



Информатика

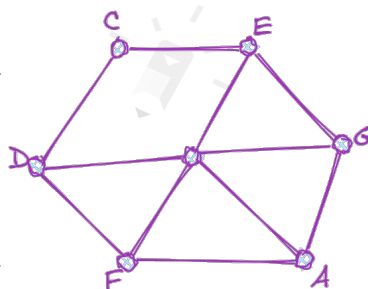
открытый курс

**Вариант основной волны ЕГЭ 2023. Решения и
ответы**

Задача 1.

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.

	П1	П2	П3	П4	П5	П6	П7
П1		*	*	*	*		*
П2	*		*	*			
П3	*	*					*
П4	*	*			*		
П5	*			*		*	
П6					*		*
П7	*		*			*	

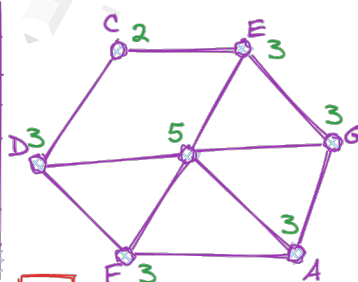


Ответ.

34

Решение.

	П1	П2	П3	П4	П5	П6	П7
B		*	*	*	*		*
A	*		*	*			
F/G	*	*					*
F/G	*	*			*		
E/D	*			*		*	
C					*		*
E/D	*		*			*	



34

Задача 2.

Алекс заполнил таблицу истинности логической функции F

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

Он успел заполнить лишь фрагмент из трех различных её строк, даже не указав, какому столбцу таблицы соответствует каждая переменная x, y, z, w .

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

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

Ответ.

yxwz

Решение.

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

Задача 3.

В файле приведён фрагмент базы данных «Продукты» о поставках товаров в магазины районов города. База данных состоит из трёх таблиц.

Таблица «Движение товаров» содержит записи о поставках товаров в магазины в течение первой декады июня 2021 г., а также информацию о проданных товарах. Поле Тип операции содержит значение Поступление или Продажа, а в соответствующее поле Количество упаковок, шт. занесена информация о том, сколько упаковок товара поступило в магазин или было продано в течение дня. Заголовок таблицы имеет следующий вид.

ID операции	Время	ID магазина	Артикул
Количество упаковок, шт.	Тип операции	Цена, руб./шт.	

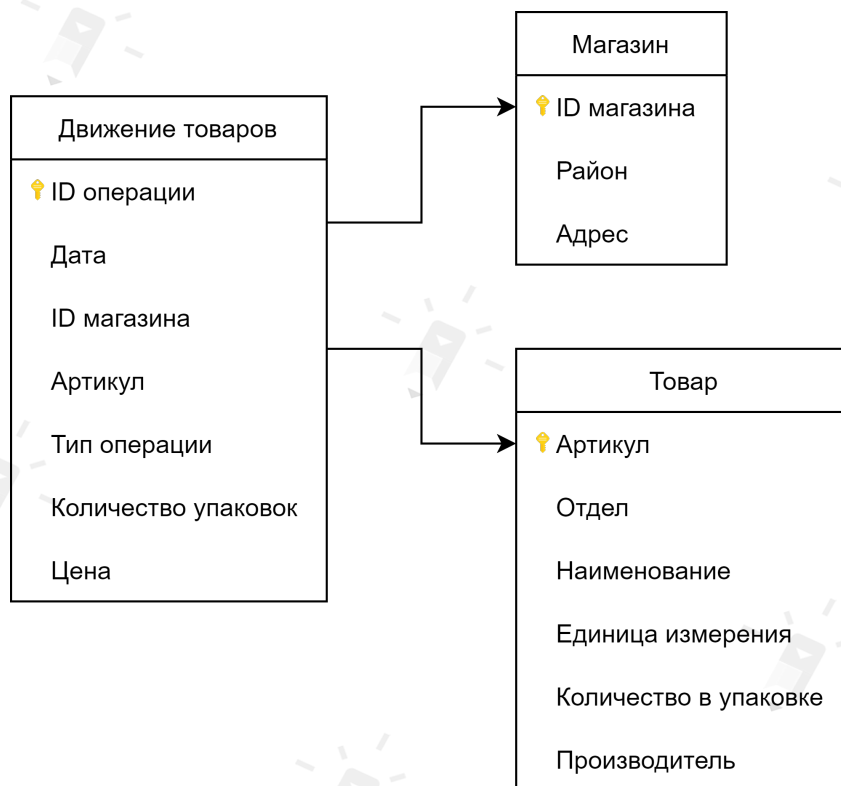
Таблица «Товар» содержит информацию об основных характеристиках каждого товара. Заголовок таблицы имеет следующий вид.

