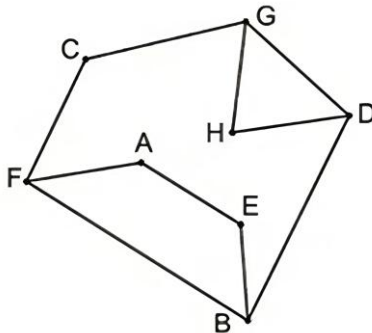


ЗАДАНИЕ 1

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					30		5	
	2				3	2		39	
	3				21		1		
	4		3	21			8		
	5	30	2						
	6			1	8				13
	7	5	39						53
	8						13	53	



Так как таблицу и схему рисовали независимо друг от друга, то нумерация населённых пунктов в таблице никак не связана с буквенными обозначениями на графе. Определите, какова сумма протяжённости дорог из пункта G в пункт C и из пункта B в пункт D. В ответе запишите целое число.

ОТВЕТ: 16

```
from itertools import *

# из каждой строки выписываем города, с которыми соединяется строка
tab = '57 457 46 236 12 348 128 67'.split()

# выписываем все соединения из графа
pic = 'cg cf gh gd hd db eb ae af fb'.split()

print(*range(1, 9))
for var in permutations('abcdefgh'):
    if all(str(var.index(x) + 1) in tab[var.index(y)] for x, y in pic):
        print(*var)
```

ЗАДАНИЕ 2

Миша заполнял таблицу истинности логической функции

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

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

				F
0			0	1
1	0			1
	1			1

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

ОТВЕТ: yxzw

1 вариант решения. Аналитика

```
print('x y z w')
for x in range(2):
    for y in range(2):
        for z in range(2):
            for w in range(2):
                if (x <= y) and (z == (w <= x)) and (not w) == 1:
                    print(x, y, z, w)
```

2 вариант решения. Автокод

```
from itertools import *

def fun(x, y, z, w):
    return (x <= y) and (z == (w <= x)) and (not w)

# буквы - это пробелы в таблице
for a, b, c, d, e, f, g in product((0, 1), repeat=7):
    table = ((0, a, b, 0, 1),
              (1, 0, c, d, 1),
              (e, 1, f, g, 1))
    if len(table) == len(set(table)):
        for p in permutations('wxyz'):
            if all(fun(**dict(zip(p, line))) == line[-1] for line in table):
                print(*p)
```

ЗАДАНИЕ 5

Десятичные числа

Автомат получает на вход четырёхзначное натуральное число и строит новое число по следующему алгоритму:

1. вычисляются суммы первой и второй, второй и третьей, третьей и четвёртой цифр;
2. из полученных сумм отбрасывается наименьшая;
3. остальные суммы записываются в порядке неубывания.

Пример. Исходное число: 1284. Суммы: $1 + 2 = 3$; $2 + 8 = 10$; $8 + 4 = 12$. Отбрасывается наименьшая сумма 3. Результат: 1012.

Укажите наибольшее число, при вводе которого автомат выдаёт значение 1115.

ОТВЕТ: 9656

```
for i in range(9999, 999, -1):
    ns = str(i) # создаем строку из числа, чтобы брать оттуда цифры
    x = int(ns[0]) + int(ns[1])
    y = int(ns[1]) + int(ns[2])
    z = int(ns[2]) + int(ns[3])
    sums = sorted([x, y, z])
    r = str(sums[1]) + str(sums[2])
    if r == '1115':
        print(i)
        break # нашли наибольший ответ, дальше не смотрим
```

Двоичные числа

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

- 1) Строится двоичная запись числа N.
- 2) Затем справа дописываются два разряда: символы 10, если число N чётное, и 11, если нечётное.

Полученная таким образом запись (в ней на два разряда больше, чем в записи исходного числа N) является двоичной записью искомого числа R. Укажите минимальное число N, после обработки которого автомат получает число, большее 105. В ответе это число запишите в десятичной системе.

