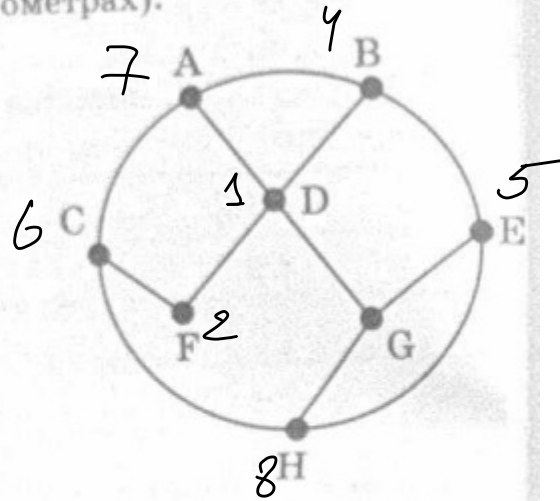


1

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		12	17	6			10	
	2	12					7		
	3	17				16			20
	4	6				4		8	
	5			16	4				9
	6		7					11	18
	7	10			8		11		
	8			20		9	18		



$$10 + 4 = 14$$

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 D и из пункта B в пункт E .

В ответе запишите целое число.

2

Миша заполнял таблицу истинности логической функции F

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

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

	w	x	z	y	F
\rightarrow	0	1		0	0
\rightarrow	0	1	1	1	0
		1	0	0	0

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

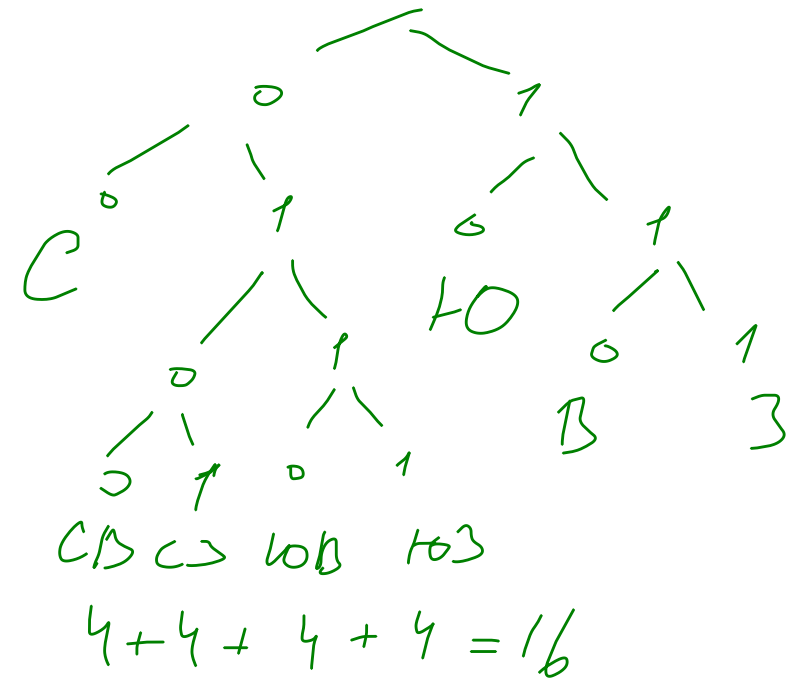
```


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	10
Восток	110
Запад	111

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.



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 число N делится на 3, то к этой записи дописываются три последние двоичные цифры;
 - б) если число N на 3 не делится, то остаток от деления сначала увеличивается на единицу, после чего полученное число умножается на 3, а затем результат умножения переводится в двоичную запись и дописывается в конец числа.
3. Полученная таким образом запись является двоичной записью искомого числа R . Результат переводится в десятичную систему и выводится на экран.

Например, для исходного числа $12_{10} = 1100_2$ результатом является число $1100100_2 = 100_{10}$, а для исходного числа $4_{10} = 100_2$ это число $100110_2 = 38_{10}$.

Укажите максимальное число R , не превышающее 416, которое может быть получено с помощью описанного алгоритма. В ответе запишите это число в десятичной системе счисления.

411

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

File Edit Format Run Options Window Help