Артикул	Отдел	Наименование	Ед. изм.	Количество в упаковке	Поставщик
---------	-------	--------------	----------	-----------------------	-----------

Таблица «Магазин» содержит информацию о местонахождении магазинов. Заголовок таблицы имеет следующий вид.

ID магазина	Район	Адрес
-------------	-------	-------

На рисунке приведена схема указанной базы данных



Используя информацию из приведённой базы данных, определите сколько килограммов Кукурузных хлопьев было продано в магазинах Октябрьского района за период с 1 по 25 января включительно.

В ответе запишите только число.

Ответ.

120

Решение.

На листе *Товар* нажимаем на **Сортировка и фильтр => Фильтр**, в столбце *Наименование товара* оставляем только *Кукурузные хлопья с сахаром*, запоминаем полученный артикул. Переходим на лист *Движение товаров*, нажимаем на **Сортировка и фильтр => Фильтр**, в столбце *Артикул* оставляем только артикул, который запомнили.

На листе *Магазин* нажимаем на **Сортировка и фильтр => Фильтр**, в столбце *Район* оставляем только *Октябрьский*, запоминаем полученные ID магазинов. Переходим на лист *Движение товаров*, нажимаем на **Сортировка и фильтр => Фильтр**, в столбце *ID магазина* оставляем только ID, которые запомнили.

На листе *Движение товаров* в столбце *Время* выбираем только даты с 1 по 25 января (включительно), в столбце *Тип операции* оставляем только *Продажа*.

Выделяем все ячейки в столбце *Количество упаковок, шт.* и смотрим сумму в правом нижнем углу листа, выписываем полученное число, переходим на лист *Товар*, смотрим значение в столбце *Количество в упаковке*, умножаем сумму всех товаров на количество кг в упаковке.

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

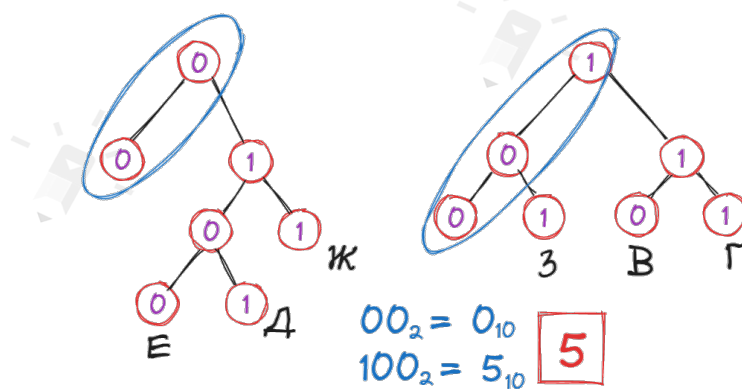
Какое наименьшее количество двоичных знаков потребуется для кодирования оставшихся букв? В ответе запишите суммарную длину кодовых слов для букв: А, Б

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

Ответ.

5

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.

Ответ.

8

Решение.

