

ДЕКАБРЬ		РАСПИСАНИЕ 2025					×
пн	вт	ср	чт	пт	сб	вс	
01	02 Задание 22 (без таймлайна)	03	04 Задание 22 таймлайн	05	06	07 Финиш вариант II	
08	09 Задание 12 I	10	11 Задание 12 II	12	13	14 27 задание	
15	16 Задание 6 I	17	18 Задание 6 II	19	20	21 25 задание	
22	23 Задание 5 I	24	25 Задание 5 II	26	27	28	
29	30	31 	● Курс «Первая часть»				

Годовой курс 25/26

2-3 вебинара в неделю

Домашки и дополнительная практика

Пробники (1-2 раза в месяц)

Конспекты прошедших вебов

Чат для вопросов

+ Годовой курс 24/25 и прочие материалы

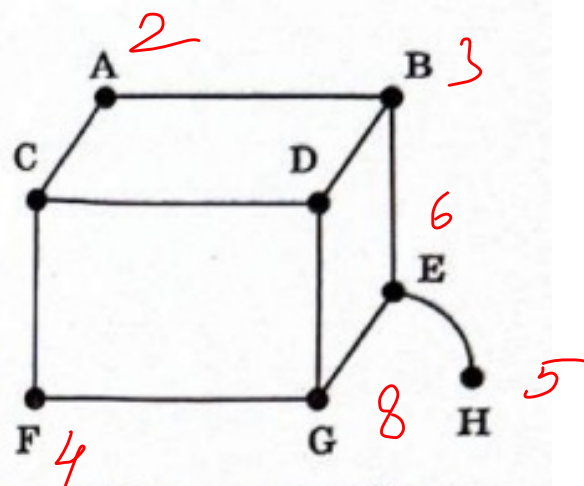
+ Интенсив

**Стоимость: 10000 рублей за весь курс**

1

На рисунке схема дорог  $N$ -ского района изображена в виде графа, в таблице содержатся сведения о протяжённости каждой из этих дорог (в километрах).

		Номер пункта							
		1	2	3	4	5	6	7	8
Номер пункта	1			4				7	9
	2			11				5	
	3	4	11				22		
	4						12	10	
	5						13		
	6			22		13			15
	7	7	5		12				
	8	9			10		15		



Так как таблицу и схему рисовали независимо друг от друга, нумерация населённых пунктов в таблице никак не связана с буквенными обозначениями на графе. Определите, какова сумма протяжённости дорог из пункта  $A$  в пункт  $B$  и из пункта  $F$  в пункт  $G$ .

В ответе запишите целое число.

$$11 + 10 = 21$$

2

Миша заполнял таблицу истинности логической функции  $F$

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

но успел заполнить лишь фрагмент из трёх различных её строк, даже не указав, какому столбцу таблицы соответствует каждая из переменных  $w, x, y, z$ .