```

m = []
for n in range(1,500):
    b = f'{n:b}'
    if n%3==0:
        b = b + b[-3:]
    else:
        b = b + f'{(n%3+1)*3:b}'
    r = int(b,2)
    if r<=416:
        m.append(r)
print(max(m))

```

Черепаше был дан для исполнения следующий алгоритм:
Повтори 4 [Вперёд 9 Налево 180 Назад 10 Направо 90]

Поднять хвост

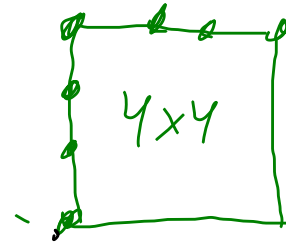
Назад 7 Налево 90 Вперёд 3 Направо 90

Опустить хвост

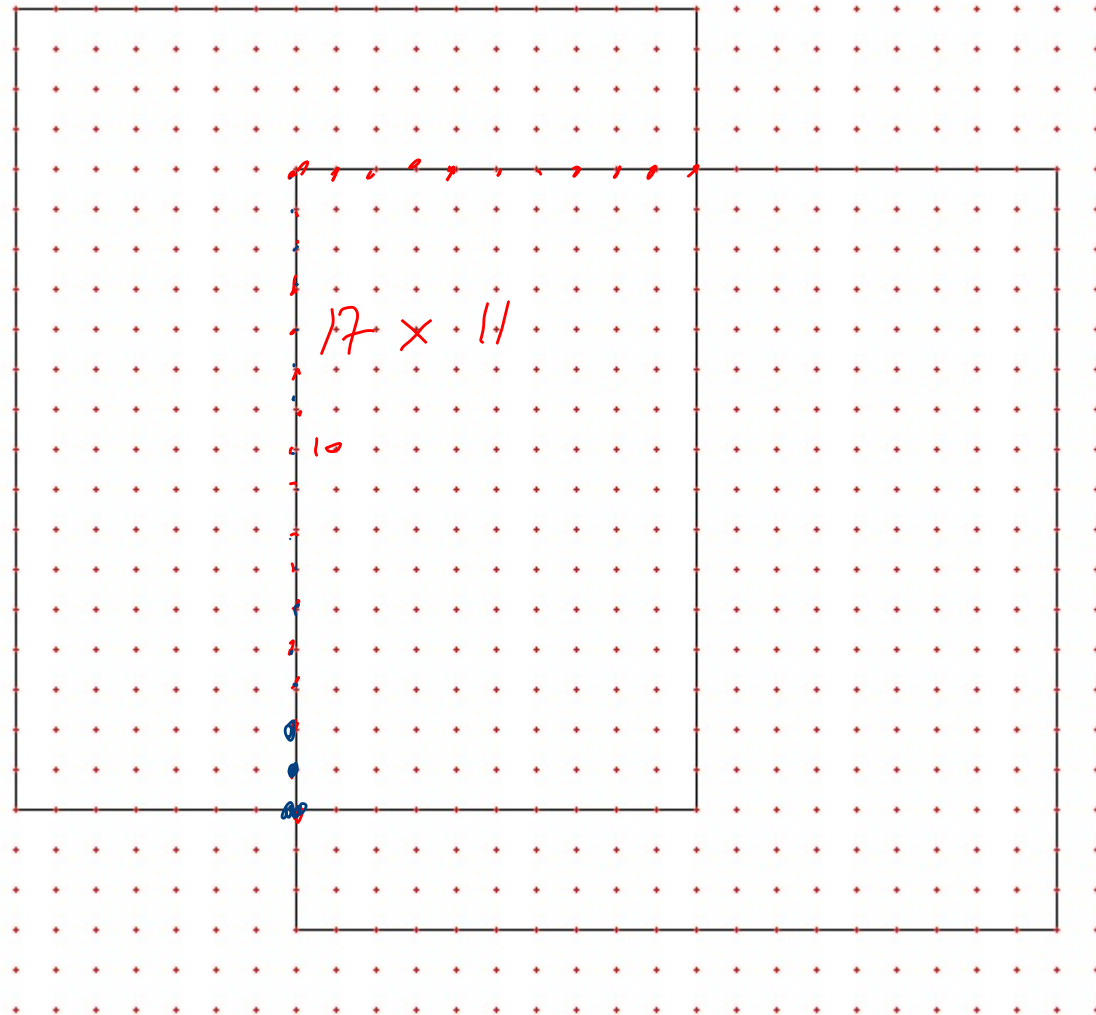
Повтори 2 [Вперёд 17 Налево 90 Вперёд 20 Налево 90]

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

квадрат 19x19



18x21



17x11

10

20x20

$$\begin{array}{l} > 20 \times 20 + 18 \times 21 - 17 \times 11 \\ \hline 591 \end{array}$$

Путешественник фотографирует достопримечательности с помощью камеры смартфона. Каждая фотография представляет собой растровое изображение размером 2000×1000 пикселей, при этом используется палитра из 2^{25} цветов. В конце дня путешественник отправляет снимки родственникам с помощью приложения-мессенджера. Для экономии трафика приложение оцифровывает снимки повторно, используя размер 800×700 пикселей и глубину цвета 15 бит. Сколько Кбайт трафика экономится при передаче 40 фотографий?
В ответе укажите целую часть полученного числа.

Экономия = $2000 \times 1000 \times 25 - 800 \times 700 \times 15$
 $40 \times \text{Экономия} / 2 \times 13$
203125.0

Все пятибуквенные слова, составленные из букв С, Т, Р, Е, Л, А, записаны в алфавитном порядке и пронумерованы.

Вот начало списка:

1. ААААА
2. ААААЕ
3. ААААЛ
4. ААААР
5. ААААС
6. ААААТ

...

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

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

4518 РЛТЛТ

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('АЕЛРСТ', repeat=5):  
    s = ''.join(x)  
    k += 1  
    if k%2==0 and s[0] not in 'АСТ' and s.count('Л')==2 and 'ЛЛ' not in s:  
        print(k,s)
```