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

for n in range(1, 1000):
    x = per(n) #переводим в 3СС
    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 3СС 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 6 команд: Поднять хвост, означающая переход к перемещению без рисования; Опустить хвост, означающая переход в режим рисования; Вперёд n (где n – целое число), вызывающая передвижение Черепахи на n единиц в том направлении, куда указывает её голова; Назад n (где n – целое число), вызывающая передвижение в противоположном голове направлении; Направо m (где m – целое число), вызывающая изменение направления движения на m градусов по часовой стрелке, 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.

Черепахе был дан для исполнения следующий алгоритм:

Повтори 2 [Вперед 13 Направо 90 Вперед 20 Направо 90]

Поднять хвост

Вперед 8 Направо 90 Назад 3 Налево 90

Опустить хвост

Повтори 2 [Вперед 16 Направо 90 Вперед 8 Направо 90]

Определите, сколько точек с целочисленными координатами будут находиться внутри объединения фигур, ограниченного заданными алгоритмом линиями, включая точки на линиях.

Ответ.

411

Решение.

```
from turtle import *
tracer(0)
k = 16
for i in range(2):
    forward(13 * k)
    right(90)
    forward(20 * k)
    right(90)
pu()

forward(8 * k)
right(90)
backward(3 * k)
left(90)

pd()
for i in range(2):
    forward(16 * k)
    right(90)
    forward(8 * k)
    right(90)

up()
for x in range(-30 * k, 30 * k, k):
    for y in range(-30 * k, 30 * k, k):
        goto(x, y)
        dot(3, 'red')
done()
```

Считаем площадь двух фигур вместе с точками на линиях (для этого прибавляем 1 к каждой из сторон) и вычитаем пересечение:

$$17 \cdot 9 + 14 \cdot 21 - 6 \cdot 6 = 411$$



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.

Ответ.

1024

Решение.

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

Задача 8.

Все шестибуквенные слова, составленные из букв М, А, Н, Г, У, С, Т, записаны в алфавитном порядке и пронумерованы.

Вот начало начала списка:

- 1.АААААА
- 2.АААААГ
- 3.АААААМ
- 4.АААААН
- 5.АААААС
- 6.АААААТ
- 7.АААААУ

...

Под каким номером в списке стоит последнее слово, которое не начинается с буквы У, содержит только две буквы М и не более одной буквы Г.

Ответ.

100810

Решение.

```
from itertools import product
count = 0
m=[]
for p in product(sorted("МАНГУСТ"), repeat=6):
    count+=1
    if p.count("М") == 2 and p[0]!="У" and p.count("Г") <=1:
        m.append(count)
print(max(m))
```

Задача 9.

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

- в строке есть одно число, которое повторяется трижды, остальные четыре числа различны;
- среднее арифметическое неповторяющихся чисел строки не больше повторяющегося числа.

В ответе запишите только число.

Ответ.

36

Решение.

```
f = open('9.txt')
count = 0
for s in f:
    m = [int(x) for x in s.split()]
    pov = [x for x in m if m.count(x)==3]
    nepov = [x for x in m if m.count(x)==1]
    if len(pov) == 3 and len(nepov) == 4:
        if (sum(nepov)/4) <= (pov[0]):
            count += 1
print(count)
```

Задача 10.

С помощью текстового редактора определите, сколько раз встречается сочетание букв «дом» или «Дом» только в составе других слов, но не как отдельное слово, в тексте повести А.И.Куприна «Поединок». В ответе укажите только число.

Ответ.

206

Решение.

Воспользуемся поисковыми средствами текстового редактора. В строке введём «дом». Всего будет найдено 209 вхождений. Воспользуемся расширенным поиском. Выберем значение- Только слово целиком. Снова введем слово «дом». Получим три вхождения.

Всего 206 раз встречается сочетание букв «дом» только в составе других слов, но не как отдельное слово.

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

Ответ.

1568

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.