	$z$	$w$	$y$	$x$	$F$
	1	0		0	1
0	<del>1</del>	<del>1</del>	0	0	1
	1		1	0	1

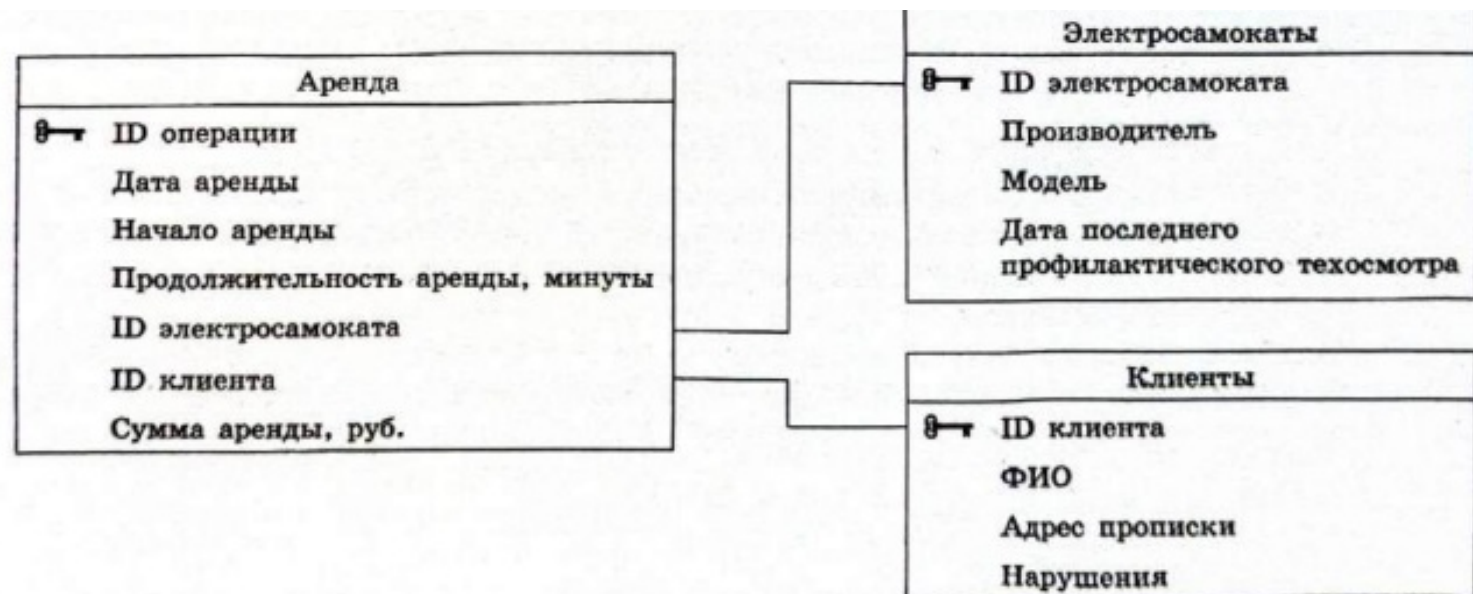
Определите, какому столбцу таблицы соответствует каждая из переменных  $w, x, y, z$ . В ответе напишите буквы  $w, x, y, z$  в том порядке, в котором идут соответствующие им столбцы (сначала буква, соответствующая первому столбцу; затем буква, соответствующая второму столбцу, и т. д.). Буквы в ответе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

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

02.py - C:/Users/axelo/OneDrive/Рабочий стол/02.py (3.12.5)

File Edit Format Run Options Window Help

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



Используя информацию из приведённой базы данных, определите общее время аренды (в минутах) электросамокатов, произведённых фирмой TRIBE, клиентом Сачковым Х. Э. за период с 10.06.2025 по 15.06.2025. В ответе запишите только число.

A	B	C	D	E	F	G
ID операции	Дата аренды	Начало аренды	Продолжительность аренды, минуты	ID электросамоката	ID клиента	Сумма аренды руб.
560	10.06.25	14:32:37	4	A52	13	91
711	12.06.25	21:22:36	32	A52	13	277
754	13.06.25	18:11:44	49	A52	13	395
770	13.06.25	21:31:43	7	A52	13	119
782	14.06.25	7:11:28	43	A52	13	427
			135,00			

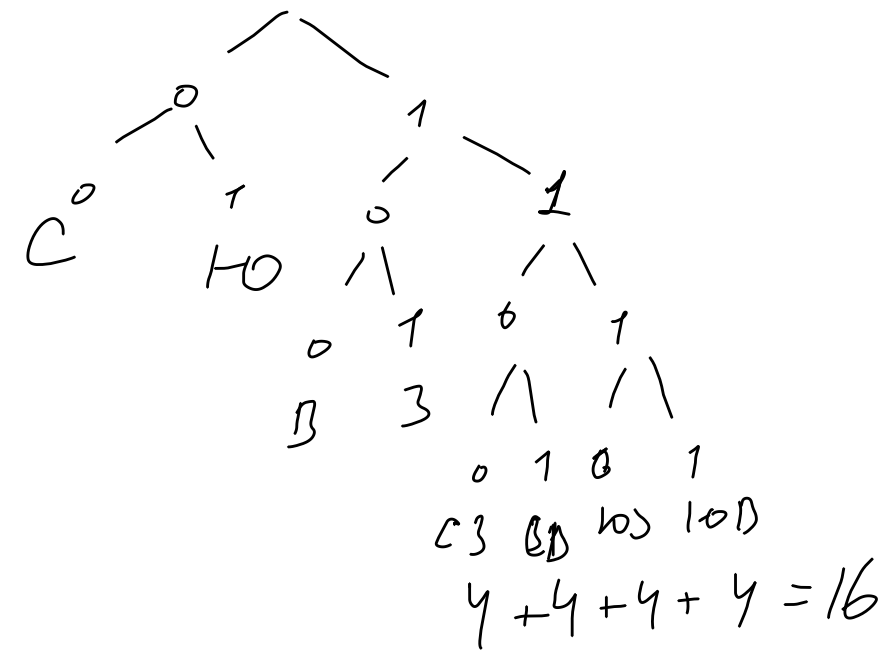
4

Для кодирования восьми сторон горизонта (север, юг, восток, запад и четыре промежуточные стороны) применили неравномерный двоичный код, удовлетворяющий условию Фано. Для кодирования используют кодовые слова.