Задание выполняется с использованием

9

Откройте файл электронной таблицы, содержащей в каждой строке семь натуральных чисел. Определите наибольший номер строки таблицы, для чисел которой выполнены оба условия:

- в строке одно число повторяется три раза, остальные четыре числа различны;
- сумма неповторяющихся чисел строки больше суммы её повторяющихся чисел.

В ответе запишите только число.

11597

09.py - C:/Users/axelo/OneDrive/Рабочий стол/09.py (3.12.5)

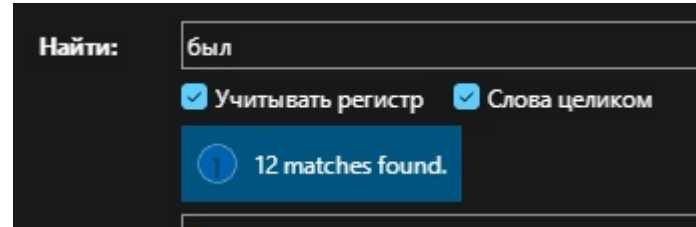
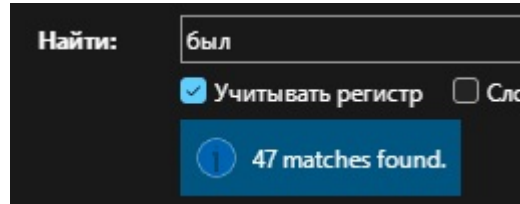
File Edit Format Run Options Window Help

```
k = 0
```

```
for s in open('09.txt'):
    a = [int(x) for x in s.split()]
    k += 1
    a1 = [x for x in a if a.count(x)==1]
    a3 = [x for x in a if a.count(x)==3]
    if len(a3)==3 and len(a1)==4 and sum(a1)>sum(a3):
        print(k)
```

10

С помощью текстового редактора определите, сколько раз встречается сочетание букв «был» со строчной буквы в составе других слов, но не как отдельное слово в тексте глав X, XI и XII романа И. С. Тургенева «Рудин». В ответе укажите только число.



$$47 - 12 = \underline{35}$$

11

Каждому изготовленному на заводе двигателю присваивается уникальный серийный номер, содержащий десятичные цифры и символы из 70-символьного специального алфавита. 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 1234567 серийных номеров занимают более 24 Мбайт памяти. Определите минимально возможную длину серийного номера. В ответе запишите только целое число.