ОТВЕТ: 26

```
for n in range(1000):
    r = f'{n:b}'
    if n % 2 == 0:
        r += '10'
    else:
        r += '11'
    r = int(r, 2)
    if r > 105:
        print(n)
        break
```

Другие системы счисления

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 кратно 3, то в конец записи дописываются две последние троичные цифры.
3. Если N не кратно 3, то остаток от деления умножается на 5, переводится в троичную систему и затем дописывается к числу.

Полученная таким образом запись является троичной записью искомого числа R .

Укажите максимальное число R , не превышающее 173, которое может быть получено с помощью описанного алгоритма. В ответе запишите это число в десятичной системе счисления.

ОТВЕТ: 26

```
def f(n):
    s = ''
    while n > 0:
        s = str(n % 3) + s
        n //= 3
    return s

c = []
for n in range(1, 100):
    s = f(n)
    if n % 3 == 0:
        s = s + s[-2:]
    else:
        s = s + f((n % 3) * 5)
    r = int(s, 3)
    if r < 173:
        c.append(r)
print(max(c))
```

ЗАДАНИЕ 6

Черепашке был дан для исполнения следующий алгоритм:

```
Повтори 3 [Вперёд 11 Направо 90 Вперёд 24 Направо 90]
Поднять хвост
Вперёд 2 Направо 90 Вперёд 5 Налево 90
Опустить хвост
Повтори 3 [Вперёд 6 Налево 90 Вперёд 9 Налево 90]
```

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

ОТВЕТ: 328

```
from turtle import *

tracer(0) # рисунок сразу появляется
lt(90) # на ось ординат
screensize(2000, 2000) # полосы прокрутки
k = 20 # коэффициент масштаба
for i in range(3):
    fd(11 * k)
    rt(90)
    fd(24 * k)
    rt(90)
up()
fd(2 * k)
rt(90)
fd(5 * k)
lt(90)
down()
for i in range(3):
    fd(6 * k)
    lt(90)
    fd(9 * k)
    lt(90)

up()
# расставляем точки
for x in range(-1, 35):
    for y in range(-1, 35):
        goto(x * k, y * k)
        dot(3)
done()
```

ЗАДАНИЕ 8

Слова по порядку

Все 7-буквенные слова, в составе которых могут быть только буквы E, G, L, A, N, D, записаны в алфавитном порядке и пронумерованы.

Вот начало списка:

1. AAAAAAA
2. AAAAAAD
3. AAAAAAE
4. AAAAAAG
5. AAAAAAL
6. AAAAAAN

Под каким номером в списке идёт слово EGELAND?

ОТВЕТ: 120128

```
from itertools import *

number = 0
for i in product('ADEGLN', repeat=7):
    number += 1
    s = ''.join(i)
    if s == 'EGELAND':
        print(number)
        break
```

Слова с ограничениями (product)

Виктор составляет таблицу кодовых слов для передачи сообщений, каждому сообщению соответствует своё кодовое слово. В качестве кодовых слов Виктор использует 5-буквенные слова, в которых могут быть только буквы Д, Г, И, А, Ш, Э, причём слово не должно начинаться с гласной или заканчиваться согласной. Сколько различных кодовых слов может использовать Виктор?

ОТВЕТ: 1944

```
from itertools import *

cnt = 0
for i in product('ДГИАШЭ', repeat=5):
    s = ''.join(i)
    if (s[0] not in 'ИАЭ') and (s[-1] not in 'ДГШ'):
        cnt += 1
print(cnt)
```

Числа с ограничениями (product)

Определите количество восьмеричных пятизначных чисел, которые не начинаются с нечётных цифр, не оканчиваются цифрами 2 или 6, а также содержат не более двух цифр 7.

ОТВЕТ: 9135

```
from itertools import *
cnt = 0
for i in product('01234567', repeat=5):
    s = ''.join(i)
    # первая цифра не ноль
    if (s[0] != '0') and (int(s[0]) % 2 == 0):
        if (s[-1] not in '26') and (s.count('7') <= 2):
            cnt += 1
print(cnt)
```

Перестановки (permutations):

Леонид составляет коды перестановкой букв слова СОРТИРОВКА. При этом в кодах никакие три согласные или три гласные буквы не должны стоять рядом. Сколько различных кодов может составить Леонид?