Сторона горизонта	Кодовое слово
Север	00
Юг	01
Восток	100
Запад	101

Какое наименьшее количество двоичных знаков требуется для кодирования четырёх оставшихся промежуточных сторон горизонта? В ответе запишите суммарную длину кодовых слов для кодирования сторон горизонта: юго-восток, юго-запад, северо-восток, северо-запад.

*Примечание.* Условие Фано означает, что никакое кодовое слово не является началом другого кодового слова. Это обеспечивает возможность однозначной расшифровки закодированных сообщений.



5

Автомат получает на вход четырёхзначное число  $K$ . По этому числу строится новое число  $L$  по следующим правилам.

1. Вычисляется число  $S$  — сумма всех цифр числа  $K$ .
2. Вычисляется число  $M$ , равное значению максимальной цифры в записи числа  $K$ .
3. Вычисляется число  $N$ , равное значению минимальной цифры в записи числа  $K$ .
4. Вычисляется число  $P_1$ :  $P_1 = S - M$ .
5. Вычисляется число  $P_2$ :  $P_2 = S - N$ .
6. Новое число  $L$  получается в результате записи рядом без пробелов чисел  $P_1$  и  $P_2$  таким образом, чтобы они следовали слева направо в неубывающем порядке.

Пример. Исходное число  $K = 6241$ .

Сумма его цифр  $S = 6 + 2 + 4 + 1 = 13$ .

$M = 6$ .

$N = 1$ .

$P_1 = 13 - 6 = 7$ .

$P_2 = 13 - 1 = 12$ .

Результат:  $L = 712$ .

Укажите наименьшее число  $K$ , при обработке которого автомат выдаст число  $L = 1318$ .

↑ возраст таковы

1666

05.py - C:/Users/axelo/OneDrive/Рабочий стол/05.py (3.12.5)

File Edit Format Run Options Window Help

```
for k in range(1000,10000):
    d = [int(c) for c in str(k)]
    s = sum(d)
    m = max(d)
    n = min(d)
    p1 = s-m
    p2 = s-n
    if p1<p2: l = int(str(p1)+str(p2))
    else: l = int(str(p2)+str(p1))
    if l==1318:
        print(k)
```

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

Черепашке был дан для исполнения следующий алгоритм:

Повтори 2 [Вперёд 9 Направо 90 Вперёд 5 Направо 270]

Назад 18 Налево 90 Вперёд 10 Направо 90

Поднять хвост

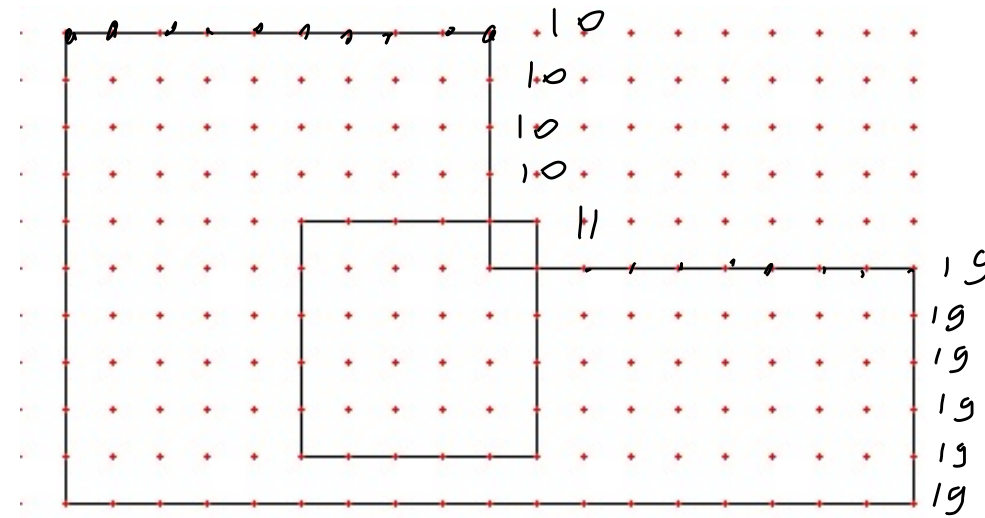
Вперёд 5 Направо 90 Вперёд 4 Налево 90

Опустить хвост

Повтори 4 [Вперёд 5 Направо 90]