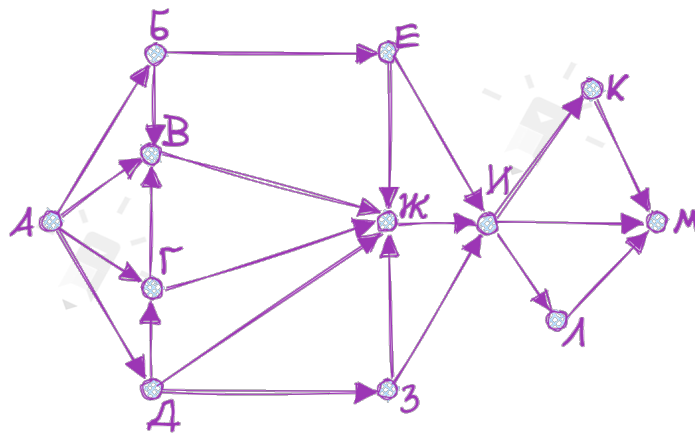
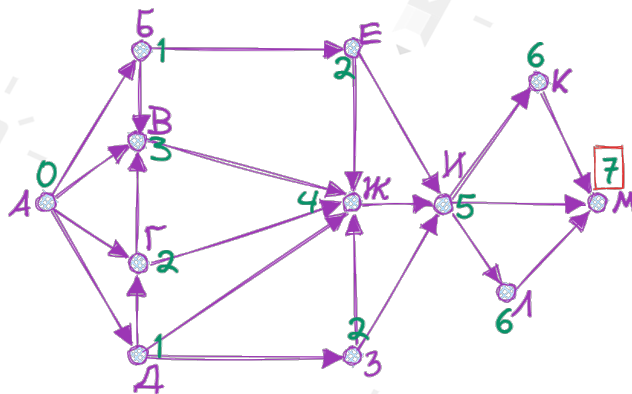
Ответ: 152

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
```

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.

Ответ.

1113295361

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 ?

Ответ.

109

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.

4041

```
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.

Элементы последовательности могут принимать целые значения от 1 до 100000 включительно. Определите количество троек последовательности, в которых только одно из чисел является четырёхзначным, а сумма элементов тройки не меньше максимального элемента последовательности, оканчивающегося на 15. В ответе запишите количество найденных троек, затем максимальную из сумм элементов таких троек. В данной задаче под тройкой подразумевается три идущих подряд элемента последовательности.

Ответ.

299 196183

Решение.

```
count = 0
m = 0
f = open('17.txt')
l = [int(i) for i in f]
max_dvy = 0
for i in range(len(l)):
    if l[i] % 100 == 15:
        max_dvy = max(max_dvy, l[i])
for i in range(len(l) - 2):
    c = 0
    s = [l[i], l[i+1], l[i+2]]
    for x in s:
        if 999 < x < 10000:
            c += 1
    if c == 1 and sum(s) >= max_dvy:
        m = max(m, (l[i] + l[i+1] + l[i+2]))
    count += 1
print(count, m)
```

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.