ОТВЕТ: 185760

```
from itertools import *
cnt = 0
# ОБЯЗАТЕЛЬНО SET
for i in set(permutations('СОРТИРОВКА')):
    s = ''.join(i)
    for x in 'CPTBK':
        s = s.replace(x, 'C')
    for x in 'OIA':
        s = s.replace(x, 'O')
    if 'CCC' not in s and 'OOO' not in s:
        cnt += 1
print(cnt)
```

ЗАДАНИЕ 9

Откройте файл электронной таблицы, содержащей в каждой строке четыре натуральных числа. Определите количество строк таблицы, для которых выполняются следующие условия:

1. Числа, содержащиеся в строке, можно разбить на две пары так что произведения чисел в них равны.
2. Квадрат второго максимального из чисел, содержащихся в строке, больше произведения минимального и максимального чисел в строке.

ОТВЕТ: 13

```
f = open("9")
cnt = 0
for s in f:
    #строки в числа
    m = [int(x) for x in s.split()]
    m.sort()
    # первое условие выполнимо только с такими парами
    if (m[0] * m[3]) == (m[1] * m[2]):
        # второе условие
        if m[-2]**2 > m[0] * m[3]:
            cnt += 1
print(cnt)
```


ЗАДАНИЕ 11

Мощность алфавита

На предприятии каждой изготовленной детали присваивается серийный номер, состоящий из 377 символов. В базе данных для хранения каждого серийного номера отведено 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 23155 серийных номеров требуется более 5536 Кбайт памяти. Определите минимально возможную мощность алфавита, используемого для записи серийных номеров. В ответе запишите только целое число.

ОТВЕТ: 33

```
from math import *

for n in range(1, 1000):
    i = ceil(log2(n))
    ser = ceil(377 * i / 8)
    if 23155 * ser > 5536 * 1024:
        print(n)
        break
```

Длина кода

После масштабной утечки данных с завода по сборке роботов каждая новая деталь стала маркироваться уникальным серийным номером, чтобы предотвратить её использование в целях машинного восстания. Номер содержит десятичные цифры, 26 латинских букв (без учёта регистра) и символы из 496-символьного алфавита, которые добавлены для дополнительной безопасности. Символы кодируются равным минимальным числом битов, а для каждого номера выделяется одинаковое минимально возможное целое количество байт. Для 725 серийных номеров требуется более 353 Кбайт памяти.

Какова минимально возможная длина серийного номера? Ответ запишите в виде целого числа.

ОТВЕТ: 399

```
from math import *

n = 10 + 26 + 496 # мощность алфавита
i = ceil(log2(n))

for l in range(1, 1000):
    id = ceil(l * i / 8) # байт на айди
    if id * 725 > 353 * 1024:
        print(l)
        break
```

ЗАДАНИЕ 12

Нахождение конечной строки

Какая строка получится в результате применения приведенной ниже программы к строке, состоящей из 111 идущих подряд цифр 3? В ответе запишите количество единиц в полученной строке.

НАЧАЛО

ПОКА нашлось (33333) ИЛИ нашлось (1111)

 ЕСЛИ нашлось (33333)

 ТО заменить (33333, 111)

 ИНАЧЕ заменить (111, 33)

 КОНЕЦ ЕСЛИ

КОНЕЦ ПОКА

КОНЕЦ

ОТВЕТ: 6

```
s = 111 * "3"
while "33333" in s or "1111" in s:
    if "33333" in s:
        s = s.replace("33333", "111", 1)
    else:
        s = s.replace("111", "33", 1)
print(s.count("1"))
```

Нахождение начальной строки

Дана программа для Редактора:

ПОКА нашлось (1) ИЛИ нашлось (2)