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

```
from turtle import *  
  
tracer(0)  
screensize(5000,5000)  
r = 20  
  
for i in range(2):  
    fd(9*r); rt(90); fd(5*r); rt(270)  
bk(18*r); lt(90); fd(10*r); rt(90)  
up()  
fd(5*r); rt(90); fd(4*r); lt(90)  
down()  
for i in range(4):  
    fd(5*r); rt(90)  
  
up()  
for x in range(-50,50):  
    for y in range(-50,50):  
        goto(x*r,y*r)  
        dot(3,'red')  
  
update()
```



$$40 + 11 + 6 \cdot 19 = 165$$

7

Путешественник фотографирует достопримечательности с помощью камеры смартфона. Каждая фотография представляет собой растровое изображение размером  $2048 \times 1000$  пикселей, при этом используется палитра из  $2^{28}$  цветов. В конце дня путешественник отправляет снимки родственникам с помощью приложения-мессенджера. Для экономии трафика приложение оцифровывает снимки повторно, используя размер  $1920 \times 512$  пикселей и глубину цвета 16 бит. Сколько Кбайт трафика экономится при передаче 30 фотографий?

В ответе укажите целую часть полученного числа.

$$30 * (2048 * 1000 * 28 - 1920 * 512 * 16) / 8 * 1024 = 152400.0$$

8 Все шестибуквенные слова, составленные из букв М, О, С, К, В, А, записаны в алфавитном порядке и пронумерованы.

Вот начало списка:

1. АААААА
2. АААААВ
3. АААААК
4. АААААМ
5. АААААО
6. АААААС

...

Определите, под каким номером в этом списке стоит первое слово с чётным номером, которое не начинается с букв А, В и К и при этом содержит в своей записи ровно две буквы К, не идущие подряд.

*Примечание.* Слово — последовательность идущих подряд букв, не обязательно осмысленная.

23774 МАКАКВ

08.py - C:/Users/axelo/OneDrive/Рабочий стол/08.py (3.12.5)

File Edit Format Run Options Window Help

```
from itertools import *
```

```
k = 0
```

```
for x in product(sorted('МОСКВА'), repeat=6):
```

```
    s = ''.join(x)
```

```
    k += 1
```

```
    if k%2==0 and s[0] not in 'ABK' and s.count('K')==2 and 'KK' not in s:
```

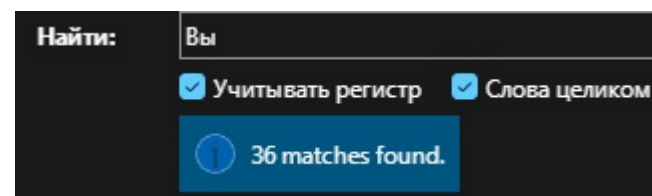
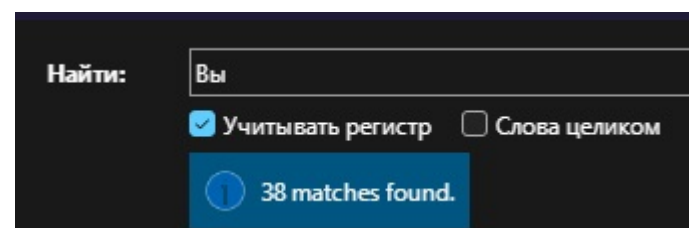
```
        print(k,s)
```

```
        break
```



10

С помощью текстового редактора определите, сколько раз встречается сочетание букв «Вы» с заглавной буквы в составе других слов, но не как отдельное слово в тексте глав V, VI и VII романа И. С. Тургенева «Рудин». В ответе укажите только число.



$38 - 36 = 2$  части других слов

**11**

Каждому изготовленному на предприятии двигателю присваивается уникальный серийный номер, состоящий из 23 символов. В базе данных каждый серийный номер занимает одинаковое и минимально возможное число байт. При этом используется посимвольное кодирование серийных номеров, все символы кодируются одинаковым и минимально возможным числом бит. Известно, что для хранения 3 222 444 серийных номеров потребовалось не менее 45 Мбайт памяти. Определите минимально возможную мощность алфавита, используемого для записи серийных номеров. В ответе запишите только целое число.

17

11.py - C:/Users/axelo/OneDrive/Рабочий стол/11.py (3.12.5)

File Edit Format Run Options Window Help

```
from math import *
```

```
for kod in range(1,1000):  
    bit = ceil(log2(kod))  
    byte = ceil(23*bit/8)  
    if 3222444*byte >= 45*1024*1024:  
        print(kod)  
        break
```

Выполните задание.

На ленте в соседних ячейках записана последовательность из 520 символов, среди которых могут быть только нули и единицы. Ячейки справа и слева от последовательности заполнены пустыми символами «λ». В начальный момент времени головка расположена в ближайшей справа к последовательности ячейке.