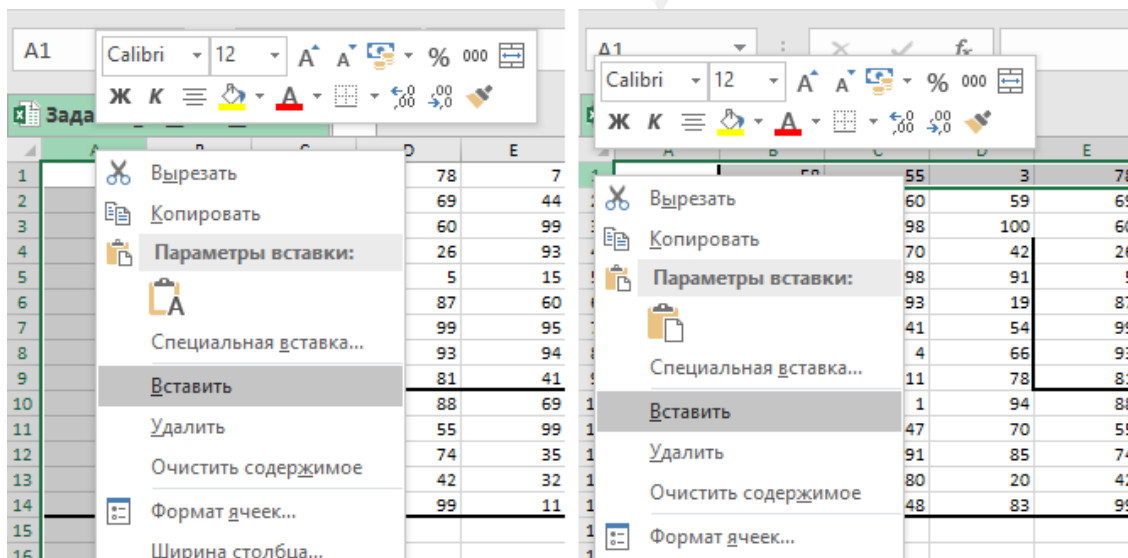
Исходные данные представляют собой электронную таблицу размером $N \times N$, каждая ячейка которой соответствует клетке квадрата.

Ответ.

1999

Решение.

Добавляем пустой столбец (нажимаем правой кнопкой мыши на столбец A и выбираем *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 $=C2+МАКС(B18;C17)$

МАКС $\times$ $\checkmark$ f_x $=C2+МАКС(B18;C17)$

Задание_18_o4ni_t6x1 * $\times$

	A	B	C	D	E	F	G	H	I	J	K	L	M
1													
2		58	55	3	78	7	89	44	8	44	96	53	36
3		61	60	59	69	44	65	1	42	47	53	50	22
4		30	98	100	60	99	43	12	28	65	61	67	26
5		44	70	42	26	93	41	45	11	29	24	45	90
6		32	98	91	5	15	91	77	38	75	88	75	94
7		25	93	19	87	60	98	10	17	56	45	40	84
8		91	41	54	99	95	87	26	90	52	30	9	76
9		33	4	66	93	94	4	72	46	67	25	97	58
10		99	11	78	81	41	94	88	24	94	46	53	86
11		78	1	94	88	69	75	91	62	4	38	96	16
12		10	47	70	55	99	70	66	58	88	29	37	25
13		94	91	85	74	35	37	35	11	81	24	16	29
14		14	80	20	42	32	30	19	71	99	47	40	51
15		56	48	83	99	11	48	75	51	36	93	84	2
16													
17													
18		58	$(B18;C17)$										
19													
20													
21													
22													
23													
24													

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.

МАКС $\times$ $\checkmark$ f_x $=E5+E20$

Задание_18_o4ni_t6x1 * $\times$

	B	C	D	E	F	G	H	I	J	K	L	M
8	91	41	54	99	95	87	26	90	52	30	9	
9	33	4	66	93	94	4	72	46	67	25	97	
10	99	11	78	81	41	94	88	24	94	46	53	
11	78	1	94	88	69	75	91	62	4	38	96	
12	10	47	70	55	99	70	66	58	88	29	37	
13	94	91	85	74	35	37	35	11	81	24	16	
14	14	80	20	42	32	30	19	71	99	47	40	
15	56	48	83	99	11	48	75	51	36	93	84	
16												
17												
18	58	113	116	194	201	290	334	342	386	482	535	
19	119	179	238	307	351	416	417	459	506	559	609	
20	149	277	377	437	536	579	591	619	571	632	699	
21	193	347	419	$=E5+E20$	629	670	715	726	600	656	744	
22	225	445	536	468	644	761	838	876	675	763	838	
23	250	538	557	555	704	859	869	893	731	808	878	
24	341	579	633	654	799	946	972	1062	783	838	887	
25	374	583	699	747	893	950	1044	1108	850	875	984	
26	473	594	777	828	934	1044	1132	1156	1250	1296	1349	
27	551	595	871	959	1028	1119	1223	1285	1289	1334	1445	
28	561	642	941	1014	1127	1197	1289	1347	1435	1464	1501	
29	655	746	1026	1100	1162	1234	1324	1358	1516	1540	1556	
30	669	826	1046	1142	1194	1264	1343	1429	1615	1662	1702	

Если граница мешает нам идти вправо, то убираем из текущей формулы значение ячейки, идущей после границы.

