем правой кнопкой мыши на столбец A и вы-

бираем *Вставить*) и пустую строку перед самой первой (нажимаем правой кнопкой мыши на строку 1 и выбираем *Вставить*).



Копируем таблицу и с помощью специальной вставки ($Ctrl + Alt + V$) вставляем только её формат.

В начало маршрута (в нашем случае $B18$) записываем значение левой верхней клетки данной нам таблицы. В клетку $C18$ записываем формулу $=C2+МАКС(B18;C17)$



Раполним ей все оставшиеся ячейки таблицы.

Теперь изменим формулы там, где у нас находятся границы.

Если границы мешают нам пройти ниже, то меняем формулы в клетках, находящихся под границами, то есть убираем из них значение ячейки идущей выше границы.

ча камней. Игроки ходят по очереди, первый ход делает Петя. За один ход игрок может добавить в кучу один или пять камней, либо увеличить количество камней в куче в четыре раза. У каждого игрока есть неограниченное количество камней, чтобы делать ходы. Игра завершается в тот момент, когда количество камней в куче становится не менее 473.

Победителем считается игрок, сделавший последний ход, т.е. первым получивший кучу из 473 камней или больше.

В начальный момент в куче было S камней: $1 \leq S \leq 472$.

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.

Решение

```
import sys
sys.setrecursionlimit(100000)

from functools import lru_cache

def moves(h):
    return h + 1, h + 5, h * 4

@lru_cache(None)
def f(h):
    if (h >= 473):
        return 'END'
    if any(f(x) == 'END' for x in moves(h)):
        return 'P1'
    if all(f(x) == 'P1' for x in moves(h)):
        return 'V1'
    if any(f(x) == 'V1' for x in moves(h)):
        return 'P2'
    if all(f(x) == 'P1' or f(x) == 'P2' for x in moves(h)):
        return 'V2'
```

```
for s in range(1, 472):
    if f(s) == 'V1':
        print(s)
```

Задача 20

Для игры, описанной в предыдущем задании, найдите два таких значения S , при которых у Пети есть выигрышная стратегия, причём Петя не может выиграть за один ход и Петя может выиграть своим вторым ходом независимо от того, как будет ходить Ваня.

В ответе запишите числа в порядке возрастания без пробелов и знаков препинаний.

Решение

```
import sys
sys.setrecursionlimit(100000)

from functools import lru_cache

def moves(h):
    return h + 1, h + 5, h * 4

@lru_cache(None)
def f(h):
    if (h >= 473):
        return 'END'
    if any(f(x) == 'END' for x in moves(h)):
        return 'P1'
    if all(f(x) == 'P1' for x in moves(h)):
        return 'V1'
    if any(f(x) == 'V1' for x in moves(h)):
        return 'P2'
    if all(f(x) == 'P1' or f(x) == 'P2' for x in moves(h)):
        return 'V2'
```

```
for s in range(1, 472):
    if f(s) == 'P2':
        print(s)
```

Задача 21

Для игры, описанной ранее, найдите два таких значения S , при которых 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.

В ответе запишите числа в порядке возрастания без пробелов и знаков препинаний.

Решение

```
import sys
sys.setrecursionlimit(100000)

from functools import lru_cache

def moves(h):
    return h + 1, h + 5, h * 4

@lru_cache(None)
def f(h):
    if (h >= 473):
        return 'END'
    if any(f(x) == 'END' for x in moves(h)):
        return 'P1'
    if all(f(x) == 'P1' for x in moves(h)):
        return 'V1'
    if any(f(x) == 'V1' for x in moves(h)):
        return 'P2'
```

```

if all(f(x) == 'P1' or f(x) == 'P2' for x in moves(h)):
    return 'V2'

for s in range(1, 472):
    if f(s) == 'V2':
        print(s)

```

Задача 22

Стандартная, небольшой файл.

Задача 23

Исполнитель преобразует число, записанное на экране.

У исполнителя есть команды, которым присвоены номера:

1. Прибавить 1
2. Прибавить 3
3. Умножить на 3.

Первая команда увеличивает число на экране на 1, вторая — на 3, третья — увеличивает число в 3 раза.

Сколько существует программ, для которых при исходном числе 7 результатом является число 20 и при этом траектория содержит число 14 и не содержит 15?

Динамическое решение:

```

a = [0]*1000
a[7] = 1
for i in range(8, 20 + 1):
    a[i] = a[i - 1] + a[i - 3]
    if i % 3 == 0:
        a[i] += a[i // 3]
    if i == 15:
        a[i] = 0
    if i == 14:
        for j in range(14):
            a[j] = 0

```

```
print(a[20])
```

Рекурсивное решение:

```
def f(start, stop):
    if start > stop or start == 15:
        return 0
    if start == stop:
        return 1
    return f(start + 1, stop) + f(start + 3, stop) + f(start * 3, stop)

print(f(7, 14) * f(14, 20))
```

Задача 24

Текстовый файл состоит из символов, обозначающих прописные буквы латинского алфавита. Определите максимальное количество идущих подряд символов, которых никакие две буквы из набора букв *A*, *B* и *C* (с учетом повторений) не записаны подряд.

Для выполнения этого задания следует написать программу.

Решение

```
f = open()
s = f.readline()
count = 1
maxim = 1
a = ['AA', 'AB', 'AC', 'BA', 'BB', 'BC', 'CA', 'CB', 'CC']
for i in range(len(s) - 1):
    if (s[i] + s[i + 1]) not in a:
        count += 1
        maxim = max(maxim, count)
    else:
        count = 1
print(maxim)
```

Задача 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.

Найдите все натуральные числа, не превосходящие 10^9 , для которых выполнены все условия:

- соответствуют маске $*31 * 65?$;
- делятся на 31 и 2031 без остатка;
- количество делителей числа является результатом любой степени двойки.

В ответе запишите в первом столбце таблицы все найденные числа в порядке возрастания, справа от каждого числа их частное от деления на 2031.

Решение №1:

```
def countDivs(n):
    a = set()
    for i in range(1, int(n ** 0.5) + 1):
        if n % i == 0:
            a.add(i)
            a.add(n // i)
    return len(a)

def degreeOfTwo(n):
    return n & (n - 1) == 0

for num in range(31 * 2031, 10**9 + 1, 31 * 2031):
    s = str(num)
    flag = degreeOfTwo(countDivs(num))
    if flag and s.count("31") != 0 and s[-3:-1] == "65":
        print(num, num // 2031)
```


Решение №2:

```
def count_div(n):
    k = 0
    for j in range(1, int(n ** 0.5) + 1):
        if n % j == 0:
            k += 1
            if j != n // j:
                k += 1
    return k

def dvoika(n):
    m = n
    while m > 1:
        if m % 2 == 0:
            m //= 2
        else:
            return False
    return True

start = 0
for i in range(31650, 10 ** 9):
    if i % 31 == 0 and i % 2031 == 0 and str(i)[-3:-1] == '65' and \
        str(i).count('31') > 0:
        start = i
        break

for i in range(start, 10 ** 9 + 1, 31 * 2031):
    counter = count_div(i)
    if dvoika(counter) and str(i)[-3:-1] == '65' and \
        str(i).count('31') > 0:
        print(i, i // 2031)
```

Задача 26

Входной файл содержит информацию о плане проведения собраний в конференц-зале. Для каждого собрания известно время проведения и длительность собрания. Определите, сколько собраний будет проведено и в какую минуту завершиться последнее собрание. Если способов выбрать последнее собрание несколько, выбрать нужно то, длительность которого больше.

В первой строке входного файла находится натуральное число N , ($N \leq 1000$), обозначающее количество собраний. Каждая из следующих N строк содержит два натуральных числа: указанное в заявке время проведения (в минутах от начала суток, не превышает 1300) и длительность (в минутах, не превышает 1000) собрания.

Запишите в ответ два числа: количество собраний, которое можно провести и минуту за ? Типовой пример организации данных во входном файле

5

685 90

716 90

723 72

818 62

Похоже на задачу про юникс-время, но найти надо другое: не момент когда процессов много одновременно, а что-то (ждём точную формулировку), предположительно макс.количество конференций которые удастся провести, тогда жадник + сканлайн

Разбирали на этих вебах <https://2.shkolково.online/courses/5/1294/11171>,
<https://2.shkolково.online/courses/5/1455/11180>

Задача 27

В первой строке подаются два натуральных числа N — количество натуральных чисел в последовательности и K — минимальное расстояние, допустимое между любыми двумя элементами.

Требуется найти минимальное значение произведения тройки элементов так, что между любыми элементами тройки расстояние между двумя элементами не менее K (т.е. разность их индексов по модулю больше или равна K).

Решение

```
from random import randint
```

```
def neeff(a, k):
```

```
    n = len(a)
```

```
    pr = 10000000000000000
```

```
    for i in range(n):
```

```
        for j in range(i+k, n):
```

```
            for p in range(j+k, n):
```

```
                pr = min(pr, a[i]*a[j]*a[p])
```

```
    return pr
```

```
def eff(a, k):
```

```
    n = len(a)
```

```
    number = 10**15
```

```
    pair = 10**15
```

```
    min_pr = 10**15
```

```
    for i in range(2*k, n):
```

```
        # Мин. допустимое расстояние от самого правого
```

```
        # числа тройки до самого левого
```

```
        number = min(number, a[i - 2*k])
```

```
        # Мин. допустимое расстояние от второго
```

```
        # числа тройки до самого левого
```

```
        # (обычная пара с расстоянием)
```

```
        pair = min(pair, number * a[i - k])
```

```
        # Составили допустимую пару с расстоянием
```

```
        # К или более. Теперь считаем произведение пары
```

```
        # на третий элемент, т.е. считаем тройку
```

```
        min_pr = min(min_pr, pair * a[i])
```

```
    return min_pr
```

```
for j in range(100):  
    n = 200  
    k = randint(2, 30)  
    a = [randint(2, 10000) for i in range(n)]  
    if eff(a, k) != neeff(a, k):  
        print('ОШИБКА')  
        break
```