Программа работы исполнителя:

	λ	1	0
$q_0$	λ, L, $q_1$		
$q_1$	λ, S, $q_1$	0, S, $q_1$	1, L, $q_1$

После выполнения программы на ленте осталось ровно 125 нулей. Определите минимально возможное число нулей в исходной последовательности.

0 1 0 ~~0~~ ~~0~~ ~~0~~ ~~0~~

0 1 1 0 0 0 0 0 → 0 1 0 1 1 1 1 1

0 1 1 0 1 → 0 1 1 0 0 ←

мин число нулей <sup>125 нулей</sup> если их справа не было

минимум было 124 нуля

**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 его маске.

Широковещательным адресом называется специализированный адрес, в котором на месте нулей в маске стоят единицы. Адрес сети и широковещательный адрес не могут быть использованы для адресации сетевых устройств.

Сеть задана IP-адресом одного из входящих в неё узлов 89.16.43.107 и сетевой маской 255.224.0.0.

Найдите в данной сети наибольший IP-адрес, который может быть назначен компьютеру. В ответе укажите найденный IP-адрес без разделителей.

*Например*, если бы найденный адрес был равен 131.16.5.88, то в ответе следовало бы записать: 13116588.

```
> from ipaddress import *
> net = ip_network('89.16.43.107/255.224.0.0',0)
> net[-2]
IPv4Address('89.31.255.254')
```

14

Значение арифметического выражения

$$5 \cdot 512^{1000} + 256^{1001} - 128^{1002} + 64^{1003} - 7 \cdot 32^{1004} - 5120$$

записали в системе счисления с основанием 32. Определите в 32-ричной записи числа количество цифр с числовым значением, не превышающим 9.

401

14.py - C:/Users/axelo/OneDrive/Рабочий стол/14.py (3.12.5)

File Edit Format Run Options Window Help

```
a = 5*512**1000 + 256**1001 - 128**1002 + 64**1003 - 7*32**1004 - 5120
k = 0
while a>0:
    if a%32<=9:
        k += 1
    a = a//32
print(k)
```

**15**

Для какого наименьшего целого неотрицательного числа  $A$  логическое выражение

$$(105 \neq y + 2x) \vee (A > x) \vee (A > y)$$

истинно (т. е. принимает значение 1) при любых целых неотрицательных  $x$  и  $y$ ?

36

15.py - C:/Users/axelo/OneDrive/Рабочий стол/15.py (3.12.5)

File Edit Format Run Options Window Help

```
def f(x,y):  
    return (105!=y+2*x) or (a>x) or (a>y)  
  
for a in range(1000):  
    if all(f(x,y)==1 for x in range(1000) for y in range(1000)):  
        print(a)
```

**16**

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

$F(n) = n$ , если  $n < 10$ ;

$F(n) = n^3 + F(n - 15)$ , если  $n \geq 10$ .

Чему равно значение выражения  $F(1000) - F(940)$ ?

3739328500

16.py - C:/Users/axelo/OneDrive/Рабочий стол/16.py (3.12.5)

File Edit Format Run Options Window Help

```
def f(n):  
    if n < 10: return n  
    return n**3 + f(n-15)
```

```
print(f(1000)-f(940))
```

17

В файле содержится последовательность целых чисел. Её элементы могут принимать целые значения от  $-100\,000$  до  $100\,000$  включительно. Определите количество троек элементов последовательности, в которых нет чётных чисел, а сумма элементов тройки больше минимального элемента последовательности, оканчивающегося на 500. В ответе запишите количество найденных троек чисел, затем минимальную из сумм элементов таких троек. В данной задаче под тройкой подразумевается три идущих подряд элемента последовательности.

825 -95103

17.py - C:/Users/axelo/OneDrive/Рабочий стол/17.py (3.12.5)

File Edit Format Run Options Window Help

```
a = [int(x) for x in open('17.txt')]

m = min(x for x in a if abs(x)%1000==500)

ans = []

for x,y,z in zip(a,a[1:],a[2:]):
    if x%2!=0 and y%2!=0 and z%2!=0 and x+y+z>m:
        ans.append(x+y+z)
print(len(ans), min(ans))
```



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 3 камня;
- убрать из кучи 5 камней;
- уменьшить количество камней в куче в 4 раза (количество камней, полученное при делении, округляется до меньшего).

Например, из кучи в 40 камней за один ход можно получить кучу из 37, 35 или 10 камней. Игра завершается, когда количество камней в куче становится не более 71. Победителем считается игрок, сделавший последний ход, то есть первым получивший кучу из 71 или менее камней. В начальный момент в куче было  $S$  камней,  $S \geq 72$ .

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.

20

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

Ответ:

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.