 заменить(1, 22)

 заменить(2, 3)

КОНЕЦ ПОКА

Исходная строка содержала только единицы и двойки. После выполнения программы получилась строка "333333". Какое максимальное количество единиц могла содержать начальная строка, если известно, что была минимум одна "2"

ОТВЕТ: 2

```
ans = []
for x in range(10): # обозначим за x количество единиц
    for y in range(1, 10): # за y - количество двоек
        s = "1" * x + "2" * y
        while "1" in s or "2" in s:
            s = s.replace("1", "22", 1)
            s = s.replace("2", "3", 1)
        if s == "333333":
            ans.append(x)
print(max(ans))
```

ЗАДАНИЕ 13

Сколько IP-адресов?

Сеть задана IP-адресом 211.46.0.0 и сетевой маской 255.255.128.0. Сколько в этой сети IP-адресов, для которых количество единиц в двоичной записи IP-адреса кратно 4, а сами адреса в двоичном виде заканчиваются на 11? В ответе укажите только число.

ОТВЕТ: 2016

```
from ipaddress import *

# создаем сеть
n = ip_network('211.46.0.0/255.255.128.0', 0)
cnt = 0
for ip in n:
    s = f'{ip:b}' # заполняем ведущими нулями
    if (s.count('1') % 4 == 0) and (s[-2:] == '11'):
        cnt += 1
print(cnt)
```

Наибольший IP-адрес

Сеть задана IP-адресом одного из входящих в неё узлов 143.168.72.213 и сетевой маской 255.255.255.240.

Определите наибольший IP-адрес данной сети, который может быть присвоен компьютеру. В ответе укажите найденный IP-адрес без разделителей.

Например, если бы найденный адрес был равен 111.22.3.44, то в ответе следовало бы записать 11122344.

ОТВЕТ: 14316872222

```
from ipaddress import *

net = ip_network("143.168.72.213/255.255.255.240", 0)
print(net[-2])
```

ЗАДАНИЕ 14

Операнды

Операнды арифметического выражения записаны в системе счисления с основанием 27.

$$8x38x68_{27} + 37x3163_{27}$$

В записи чисел переменной x обозначена неизвестная цифра из алфавита 27-ричной системы счисления. Определите наибольшее значение x , при котором значение данного арифметического выражения кратно 26. Для найденного x вычислите частное от деления значения арифметического выражения на 26 и укажите его в ответе в десятичной системе счисления. Основание системы счисления указывать не нужно.

ОТВЕТ: 176999110

```
# развернули строку, так как нужен наибольший икс
for x in "0123456789abcdefghijklmnopq"[:-1]:
    a = int(f"8{x}38{x}68", 27)
    b = int(f"37{x}3163", 27)
    if (a + b) % 26 == 0:
        print((a + b) // 26)
        break
```

Степени

Значение арифметического выражения

$$3 \cdot 256^{320} - 2 \cdot 64^{290} + 4^{250} - 1023$$

записали в системе счисления с основанием 4. Найдите количество ненулевых разрядов в этой записи.

ОТВЕТ: 657

1 вариант решения

```
from string import *
alf = "0123456789" + ascii_lowercase
def perevod(x, osn):
    s = ""
    while x > 0:
        ost = x % osn
        s = alf[ost] + s
        x = x // osn
    return s

n = 3*256**320 - 2*64**290 + 4**250 - 1023
temp = perevod(n, 4)
print(len(temp) - temp.count("0"))
```

2 вариант решения:

```
s = 3*256**320 - 2*64**290 + 4**250 - 1023
cnt = 0
while s != 0:
    if s % 4 != 0: # считаем количество ненулевых остатков
        cnt += 1
    s //= 4
print(cnt)
```

Степени с иксом:

Значение арифметического выражения $6^{900} + 6^{10} - x$, где x – натуральное число, не превышающее 10000, записали в системе счисления с основанием 6. Определите максимальное значение x , при котором данная запись содержит одинаковое количество цифр «3» и «5».

ОТВЕТ: 9591

```
for x in range(10000, 0, -1):
    s = 6**900 + 6**10 - x
    temp = ""
    while s != 0:
        temp = str(s % 6) + temp
        s //= 6
    if temp.count("3") == temp.count("5"):
        print(x)
        break
```

ЗАДАНИЕ 15

Делы

Обозначим через $\text{ДЕЛ}(x, y)$ утверждение «натуральное число x делится без остатка на натуральное число y ». Для какого наибольшего натурального числа A логическое выражение

$$(\neg \text{ДЕЛ}(x, 7) \wedge \text{ДЕЛ}(x, 13)) \rightarrow (x > A - 40)$$

истинно при любом натуральном значении переменной x ?

ОТВЕТ: 52

```
def f(x):  
    return ((not (x % 7 == 0)) and (x % 13 == 0)) <= (x > (a - 40))  
  
for a in range(1000, 0, -1):  
    if all(f(x) for x in range(1, 2000)):  
        print(a)  
        break
```

Больше/меньше

Для какого наименьшего натурального числа A формула

$$(y > 10) \vee (x * A > y + x)$$

тождественно истинна, т.е. принимает значение 1 при любых натуральных x и y ?

ОТВЕТ: 12

```
def f(x, y):  
    return (y > 10) or ((x * a) > (y + x))  
  
for a in range(1, 1001):  
    if all(f(x, y) for x in range(1, 1001) for y in range(1, 1001)):  
        print(a)  
        break
```

Отрезки мин

На числовой прямой даны два отрезка: $M = [32; 68]$ и $N = [54; 76]$. Укажите наименьшую возможную длину такого отрезка A , для которого логическое выражение $\neg((x \in M) \vee (x \in N)) \equiv \neg(x \in A)$

тождественно истинно при любом значении переменной x

ОТВЕТ: 44

```
m = list(range(32, 69))
n = list(range(54, 76))
a = [] # если наименьшую длину, то начинаем с пустого списка
for x in range(101):
    if not((not (x in m or x in n)) == (x not in a)):
        a.append(x)
print(a)
```

Отрезки макс

На числовой прямой даны два отрезка: $C = [48; 94]$ и $J = [83; 100]$. Укажите наибольшую возможную длину такого отрезка A , для которого логическое выражение $\neg((x \in C) \vee (x \in J)) \rightarrow \neg(x \in A)$

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

ОТВЕТ: 52

```
c = list(range(48, 94))
j = list(range(83, 100))
a = list(range(200)) # если наибольшую длину, то начинаем с полного списка
# берём все x, которые лежат в A
for x in range(200):
    if not((not (x in c or x in j)) <= (x not in a)):
        a.remove(x)
print(a)
```

ЗАДАНИЕ 16

Поиск значения

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

$F(n) = 1$ при $n = 1$;

$F(n) = n \cdot F(n - 1)$, если $n > 1$.

Чему равно значение выражения $F(5000) / F(4998)$?

ОТВЕТ: 5000

```
from functools import *

@lru_cache(maxsize=None)
def f(n):
    if n == 1:
        return 1
    return n * f(n - 1)

for i in range(1, 6000):
    f(i)

print(f(5000) / f(4999))
```

Поиск аргумента

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

$F(n) = 0$ при $n = 0$

$F(n) = F(n / 2) - 1$ при чётных $n > 0$

$F(n) = 1 + F(n - 1)$ при нечётных $n > 0$

Сколько существует чисел n , меньших 1000, для которых значение $F(n)$ будет равно 0?

```
def f(n):
    if n == 0:
        return 0
    if n % 2 == 0:
        return f(n / 2) - 1
    return 1 + f(n - 1)

cnt = 0
for n in range(1000):
    if f(n) == 0:
        cnt += 1
print(cnt)
```


ЗАДАНИЕ 17

В файле содержится последовательность целых чисел, не превышающих по модулю 100 000. Определите количество пар последовательности, в которых хотя бы одно число отрицательно, а сумма чисел пары меньше количества чисел последовательности, кратных 32. В ответе запишите количество найденных пар, затем максимальную из сумм элементов таких пар. В данной задаче под парой подразумевается два идущих подряд элемента последовательности.

ОТВЕТ: 4969 299

```
f = open('17.txt')
a = [int(x) for x in f]
k32 = len([x for x in a if abs(x) % 32 == 0])

ans = []
for i in range(len(a) - 1):
    x, y = a[i], a[i + 1]
    if (x < 0 or y < 0) and (x + y) < k32:
        ans += [x + y]
print(len(ans), max(ans))
```

ЗАДАНИЯ 19-21

1 куча

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 два камня или убрать из кучи пять камней или уменьшить количество камней в куче в три раза (количество камней, полученное при делении, округляется до меньшего). Например, из кучи в 20 камней за один ход можно получить кучу из 18, 15 или 6 камней. Игра завершается, когда количество камней в куче становится не более 19.

Победителем считается игрок, сделавший последний ход, то есть первым получивший кучу, в которой будет 19 или меньше камней. В начальный момент в куче было S камней, $S \geq 20$.

Будем говорить, что игрок имеет выигрышную стратегию, если он может выиграть при любых ходах противника.

Задание 19.

Укажите минимальное значение S , при котором Петя не может выиграть за один ход, но при любом ходе Пети Ваня может выиграть своим первым ходом

Задание 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.

Задание 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.

ОТВЕТ: 60; 62 63; 64

```
def f(s, m): # s-количество камней, m-осталось ходов до конца игры
    if s <= 19: return m % 2 == 0 # условие победы
    if m == 0: return 0
    h = [f(s - 2, m - 1), f(s - 5, m - 1), f(s // 3, m - 1)] # ходы
    return any(h) if m % 2 != 0 else all(h)

print('19', min([s for s in range(20, 100) if f(s, 2)]))
print('20', *[s for s in range(20, 100) if f(s, 3) and (not f(s, 1))])
print('21', min([s for s in range(20, 100) if f(s, 4) and (not f(s, 2))]))
```

2 кучи

Два игрока, Петя и Ваня, играют в следующую игру. Перед игроками лежат две кучи камней. Игроки ходят по очереди, первый ход делает Петя. За один ход игрок может добавить в одну из куч (по своему выбору) один камень или увеличить количество камней в куче в три раза. Для того чтобы делать ходы, у каждого игрока есть неограниченное количество камней.

Игра завершается в тот момент, когда суммарное количество камней в кучах становится не менее 65. Победителем считается игрок, сделавший последний ход, т.е. первым получивший такую позицию, при которой в кучах оказывается 65 или больше камней.

В начальный момент в первой куче было шесть камней, во второй куче – S камней; $1 \leq S \leq 58$.

Будем говорить, что игрок имеет выигрышную стратегию, если он может выиграть при любых ходах противника.

Задание 19.

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

Задание 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.

Задание 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 укажите наименьшее из них.

ОТВЕТ: 7; 10 19; 18

```
def f(a, b, m): # a,b - количества камней в кучах, m-осталось ходов до конца
    игры
    if a+b >= 65: return m % 2 == 0 # условие победы
    if m == 0: return 0
    h = [f(a+1, b, m-1), f(a, b+1, m-1), f(a*3, b, m-1), f(a, b*3, m-1)] # ходы
    return any(h) if m % 2 != 0 else all(h)
    # return any(h) if m % 2 != 0 else any(h) #ДЛЯ 19 НОМЕРА!!!

# print('19)', *[s for s in range(1, 59) if f(6, s, 2)])
print('20)', *[s for s in range(1, 59) if f(6, s, 3) and (not f(6, s, 1))])
print('21)', *[s for s in range(1, 59) if f(6, s, 4) and (not f(6, s, 2))])
```

ЗАДАНИЕ 23

Сложение

У исполнителя есть две команды, которым присвоены номера:

1. Прибавить 1
2. Прибавить следующее

Первая из них увеличивает число на экране на 1, вторая прибавляет к числу на экране число, большее на 1. Программа для исполнителя – это последовательность команд.

Сколько есть программ, которые число 3 преобразуют в число 28, при этом траектория вычислений содержит число 12 и не содержит число 20?

ОТВЕТ: 8

```
def f(s, e):  
    if s > e or s == 20:  
        return 0  
    if s == e:  
        return 1  
    return f(s + 1, e) + f(s + (s + 1), e)  
  
print(f(3, 12) * f(12, 28))
```

Вычитание

Исполнитель Калькулятор преобразует число на экране. У исполнителя есть три команды, которые присвоены номера:

- A. вычти 1
- B. вычти 6
- C. найди целую часть от деления на 2

Первая команда уменьшает число на экране на 1, вторая команда уменьшает это число на 6, третья команда делит число нацело на 2. Программа для исполнителя – это последовательность команд.

Сколько существует таких программ, которые исходное число 34 преобразуют в число 6, и при этом траектория вычислений содержит числа 19 и 29 и не содержит числа 24?

ОТВЕТ: 115

```
def f(s, e):  
    if s < e or s == 24:  
        return 0  
    if s == e:  
        return 1  
    return f(s - 1, e) + f(s - 6, e) + f(s // 2, e)  
print(f(34, 29) * f(29, 19) * f(19, 6))
```

ЗАДАНИЕ 25

Маски

Среди натуральных чисел, не превышающих 10^{10} , найдите все числа, соответствующие маске `3?12?14*5`, делящиеся на 1917 без остатка. В ответе запишите в первом столбце таблицы все найденные числа в порядке возрастания, а во втором столбце – соответствующие им результаты деления этих чисел на 1917.

ОТВЕТ:

351261495 183235
3212614035 1675855
3412614645 1780185
3712414275 1936575
3912414885 2040905

```
from fnmatch import *  
for i in range(1917, 10**10, 1917):  
    if fnmatch(str(i), '3?12?14*5'):  
        print(i, i // 1917)
```

Делители

Пусть M – разность максимального и минимального натуральных делителей целого числа, не считая единицы и самого числа. Если таких делителей у числа нет, то считаем значение M равным нулю.

Напишите программу, которая перебирает целые числа, большие 860 000, в порядке возрастания и ищет среди них такие, для которых значение M оканчивается на 30.

Вывести первые пять найденных чисел и соответствующие им значения M .

Формат вывода: для каждого из пяти таких найденных чисел в отдельной строке сначала выводится само число, затем — значение M .

Строки выводятся в порядке возрастания найденных чисел.

Например, для числа 20 $M = 10 - 2 = 8$.

ОТВЕТ:

860064 430030

860199 286730

860264 430130

860464 430230

860499 286830

```
#функция для нахождения делителей кроме 1 и самого числа
```

```
def divs(n):  
    a = set()  
    for i in range(2, int(n**0.5) + 1):  
        if n % i == 0:  
            a.add(i)  
            a.add(n // i)  
    return a  
  
cnt = 0  
for i in range(860001, 861000):  
    d = divs(i)  
    if len(d) == 0: # если делителей нет, то m = 0  
        m = 0  
    else:  
        m = max(d) - min(d)  
    if m % 100 == 30:  
        print(i, m)  
        cnt += 1  
    #если написали 5 чисел - остановка  
    if cnt == 5:  
        break
```

ЗАДАНИЕ 27

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.

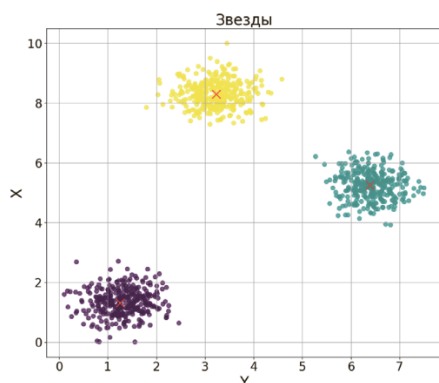
Гарантируется, что такое разбиение существует и единственно для заданных размеров прямоугольников.