А	В
Кодировка	80
Размер в битах	7
Длина номера	23
Размер номера	21
> 24 Мб	ИСТИНА

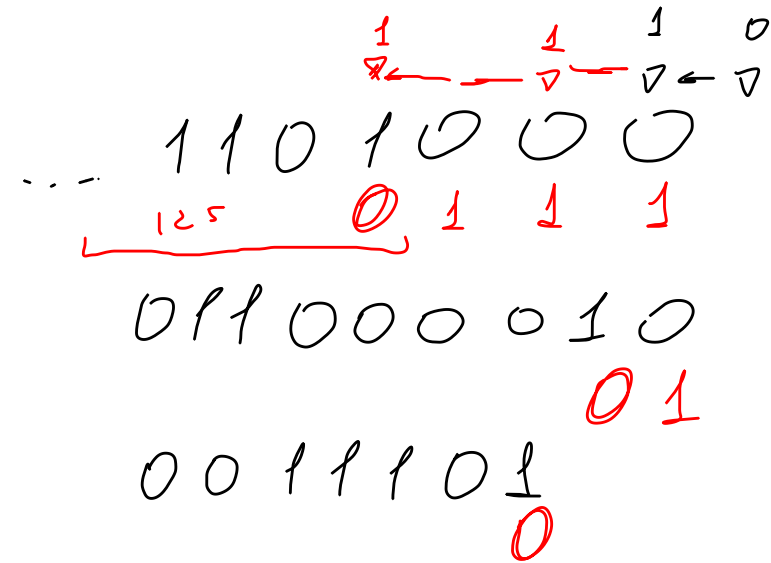
Выполните задание.

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

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 максимально возможное число нулей в исходной последовательности.



Max 519 нулей в начальной строке

13

В терминологии сетей TCP/IP маской сети называется двоичное значение, 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 77.180.176.14 и сетевой маской 255.255.254.0.

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.

77.180.177.254

13.py - C:\Users\axelo\OneDrive\Рабочий стол\13.py (3.12.5)

File Edit Format Run Options Window Help

```
from ipaddress import *
```

```
net = ip_network('77.180.176.14/255.255.254.0',0)
```

```
print(net[-2])
```

14

Значение арифметического выражения

$$3 \cdot 2187^{1801} + 729^{2000} - 4 \cdot 243^{2100} + 81^{2200} - 2 \cdot 27^{2400} - 13122$$

записали в системе счисления с основанием 27. Определите в 27-ричной записи числа количество цифр с числовым значением, превышающим 8.

3432

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

File Edit Format Run Options Window Help

```
a = 3*2187**1801 + 729**2000 - 4*243**2100 + 81**2200 - 2*27**2400 - 13122
```

```
k = 0
```

```
while a>0:
```

```
    if a%27>8: k+=1
```

```
    a = a//27
```

```
print(k)
```

15

На числовой прямой дан отрезок $A = [3;60]$; B — множество всех натуральных делителей числа 177, отличных от единицы и от самого числа 177; C — множество всех натуральных делителей некоторого натурального числа y , отличных от единицы и от самого числа y (число y таково, что множество C непустое). Укажите наибольшее возможное значение числа y , для которого выражение

$$(x \in C) \rightarrow ((x \in A) \wedge \neg(x \in B))$$

истинно (т. е. принимает значение 1) при любом значении переменной x .

2491

2809

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

File Edit Format Run Options Window Help

```
def f(x):
    A = 3<=x<=60
    B = x in [3,59]
    C = x in dy
    return C <= (A and not B)

for y in range(1,10000):
    dy = [x for x in range(2,y) if y%x==0]
    if len(dy)>0 and all(f(x)==1 for x in range(1,100000)):
        print(y)
```

16

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

$$F(1) = 2;$$

$$F(n) = 3 \cdot F(n - 1) - n, \text{ если } n \geq 2.$$

Чему равно значение выражения $\frac{F(2025) - F(2023) - 1}{3^{2022}}$?

6.0

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

File Edit Format Run Options Window Help

```
from functools import *
```

```
@lru_cache(None)
```

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

```
    if n==1: return 2
```

```
    return 3*f(n-1)-n
```

```
for i in range(2,2026): f(i)
```

```
print((f(2025)-f(2023)-1)/3**2022)
```

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 не менее двух из трёх элементов являются двузначными числами, а сумма элементов тройки превосходит сумму минимального двузначного и максимального двузначного элементов последовательности. В ответе запишите количество найденных троек чисел, затем максимальную из сумм элементов таких троек. В данной задаче под тройкой подразумевается три идущих подряд элемента последовательности.