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

```
19 [288, 289, 290]
20 [291, 292, 293, 294, 295]
21 [296, 297, 298]
```

19.py - C:/Users/axelo/OneDrive/Рабочий стол/19.py (3.12.5)

File Edit Format Run Options Window Help

```
def f(s,m):
    if s<=71: return m%2==0
    if m==0: return 0
    h = [f(s-3,m-1), f(s-5,m-1), f(s//4,m-1)]
    return any(h) if (m-1)%2==0 else all(h)

print(19,[s for s in range(72,1000) if f(s,2)])
print(20,[s for s in range(72,1000) if not f(s,1) and f(s,3)])
print(21,[s for s in range(72,1000) if not f(s,2) and f(s,4)])
```

22

В файле содержится информация о совокупности  $N$  вычислительных процессов, которые могут выполняться параллельно или последовательно. Приостановка выполнения процесса не допускается. 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 (в мс), за которое завершатся 12 процессов. Считать, что каждый процесс начинается в самое раннее допустимое время. Минимальное время отсчитывается непрерывно с первой миллисекунды. В ответе укажите только число — количество мс.

A	B	C	D	E	F	G	H	I	J
0									0
18	2	0			0	0	0		2
15	3	0			0	0	0		3
5	2	15			3	0	0		5
8	6	0			0	0	0		6
7	1	8			6	0	0		7
4	4	5			5	0	0		9
6	2	7	18		7	2	0		9
11	3	6			9	0	0		12
12	3	4	5		9	5	0		12
14	7	5			5	0	0		12
2	6	6			9	0	0		15
16	3	11			12	0	0		15
10	4	11			12	0	0		16
9	5	11			12	0	0		17
13	5	4	14		0	12	0		17

15

**23**

Исполнитель преобразует число на экране.

У исполнителя есть три команды, которые обозначены латинскими буквами:

**A. Вычесть 1**

**B. Вычесть 4**

**C. Найти целую часть от деления на 2**

Программа для исполнителя — это последовательность команд.

Сколько существует программ, для которых при исходном числе 30 результатом является число 4, при этом траектория вычислений не содержит числа 8 и содержит 12?

Траектория вычислений программы — это последовательность результатов выполнения всех команд программы.

*Например,* для программы **СAB** при исходном числе 13 траектория состоит из чисел 6, 5, 1.

**1358**

23.py - C:/Users/axelo/OneDrive/Рабочий стол/23.py (3.12.5)

File Edit Format Run Options Window Help

```
def f(c,e):
    if c<e or c==8: return 0
    if c==e: return 1
    return f(c-1,e)+f(c-4,e)+f(c//2,e)
print(f(30,12)*f(12,4))
```

**24**

Текстовый файл состоит из десятичных цифр и заглавных букв латинского алфавита. Определите в прилагаемом файле максимальное количество идущих подряд символов, оканчивающихся подстрокой CAT, среди которых цифра 1 содержится ровно 700 раз, а подстрока CAT встречается не менее 4 раз. В ответе запишите число — количество символов в найденной последовательности. Для выполнения этого задания следует написать программу.

25356

24.py - C:/Users/axelo/OneDrive/Рабочий стол/24.py (3.12.5)

File Edit Format Run Options Window Help

```
s = open('24.txt').readline()

m = 0
l = 0
k1 = 0
kcat = 0

for r in range(len(s)):
    if s[r]=='1': k1+=1
    if r>=2 and s[r-2]+s[r-1]+s[r] == 'CAT':
        kcat += 1
        while k1>700:
            if s[l]=='1': k1-=1
            if s[l]+s[l+1]+s[l+2]=='CAT': kcat-=1
            l += 1
        if k1==700 and kcat>=4:
            m = max(m, r-l+1)

print(m)
```

25

Напишите программу, которая перебирает целые числа, большие 3 000 000, в порядке возрастания и ищет среди них числа, представимые в виде произведения ровно двух простых множителей, не обязательно различных, каждый из которых содержит в своей записи ~~цифру 1~~ цифру 1, ~~цифру 3~~ цифру 3. В ответе в первом столбце таблицы запишите первые 5 найденных чисел в порядке возрастания, а во втором столбце — для каждого числа наибольший из соответствующих им найденных множителей. Количество строк в таблице для ответа избыточно.