МАКС

✕

✓

fx

=K14+J30

Задание_18_o4ni_t6x1 *

	E	F	G	H	I	J	K	L	M	N	O	P
8	99	95	87	26	90	52	30	9	76	86	18	
9	93	94	4	72	46	67	25	97	58	58	10	
10	81	41	94	88	24	94	46	53	86	11	28	
11	88	69	75	91	62	4	38	96	16	90	24	
12	55	99	70	66	58	88	29	37	25	18	83	
13	74	35	37	35	11	81	24	16	29	4	88	
14	42	32	30	19	71	99	47	40	51	98	68	
15	99	11	48	75	51	36	93	84	2	14	80	
16												
17												
18	194	201	290	334	342	386	482	535	571	656	740	
19	307	351	416	417	459	506	559	609	631	699	801	
20	437	536	579	591	619	571	632	699	725	731	826	
21	463	629	670	715	726	600	656	744	834	860	953	
22	468	644	761	838	876	675	763	838	932	882	971	
23	555	704	859	869	893	731	808	878	1016	981	989	
24	654	799	946	972	1062	783	838	887	1092	1067	1085	
25	747	893	950	1044	1108	850	875	984	1150	1125	1135	
26	828	934	1044	1132	1156	1250	1296	1349	1435	1136	1164	
27	959	1028	1119	1223	1285	1289	1334	1445	1461	1226	1250	
28	1014	1127	1197	1289	1347	1435	1464	1501	1526	1244	1333	
29	1100	1162	1234	1324	1358	1516	1540	1556	1585	1248	1421	
30	1142	1194	1264	1343	1429	1615	=K14+J30	1702	1753	1851	1919	
31	1241	1252	1312	1418	1480	1651	1755	1839	1841	1865	1999	
32												
33												

Выписываем значение из ячейки O31 в ответ.

Задача 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 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.

Ответ.

118

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.

Ответ.

113117

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.

Ответ.

112116

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.

В файле содержится информация о совокупности N вычислительных процессов, которые могут выполняться параллельно или последовательно. Будем говорить, что процесс B зависит от процесса A , если для выполнения процесса B необходимы результаты выполнения процесса A . В этом случае процессы могут выполняться только последовательно. Информация о процессах представлена в файле в виде таблицы. В первом столбце таблицы указан идентификатор процесса (ID), во втором столбце таблицы – время его выполнения в миллисекундах, в третьем столбце перечислены с разделителем «;» ID процессов, от которых зависит данный процесс. Если процесс является независимым, то в таблице указано значение 0.

Определите минимальное время, через которое завершится выполнение всей совокупности процессов, при условии, что все независимые друг от друга процессы могут выполняться параллельно.

Типовой пример организации данных в файле:

ID процесса B	Время выполнения процесса B, мс	ID процесса(ов) A
1	4	0
2	3	0
3	1	1; 2
4	7	3

В данном случае независимые процессы 1 и 2 могут выполняться параллельно, при этом процесс 1 завершится через 4 мс, а процесс 2 – через 3 мс с момента старта. Процесс 3 может начаться только после завершения обоих процессов 1 и 2, то есть, через 4 мс после старта. Он длится 1 мс и закончится через $4 + 1 = 5$ мс после старта. Выполнение процесса 4 может начаться только после завершения процесса 3, то есть, через 5 мс. Он длится 7 мс, так что минимальное время завершения всех процессов равно $5 + 7 = 12$ мс.

Ответ.

301

Решение.

Для начала распределим значения из столбца С. Для этого выделим данные в этом столбце и с помощью кнопок «Данные»→«Текст по столбцам» расформируем их на разные столбцы.