8 99191

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('17var01.txt')]

mn = min(x for x in a if 10<=abs(x)<100)
mx = max(x for x in a if 10<=abs(x)<100)

ans = []

for x,y,z in zip(a,a[1:],a[2:]):
    if (10<=abs(x)<100)+(10<=abs(y)<100)+(10<=abs(z)<100)>=2 and\
        x+y+z > mn+mx:
        ans.append(x+y+z)
print(len(ans), max(ans))
```

Квадрат разлинован на $N \times N$ клеток ($1 < N < 30$). 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. Квадрат ограничен внешними стенами. Между соседними клетками квадрата также 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 клеткам маршрута Робота.

В «угловых» клетках поля — тех, которые справа и снизу ограничены стенами, Робот не может продолжать движение, поэтому накопленная сумма считается итоговой. Таких конечных клеток на поле может быть несколько, включая правую нижнюю клетку поля. При разных запусках итоговые накопленные суммы могут различаться.

Определите максимальную и минимальную денежные суммы среди всех возможных итоговых сумм, которые может собрать Робот, пройдя из левой верхней клетки в конечную клетку маршрута. В ответе укажите два числа — сначала максимальную сумму, затем минимальную.

Исходные данные представляют собой электронную таблицу размером $N \times N$, каждая ячейка которой соответствует клетке квадрата. Внутренние и внешние стены обозначены утолщёнными линиями.

Пример входных данных

1	8	8	4
9	1	5	3
1	3	12	2
2	2	6	6

40	84	94	130	200	218	276	308	346	359	389	447	517	562	585	658	688	716	742	766
53	118	120	154	192	238	287	295	375	363	432	462	480	485	528	584	621	656	731	755
125	136	787	228	207	246	279	308	355	374	394	471	498	523	551	622	627	706	752	831
183	202	265	340	267	286	290	367	386	442	439	486	552	597	587	618	698	743	774	806
225	259	592	463	367	321	311	331	356	394	484	494	544	566	579	582	651	691	753	791
256	370	492	593	567	334	357	381	379	399	559	561	559	612	582	630	682	725	734	788
277	388	721	793	887	342	399	456	445	428	597	610	605				713	784	768	813
356	478	1243	1023	1111	392	397	464	497	458	478	542	555				766	791	806	880
371	491	603	825	891	400	412	488	520	488	504	548	574				794	823	839	883
385	545	548	604	635	479	425	568	596	503	556	627	610				853	831	835	930
406	624	593	645	689	528	458	629	633	549	590	613	620				877	833	876	1008
452	665	666	652	707	601	498	696	678	571	635	678	685				896	853	862	1055
474	688	676	663	695	626	566	572	625	585	650	691	747	824	885	909	974	867	882	1129
539	725	740	706	753	706	619	623	681	662	655	708	731	743	781	854	910	874	889	1167
601	780	748	744	805	775	681	643	707	717	696	731	801	777	794	865	943	993	1003	1014
665	796	825	780	806	836	693	680	690	718	729	742	806	847	849	923	930	987	1020	1051
745	799	817	813	865	861	704	730	750	792	765	792	818	882	902	918	927	964	975	1008
815	846	905	967	872	883	745	809	770	870	881	992	1014	961	992	1069	996	985	1028	1067
863	887	946	1019	877	906	756	814	803	858	934	963	1038	978	994	1044	1049	1045	1061	1117
908	900	952	982	882	895	799	869	873	957	1057	1606	1116	1011	1064	1065	1058	1124	1084	1116



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 60. Победителем считается игрок, сделавший последний ход, то есть первым получивший кучу из 60 или менее камней. В начальный момент в куче было S камней, $S \geq 61$.

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.

16

20

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

- РЕСТАРТ: C:/Users/axelo/OneDrive/Рабочий стол/19.py

19 [244, 245, 246]

20 [247, 248, 249, 250, 251, 976, 977, 978, 979, 980, 981, 982, 983, 984, 985, 986, 987]

21 [252, 253, 254, 988, 989, 990]

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<=60: 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(61,1000) if f(s,2)])
```

```
print(20,[s for s in range(61,1000) if not f(s,1) and f(s,3)])
```

```
print(21,[s for s in range(61,1000) 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 ~~максимальное~~ количество процессов, которые параллельно выполняются на 15-й мс. Считать, что каждый процесс начинается в самое раннее допустимое время. Нумерация миллисекунд начинается с 1.

Ответ: _____.

Примечание. В дополнительных материалах приведён типовой пример организации данных во входном файле, имеющий иллюстративный характер.

A	B	C	D	E	F	G	H	I	J	K
ID процесса B	Время выполнения процесса B (мс)									
2	6	6	0	0	9	0	0	10	15	ИСТИНА
9	5	11	0	0	12	0	0	13	17	ИСТИНА
10	4	11	0	0	12	0	0	13	16	ИСТИНА
13	5	4	14	0	9	12	0	13	17	ИСТИНА
16	3	11	0	0	12	0	0	13	15	ИСТИНА

23

Исполнитель преобразует число на экране.

У исполнителя есть две команды, которые обозначены номерами:

1. Прибавь 1
2. Поменяй местами

Первая из этих команд увеличивает число на экране на 1. Вторая команда может применяться только к числу, у которого цифра разряда десятков по значению меньше цифры, стоящей в разряде единиц, и действует, заменяя число на экране числом, в котором цифры двух младших разрядов поменялись местами.

Программа для исполнителя — это последовательность команд.

Сколько существует программ, для которых при исходном числе 101 результатом является число 154?

Ответ:

1
123
012

89

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: return 0
    if c==e: return 1
    s = str(c)
    if int(s[1])<int(s[2]):
        return f(c+1,e) + f(int(s[0]+s[2]+s[1]),e)
    else:
        return f(c+1,e)
print(f(101,154))
```

24

Текстовый файл состоит из десятичных цифр и заглавных букв латинского алфавита. Определите в прилагаемом файле последовательность из максимального количества идущих подряд символов, среди которых ровно 35 нечётных цифр и при этом начинающуюся с буквы S, не содержащую других букв S, кроме первой. В ответе запишите число — количество символов в найденной последовательности. Для выполнения этого задания следует написать программу.

272

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

File Edit Format Run Options Window Help

```
s = open('24var01.txt').readline()
s = s.replace('3','1').replace('5','1').replace('7','1').replace('9','1')

m = 1

for l in range(len(s)):
    if s[l]=='S':
        for r in range(l+m, len(s)):
            c = s[l:r]
            if c.count('S')>1 or c.count('1')>35: break
            if c.count('1')==35:
                m = max(m, len(c))

print(m)
```

25

Напишите программу, которая перебирает все целые шестизначные числа в порядке возрастания и ищет среди них те, которые не содержат в своей записи цифру 1 и которые можно представить в виде суммы нечётного натурального числа, делящегося без остатка на 103, и натуральной степени числа 3. В ответе в первом столбце таблицы запишите первые 5 найденных чисел в порядке возрастания, а во втором столбце — для каждого числа соответствующий показатель степени числа 3. Количество строк в таблице для ответа избыточно.

```
200004 4
200034 6
200036 9
200050 7
200056 10
```

25.py - C:/Users/axelo/OneDrive/Рабочий стол/25.py (3.12.5)

File Edit Format Run Options Window Help

```
k = 0
for x in range(100000,1000000):
    if k==5: break
    if '1' not in str(x):
        for n in range(1,13):
            t = x - 3**n
            if t>0 and t%2!=0 and t%103==0:
                print(x,n)
                k += 1
```

26

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. 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.

Ответ:

Примечание. В дополнительных материалах приведён типовой пример организации данных во входном файле, имеющий иллюстративный характер.

Самолётки



Команды



679 194496

```
26.py - C:/Users/axelo/OneDrive/Рабочий стол/26.py (3.12.5)
File Edit Format Run Options Window Help
f = open('26var01.txt')
N, M = [int(x) for x in f.readline().split()]

cmd = [int(f.readline()) for i in range(N)]
sam = [int(f.readline()) for i in range(M)]

cmd.sort(reverse=1)
sam.sort(reverse=1)

ans = []
for x in sam:
    for i in range(N):
        if cmd[i] != -1 and cmd[i] <= x:
            ans.append(cmd[i])
            cmd[i] = -1
            break
print(len(ans), max(ans))
```

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 $H = 6,5$, $W = 5$ для каждого кластера. Известно, что количество точек не превышает 10 000. Структура хранения информации в файле Б аналогична структуре в файле А.

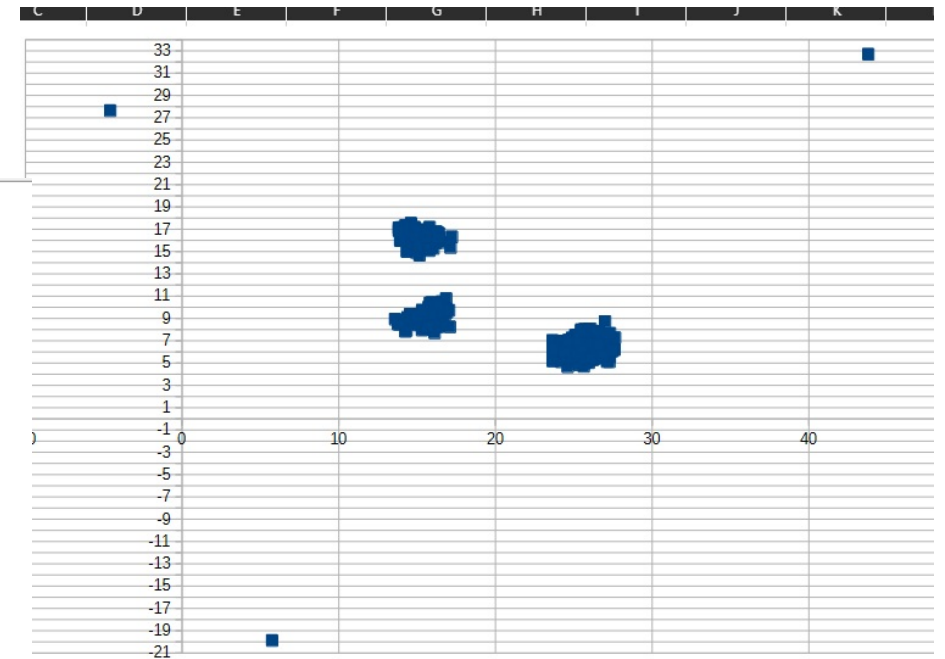
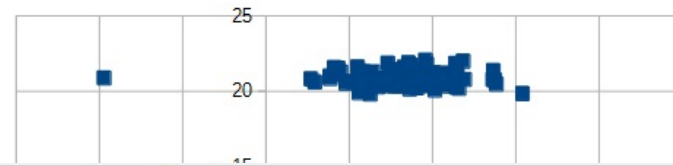
Известно, что в файле А имеются координаты ровно двух, а в файле Б ровно трёх «лишних» точек, представляющих аномалии, которые возникли в результате помех при передаче данных. Эти точки не относятся ни к одному из кластеров, их учитывать не нужно.

Для файла А определите координаты центра каждого кластера, затем найдите два числа: P_1 — минимальное расстояние между центром одного кластера и точкой другого кластера и P_2 — максимальное расстояние между центром кластера и точкой другого кластера.

Для файла Б определите координаты центра каждого кластера, затем найдите два числа: Q_1 — среднее арифметическое расстояний от центра кластера с минимальным количеством точек до точек этого кластера и Q_2 — среднее арифметическое расстояний от центра кластера с максимальным количеством точек до точек этого кластера. Гарантируется, что во всех кластерах количество точек различно. Нулевое расстояние от центра кластера до самого себя не учитывается.

В ответе запишите четыре числа: в первой строке — сначала целую часть произведения $P_1 \times 10\,000$, затем целую часть произведения $P_2 \times 10\,000$; 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.



```
clA = [[],[[]]
for s in open('27var01A.txt'):
    x,y = [float(d) for d in s.split()]
    if x<0 or y<0:
        pass
    elif y>15: clA[0].append((x,y))
    else: clA[1].append((x,y))
```

```
clB = [[],[[]],[[]]
for s in open('27var01B.txt'):
    x,y = [float(d) for d in s.split()]
    if x<0 or y>20 or y<0:
        pass
    elif y>13: clB[0].append((x,y))
    elif x<20: 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]
```

```
c1 = centr(clA[0])
c2 = centr(clA[1])
ras1 = [dist(c1,p) for p in clA[1]]
ras2 = [dist(c2,p) for p in clA[0]]
p1, p2 = min(ras1+ras2), max(ras1+ras2)
print(p1*10000//1, p2*10000//1)
```

```
print([len(cl) for cl in clB])
c1 = centr(clB[0])
q1 = sum(dist(c1,p) for p in clB[0] if p!=c1)/(len(clB[0])-1)
c2 = centr(clB[2])
q2 = sum(dist(c2,p) for p in clB[2] if p!=c1)/(len(clB[2])-1)
print(q1*10000//1, q2*10000//1)
```