*разложить на простые множители*

```
= RESTART: C:/Users/...
3000001 3517
3000009 1000003
3000013 3881
3000023 230771
3000031 22901

def fact(n):
    div = []
    i = 2
    while i**2 <= n:
        while n%i == 0:
            div.append(i)
            n = n//i
        i += 1
    if n > 1:
        div.append(n)
    return div

k = 0
for x in range(3_000_001, 3_100_000):
    if k == 5: break
    d = fact(x)
    if len(d) == 2 and ('1' in str(d[0]) or '3' in str(d[0])) and \
        ('1' in str(d[1]) or '3' in str(d[1])):
        print(x, max(d))
        k += 1
```

Организаторам спортивных соревнований необходимо доставить как можно большее число команд в некоторый город. Для доставки используются самолёты определённой пассажироместимости. Входной файл содержит сведения о количестве человек в команде и о пассажироместимости самолётов, которые имеются в наличии.

Из соображений безопасности в одном самолёте может лететь только одна команда, а количество занятых спортсменами мест должно быть не больше, чем количество свободных мест (т. е. пассажироместимость используемого для транспортировки команды самолёта должна быть не менее чем вдвое больше количества человек в команде).

Найдите способ доставить на соревнования максимально возможное число команд. Если способов несколько, то нужно выбрать такой, чтобы можно было доставить команду с максимальным числом участников.

*Входные данные.*

В первой строке входного файла находятся два натуральных числа  $N$  ( $N \leq 1000$ ) и  $M$  ( $M \leq 1000$ ) — количество команд и количество самолётов соответственно. Следующие  $N$  строк содержат числа, обозначающие количество человек в команде, затем идут  $M$  строк, где указана пассажироместимость самолётов. Числа  $M$  и  $N$  могут быть не равны.

Запишите в ответе два натуральных числа: сначала максимальное количество команд, которые могут прибыть на соревнования, затем максимальную численность команды в этом случае.

398 96867

26.py - C:/Users/axelo/OneDrive/Рабочий стол/26.py (3.12.5)

File Edit Format Run Options Window Help

```
f = open('26var03.txt')
```

```
N, M = [int(x) for x in f.readline().split()]
```

```
human = [int(f.readline()) for i in range(N)]
```

```
air = [int(f.readline()) for i in range(M)]
```

```
human.sort(reverse=1)
```

```
air.sort(reverse=1)
```

```
ans = []
```

```
for x in human:
```

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

```
        if air[i] >= 2*x:
```

```
            ans.append(x)
```

```
            air[i] = 0
```

```
            break
```

```
print(len(ans), max(ans))
```

```
ans = []
for x in human:
    for y in air:
        if y >= 2*x:
            ans.append(x)
            air.remove(y)
            break
```

Фрагмент звёздного неба спроецирован на плоскость с декартовой системой координат. Учёный решил провести кластеризацию полученных точек, являющихся изображениями звёзд, то есть разбить их множество на  $N$  непересекающихся непустых подмножеств (кластеров), таких, что точки каждого подмножества лежат внутри прямоугольника со сторонами длиной  $H$  и  $W$ , причём эти прямоугольники между собой не пересекаются. Стороны прямоугольников не обязательно параллельны координатным осям. Гарантируется, что такое разбиение существует и единственно для заданных размеров прямоугольников.

Будем называть центром кластера точку этого кластера, сумма расстояний от которой до всех остальных точек кластера минимальна. Для каждого кластера гарантируется единственность его центра. Расстояние между двумя точками на плоскости  $A(x_1, y_1)$  и  $B(x_2, y_2)$  вычисляется по формуле:

$$d(A, B) = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}.$$

В файле А хранятся координаты точек двух кластеров, где  $H = 6,5$  и  $W = 4,5$  для каждого кластера. В каждой строке записана информация о расположении на карте одной звезды: сначала координата  $x$ , затем координата  $y$ . Известно, что количество точек не превышает 1000.

В файле Б хранятся координаты точек трёх кластеров, где  $H = 4,5$ ,  $W = 4$  для каждого кластера. Известно, что количество точек не превышает 10 000. Структура хранения информации в файле Б аналогична структуре в файле А.

Известно, что в файле А имеются координаты ровно двух, а в файле Б ровно трёх «лишних» точек, представляющих аномалии, которые возникли в результате помех при передаче данных. Эти точки не относятся ни к одному из кластеров, их учитывать не нужно.

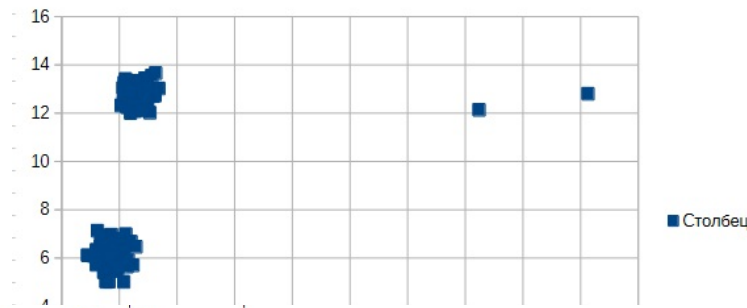
Для файла А определите координаты центра каждого кластера, затем найдите два числа:  $P_x$  — расстояние по оси абсцисс между центрами кластеров и  $P_y$  — расстояние по оси ординат между центрами кластеров.

Для файла Б определите координаты центра каждого кластера, затем найдите два числа:  $Q_1$  — минимальное расстояние между центрами кластеров и  $Q_2$  — максимальное расстояние между центрами кластеров.

В ответе запишите четыре числа: в первой строке — сначала целую часть произведения  $P_x \times 10\,000$ , затем целую часть произведения  $P_y \times 10\,000$ ; во второй строке — сначала целую часть произведения  $Q_1 \times 10\,000$ , затем целую часть произведения  $Q_2 \times 10\,000$ .

Возможные данные одного из файлов проиллюстрированы графиком.

Внимание! График приведён в иллюстративных целях для произвольных значений, не имеющих отношения к заданию. Для выполнения задания используйте данные из прилагаемого файла.



```
clA = [[],[ ]]
for s in open('27A.txt'):
    x,y = [float(d) for d in s.replace(',','').split()]
    if x>20: pass
    elif y>10: clA[0].append([x,y])
    else: clA[1].append([x,y])
```

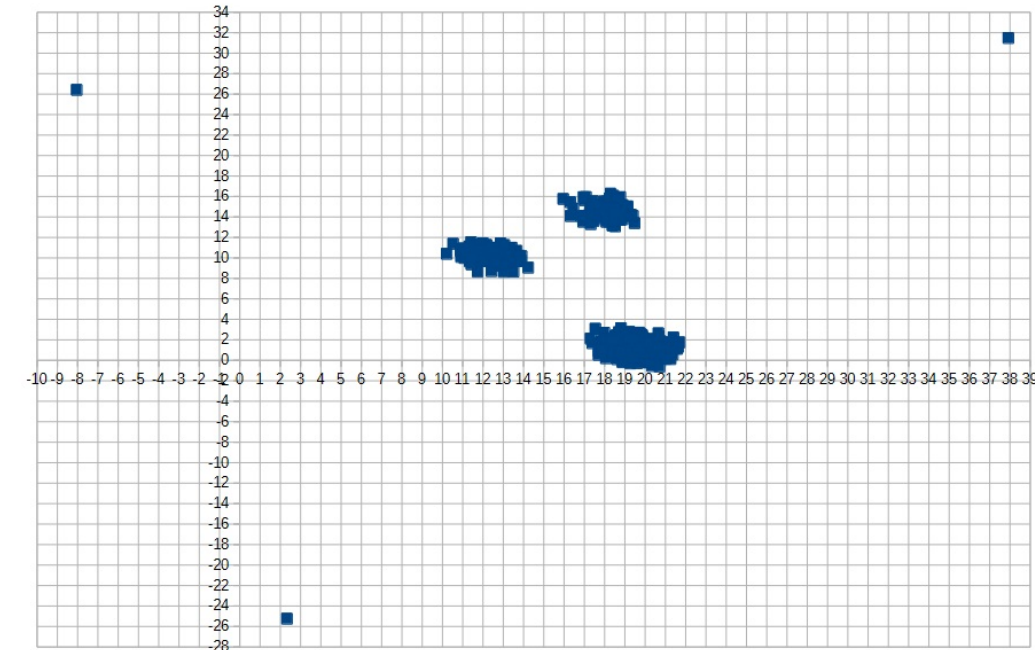
```
clB = [[],[ ],[ ]]
for s in open('27B.txt'):
    x,y = [float(d) for d in s.replace(',','').split()]
    if x<9 or x>25: pass
    elif y<6: clB[0].append([x,y])
    elif x<15: clB[1].append([x,y])
    else: clB[2].append([x,y])
```

```
from math import dist
```

```
def centr(cl):
    m = []
    for p in cl:
        s = sum(dist(p,p1) for p1 in cl)
        m.append([s,p])
    return min(m)[1]
```

```
c0 = centr(clA[0])
c1 = centr(clA[1])
px = int(abs(c0[0] - c1[0])*10000)
py = int(abs(c0[1] - c1[1])*10000)
print(px,py)
```

```
c0 = centr(clB[0])
c1 = centr(clB[1])
c2 = centr(clB[2])
r = [dist(c0,c1), dist(c0,c2), dist(c1,c2)]
q1 = int(min(r)*10000)
q2 = int(max(r)*10000)
print(q1,q2)
```



```
23684 65128
72353 130974
```