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

В файле A хранятся данные о звёздах двух кластеров, где $H=11$, $W=11$ для каждого кластера. В каждой строке записана информация о расположении на карте одной звезды: сначала координата x , затем координата y . Значения даны в условных единицах. Известно, что количество звёзд не превышает 1000.

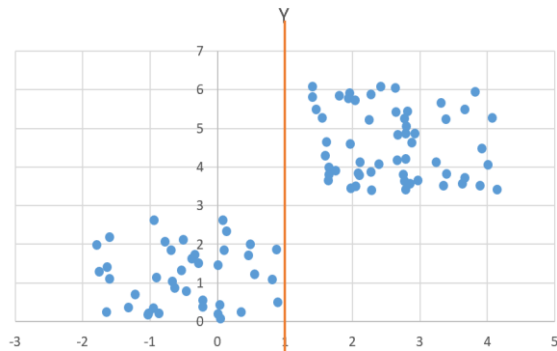
В файле B хранятся данные о звёздах трёх кластеров, где $H=13$, $W=13$ для каждого кластера. Известно, что количество звёзд не превышает 10 000. Структура хранения информации о звёздах в файле B аналогична файлу A.

Для каждого файла определите координаты центра каждого кластера, затем вычислите два числа: P_x – среднее арифметическое абсцисс центров кластеров, и P_y – среднее арифметическое ординат центров кластеров. В ответе запишите четыре числа: в первой строке сначала целую часть произведения $P_x \times 10\,000$, затем целую часть произведения $P_y \times 10\,000$ для файла A, во второй строке – аналогичные данные для файла B. Возможные данные одного из файлов иллюстрированы графиком.

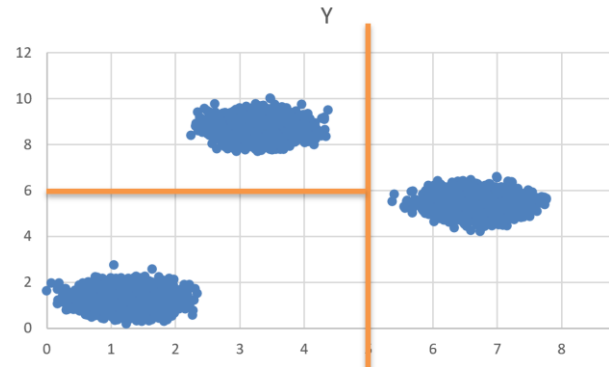


ОТВЕТ:
10738 30730
37522 51277

ФАЙЛ А: разделение по $x = 1$



ФАЙЛ В: $x > 5; y > 6$



```
from math import dist

clustersA = [[], []] # собираем кластеры из A
for s in open("27A.txt"):
    x, y = map(float, s.split())
    if x < 1: clustersA[0].append([x, y])
    else: clustersA[1].append([x, y])

clustersB = [[], [], []]
for s in open("27B.txt"): # собираем кластеры из B
    x, y = map(float, s.split())
    if x > 5: clustersB[0].append([x, y])
    elif y > 6: clustersB[1].append([x, y])
    else: clustersB[2].append([x, y])

def center(kl): # функция центра
    cords = []
    mini = 10**10
    for p in kl:
        sm = 0
        for p1 in kl:
            sm += dist(p, p1)
        if sm < mini:
            mini = sm
            cords = p
    return cords

centersA = [center(kl) for kl in clustersA]
centersB = [center(kl) for kl in clustersB]

px1 = sum(p[0] for p in centersA) / 2 * 10000
py1 = sum(p[1] for p in centersA) / 2 * 10000

px2 = sum(p[0] for p in centersB) / 3 * 10000
py2 = sum(p[1] for p in centersB) / 3 * 10000

print(int(px1), int(py1))
print(int(px2), int(py2))
```