В клетку A102(то есть в конце столбца А) в конце добавляем 0, чтобы функция ВПР не возвращала ошибку и корректно считала время независимых процессов.

В ячейку I2 поместим формулу =МАКС(F2:H2)+B2 и растянем её вниз. В ячейку F2 поместим формулу =ВПР(C2;\$A\$1:\$I\$102;9;0) и растянем её на диапазон F2:H101. В ячейку J2 запишем формулу: =МАКС(I2:I101). Это значение и будет являться ответом.

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?

Ответ.

18

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 A, B, C, D, E, F .

Определите в прилагаемом файле максимальное количество идущих подряд символов (длину непрерывной подпоследовательности), среди которых символ B встречается ровно 53 раза.

Для выполнения этого задания следует написать программу.

Ответ.

515

Решение.

```
f = open('24__3ns0t.txt')
s = f.read()

s = s.split('B')
mx = -1

for i in range(len(s) - 53):
    new_s = ".".join(s[i:i+54])
    mx = max(len(new_s) + 53, mx)

print(mx)
```

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 .

Ответ.

```
53831655 26505
333126651 164021
512313657 252247
647931651 319021
831966654 409634
```

Решение.

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)

```

Решение №3:

```

from fnmatch import *

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)

for x in range(31 * 2031, 10**9 + 1, 31 * 2031):
    n = countDivs(x)
    if fnmatch(str(x), "**31*65?") and n & (n - 1) == 0:
        print(x, x // 2031)

```


Задача 26.

Входной файл содержит сведения о заявках на проведение занятий в конференц-зале. В каждой заявке указаны время начала и время окончания мероприятия (в минутах от начала суток). Если время начала одного мероприятия меньше времени окончания другого, то провести можно только одно из них. Если время окончания одного мероприятия совпадает с временем начала другого, то провести можно оба.

Определите максимальное количество мероприятий, которое можно провести в конференц-зале и самое позднее время окончания последнего мероприятия.

Входные данные

В первой строке входного файла находится натуральное число N ($N < 1000$) — количество заявок на проведение мероприятий.

Следующие N строк содержат пары чисел, обозначающих время начала и время окончания мероприятий. Каждое из чисел натуральное, не превосходящее 1440.

Запишите в ответе два числа: максимальное — количество мероприятий, которое можно провести в конференц-зале и самое позднее время окончания последнего мероприятия (в минутах от начала суток).

Типовой пример организации данных во входном файле

```
5
10 150
100 110
131 170
131 180
120 130
```

При таких исходных данных можно провести максимум три мероприятия, например, по заявкам 2, 3 и 5.

Конференц-зал освободится самое позднее на 180-й минуте, если состоится мероприятия по заявкам 2, 4, 5. Типовой пример имеет иллюстративный — характер. Для выполнения задания используйте данные из прилагаемых файлов.

Ответ.

16 1345

Решение.

```
f = open('26.txt')

n = int(f.readline())
data = []
for i in range(n):
    start, end = map(int, f.readline().split())
    data.append([start, end])

data.sort( key = lambda x: [x[1], x[0]] )

allActions = [data[0][1]]
for meet in data:
    if meet[0] >= allActions[-1]:
        allActions.append(meet[1])

for meet in data:
    if meet[0] >= allActions[-2]:
        allActions[-1] = max(meet[1], allActions[-1])

print(len(allActions), allActions[-1])
```

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).

Ответ.

169805608193110 127566709650

Решение.

```
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
        # K или более. Теперь считаем произведение пары
        # на третий элемент, т.е. считаем тройку
        min_pr = min(min_pr, pair * a[i])

    return min_pr

f = open('27_B.txt')

n, k = map(int, f.readline().split())
a = [int(f.readline()) for i in range(n)]

print(eff(a, k))
print(neeff(a, k))
```

Подписывайся на наши социальные сети:

