

## Задачи первого дня ЕГЭ по информатике.

### Содержание

1	Задача 1	4
2	Задача 2	5
3	Задача 3	6
4	Задача 4	7
5	Задача 5	8
6	Задача 6	9
7	Задача 7	11
8	Задача 8.1	11
9	Задача 8.2	12
10	Задача 9	13
11	Задача 10	15
12	Задача 11.1	15
13	Задача 11.2	16
14	Задача 12	16
15	Задача 13	18
16	Задача 14.1	19

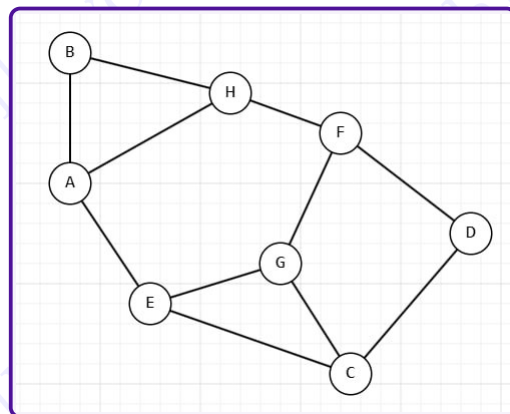
17	Задача 14.2	20
18	Задача 15	21
19	Задача 16	22
20	Задача 17.1	22
21	Задача 17.2	23
22	Задача 18	24
23	Задача 19.1	27
24	Задача 20.1	28
25	Задача 21.1	28
26	Задача 19.2	29
27	Задача 20.2	30
28	Задача 21.2	31
29	Задача 22.1	32
30	Задача 22.2	33
31	Задача 23	34
32	Задача 24.1	35
33	Задача 24.2	36
34	Задача 25	37
35	Задача 26.1	38
36	Задача 26.2	43

37	Задача 26.3	46
38	Задача 26.4	48
39	Задача 26.5	50
40	Задача 27.1	52
41	Задача 27.2	56

## 1 Задача 1

На рисунке схема дорог М-ского района изображена в виде графа, в таблице содержатся сведения о протяжённости каждой из этих дорог (в километрах). 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 протяжённостей дорог из пункта В в пункт Н и из пункта А в пункт Е. В ответе запишите целое число.

	П1	П2	П3	П4	П5	П6	П7	П8
П1		53		1			2	
П2	53			13				8
П3					30		39	74
П4	1	13				5		
П5			30					2
П6				5			21	
П7	2		39			21		
П8		8	74		3			



### Решение

Нам нужно найти уникальную вершину второй степени, которая ведёт в две вершины степени 3, соединяющиеся между собой. Такой уникальной вершиной является В. Значит вершина В это П5. Из вершины В ведут две дороги - П3 и П8. Выясним, какая из этих вершин ведёт в вершину степени 3, идущую в вершину степени 2. Вершина П3 идёт в вершину П7, идущую в вершину П6 второй степени. Таким образом: Н - П3, А - П8, F - П7, D - П6, Е - П2, G - П4, С - П1. Значит путь из А в Е равен 8, а путь из В в Н равен 30. Их сумма равна 38.

**Ответ: 38**

## 2 Задача 2

Максим заполнял таблицу истинности логической функции

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

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

				F
		0	1	0
	1		0	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.

### Решение

---

```
for x in range(2):
    for y in range(2):
        for z in range(2):
            for w in range(2):
                if (z or (z == w) or (not(y <= x))) == False:
                    print(x, y, z, w)
```

---

Получаем:

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

В исходной таблице первый столбец полностью свободен – значит, он соответствует столбцу  $w$ . Аналогично, только в третьем столбце исходной таблицы нет единиц – это столбец  $z$ .

Таблица сейчас выглядит так:

$w$	?	$z$	?
1	?	0	1
1	1	0	0
1	0	0	?

Смотрим по строкам: вторая строка соответствует второй из вывода, первая строка может соответствовать только третьей из вывода, тогда третья только первой из вывода. В выводе у переменной  $x$  один 0, две 1, а у переменной  $y$  два 0, одна 1 → второму столбцу соответствует переменная  $x$ , четвертому переменная  $y$ .

**Ответ:**  $wxyz$

### 3 Задача 3

В файле `3_03.xlsx` приведен фрагмент базы данных «Хозтовары» о поставках товаров для ухода, уборки и дома. База данных состоит из трёх таблиц. Таблица «Движение товаров» содержит записи о поставках товаров в магазины в течение сентября 2021 г., а также информацию о проданных товарах. Поле Тип операции содержит значение Поступление или Продажа, а в соответствующее поле Количество упаковок, шт внесена информация о том, сколько упаковок товара поступило в магазин или было продано в течение дня. Заголовок таблицы имеет следующий вид. Используя информацию из приведённой базы данных, определите общую сумму выручки с продажи всех видов шампуня для волос, проданных магазинами, расположенными на ул. Metallургов, за период с 1 по 23 сентября включительно. В ответе запишите только число - сумму выручки.

## Решение

Для начала перенесём все нужные данные на лист «Движение товаров». С помощью  $ID$  магазина определим его адрес: в ячейку  $H2$  запишем формулу

$$=ВПР(C2;Магазин!A:C;3;0)$$

с помощью артикула определяем название товара: в ячейку  $I2$  записываем формулу

$$=ВПР(D2;Товар!A:F;3;0)$$

Для поиска суммы выручки нам нужно будет знать стоимость для количества упаковок, которое записано в столбике  $E$ , поэтому в ячейке  $J2$  записываем формулу

$$=E2*G2$$

Теперь осталось расставить фильтры: среди адресов магазинов выбираем только те, где есть «ул. Metallургов», в датах оставляем только 1-23 сентября (включительно) и в столбике «Тип операции» оставляем только «Продажа», в столбике с названиями товаров оставляем только те, в которых есть «шампунь для волос». Для оставшихся строчек смотрим сумму чисел в столбике  $J$

## 4 Задача 4

Для кодирования некоторой последовательности, состоящей из букв Л, И, Ц, Я, А, Т, Н решили использовать неравномерный двоичный код, удовлетворяющий условию Фано. Для буквы Н использовали кодовое слово 10, для буквы К – кодовое слово 110. Какова наименьшая возможная суммарная длина слова «ЛАТИНИЦА».

## Решение

Для наименьшей возможной суммарной длины слова при распределении кодов букв будем учитывать их частоту в слове. Очевидно, что буквы А и И должны иметь наименьшую длину. Продлим ветку 0 надвое. На одну из веток поставим букву А - 01, вторую ветку продлим и разместим две буквы: И - 000, Л - 001.

Ветку 110 продлим надвое и разместим букву Т - 1100, вторую ветку продлим и разместим две буквы: Ц - 11010 и Я - 11011.

Тогда суммарная длина слова будет:  $3 + 2 + 4 + 3 + 2 + 3 + 5 + 2 = 24$ .

**Ответ: 24**

## 5 Задача 5

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

1. Строится троичная запись числа  $N$ .
2. Далее эта запись обрабатывается по следующему правилу:
  - а) если число  $N$  делится на 3, то к этой записи дописывается справа две последние цифры троичной записи;
  - б) если число  $N$  не делится на 3, то остаток от деления умножается на 5, переводится в троичную запись и дописывается в конец числа.

Полученная таким образом запись является троичной записью возможного числа  $R$ .

3. Результат переводится в десятичную систему и выводится на экран.

Укажите наименьшее число  $N$ , после обработки которого с помощью этого алгоритма получается число  $R$ , не меньшее чем 290. В ответе запишите это число в десятичной системе счисления.

### Решение

---

```
def f3(n):
    s = ''
    while n > 0:
        s += str(n%3)
        n //=3
    return s[::-1]
for n in range (3, 1000):
    s = f3(n)
    if n%3 == 0: s += s[-2:]
    else: s += f3((n%3)*5)
    r = int(s,3)
    if r >= 290:
        print(n)
        break
```

---

Ответ: 11

## 6 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  $t$**  (где  $t$  – целое число), вызывающая изменение направления движения на  $t$  градусов по часовой стрелке, **Налево  $t$**  (где  $t$  – целое число), вызывающая изменение направления движения на  $t$  градусов против часовой стрелки. Запись **Повтори  $k$  [Команда1 Команда2 ... Команда $S$ ]** означает, что последовательность из  $S$  команд повторится  $k$  раз.

Черепахе был дан для исполнения следующий алгоритм:

Повтори 2 [Вперёд 24 Направо 90 Вперёд 20 Направо 90]

Поднять хвост

Вперёд 7 Направо 90 Вперёд 7 Налево 90

Опустить хвост

Повтори 2 [Вперёд 30 Направо 90 Вперёд 27 Направо 90]

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

## Решение

---

```
from turtle import *
left(90)
k = 20
tracer(0)
screensize(2000, 2000)
pd()
for i in range(2):
    forward(24*k)
    right(90)
    forward(20*k)
    right(90)
pu()
forward(7*k)
right(90)
forward(7*k)
left(90)
pd()
for j in range(2):
    forward(30*k)
    right(90)
    forward(27*k)
    right(90)
pu()
for x in range(-100,100):
    for y in range(-100,100):
        goto(x*k, y*k)
        dot(3)
done()
```

---

**Ответ: 1141**

## 7 Задача 7

Виталий фотографирует интересные места и события с помощью своего смартфона. Каждая фотография представляет собой растровое изображение размером  $1920 \times 1080$  пикселей, при этом используется палитра из  $2^{22}$  цветов.

В конце дня Виталий отправляет снимки друзьям с помощью приложения-мессенджера. Для экономии трафика приложение оцифровывает снимки повторно, используя размер  $1280 \times 1024$  пикселей и глубину цвета 21 бит.

Сколько Кбайт трафика экономится при передаче 120 фотографий? В ответе укажите целую часть полученного числа.

### Решение

Объем  $I_1$  фотографий до сжатия:  $1920 * 1080 * 22 * 120$  бит.

Объем  $I_2$  фотографий после сжатия:  $1280 * 1024 * 21 * 120$  бит.

Экономия килобайт составит:  $\frac{I_1 - I_2}{8 * 1024} = 265050$  Кбайт.

**Ответ: 265050**

## 8 Задача 8.1

Все пятибуквенные слова, составленные из букв В, Е, Н, Е, Р, А записаны в алфавитном порядке и пронумерованы.

Вот начало списка:

1. ААААА
2. ААААВ
3. ААААЕ
4. ААААН
5. ААААР

Под каким нечетным номером в списке стоит последнее слово, которое не начинается с буквы Н и содержит ровно две буквы В?

## Решение

---

```
from itertools import *
number = 1
for s in product(sorted("БЕHPA"), repeat=5):
    x = "".join(s)
    if (x[0] != "H") and (x.count("B") == 2) and (number % 2 == 1):
        print(x, number)
    number += 1
```

---

Ответ: 3107

## 9 Задача 8.2

Все шестибуквенные слова, составленные из букв Т, Е, О, Р, И, Я, записаны в алфавитном порядке и пронумерованы.

Вот начало списка:

1. EEEEEЕ
2. EEEEEИ
3. EEEEEО
4. EEEEEР
5. EEEEEТ

Определите, под каким номером в этом списке стоит последнее слово с нечётным номером, которое не начинается с букв Е, И или О и при этом содержит в своей записи ровно одну букву Т.

## Решение

---

```
from itertools import product
cnt = 1, ans = -1
for word in product('ЕИОРТЯ', repeat=6):
    s = ''.join(word)
    if cnt % 2 != 0 and s[0] not in 'ЕИО' and s.count('Т') == 1:
        ans = cnt
    cnt += 1
print(ans)
```

---

Ответ: 46655

## 10 Задача 9

Откройте файл электронной таблицы, содержащей в каждой строке шесть натуральных чисел. Определите среднее арифметическое чисел всех строк таблицы, содержащих числа, для которых выполнены оба условия:

- в строке есть одно число, которое повторяется трижды, остальные три числа различны;
- повторяющееся число строки меньше, чем удвоенное минимальное из её повторяющихся чисел.

В ответе запишите только целую часть полученного числа.

### Решение программой

---

```
f = open("9.txt")
res = []
for i in f:
    a = [int(x) for x in i.split(";")]
    p3 = [x for x in a if a.count(x) == 3]
    np = [x for x in a if a.count(x) == 1]
    if len(p3) == 3 and len(np) == 3:
        if p3[0] < min(np) * 2:
            res += a
print(sum(res) // len(res))
```

---

## Решение через Excel

В ячейку G1 запишем формулу

$$=СЧЁТЕСЛИ(A1:F1;A1)$$

и растянем вправо для каждого числа, чтобы узнать количество повторов каждого числа

В ячейку M1 записываем формулу

$$=ЕСЛИ(И(СЧЁТЕСЛИ(G1:L1;1)=3;СЧЁТЕСЛИ(G1:L1;3)=3);1;0)$$

для проверки первого условия

Для проверки второго условия нам нужно узнать число, которое в строке повторяется, используем формулу в ячейке N1:

$$=СУММЕСЛИ(G1:L1;3;A1:F1)/3$$

В ячейке O1 пишем формулу

$$=ЕСЛИ(A1=\$N1;100000000000000;A1)$$

и растягиваем для каждого числа, чтобы дальше МИН() выбрал минимальное число только среди неповторяющихся чисел

В ячейке V1 используем формулу

$$=ЕСЛИ(N1<U1*2;1;0)$$

для проверки второго условия

В ячейке W1 проверяем, выполнены ли оба условия

$$=M1*V1$$

Для ответа нужно найти среднее арифметическое чисел строчек, которые подошли под условие, поэтому в ячейку X1 запишем формулу

$$=ЕСЛИ(W1=1;СУММ(A1:F1);)$$

В ячейке Y1 получим ответ через формулу

$$=СУММ(X:X)/СЧЁТ(X:X)$$

**Ответ: 78**

## 11 Задача 10

В приведенном файле необходимо посчитать, сколько раз встречается слово «сам» в составе других слов, но не как отдельное слово.

### Решение

Нажмем на поиск и в расширенном варианте уберем галочку с параметра «только слово целиком» и поставим на «не учитывать регистр», вводим в поле «сам».

После этого ставим галочку на параметр «только слово целиком» и снова нажимаем поиск. Тогда, чтобы получить «сам» только в составе других слов, а не как отдельное слово, нужно вычесть из первого второе и получить ответ.

## 12 Задача 11.1

Серийный номер состоит из 3410 символов. Для хранения каждого из серийных номеров выделено одинаковое минимальное число байт. Для хранения 2984523 серийных номеров потребовалось не менее 14 Гбайт. Найдите минимально возможную мощность алфавита для записи одного серийного номера.

### Решение

Общий объем памяти: 14 Гбайт =  $14 * 1024^3 = 15032385536$  байт.

Для 1 серийного номера нужно:  $\frac{15032385536}{2984523} \approx 5036,78$  байт.

Поскольку выделяется целое число байт, берем 5037 байт.

Для хранения одного символа нужно:  $\frac{5037 * 8}{3410} \approx 11,82$  бита.

Поскольку символ кодируется целым числом бит, берем 12 бит.

Проверим достаточность 11 бит: при 11 битах один номер займет:

$$\frac{3410 * 11}{8} = 4688,75 \rightarrow 4689 \text{ байт}$$

Для всех номеров:  $4689 * 2984523 \approx 13,03$  Гбайт < 14 Гбайт – недостаточно.

При 12 битах: один номер займет  $\frac{3410 * 12}{8} = 5115$  байт. Для всех номеров:  $5115 * 2984523 \approx 14,22$  Гбайт > 14 Гбайт – достаточно.

Минимальная мощность алфавита при 12 битах:  $2^{11} + 1 = 2049$  символов.

**Ответ: 2049**

## 13 Задача 11.2

На предприятии каждой изготовленной детали присваивается серийный номер, состоящий из 248 символов. В базе данных для хранения каждого серийного номера отведено одинаковое и минимальное возможное количество байт. При этом используется посимвольное кодирование номеров, все символы кодируются одинаковым и минимально возможным количеством бит. Известно, что для хранения 75 600 номеров требуется не менее 16 Мбайт памяти. Определите минимально возможную мощность алфавита, используемого для записи серийных номеров. В ответе запишите только целое число.

### Решение

Пусть  $i$  – количество бит для кодирования одного символа.

Составим неравенство:  $\frac{248 \times i \times 75600}{8} \geq 16 \times 1024^2$

$$16 * 1024^2 = 16777216 \text{ байт}$$

$$18748800 * i \geq 134217728$$

$$i \geq 7,16$$

Минимальное целое значение:  $i = 8$  бит.

Проверим достаточность 7 бит: объем =  $\frac{248 \times 7 \times 75600}{8} = 16399200$  байт  $\approx 15,64$  Мбайт < 16 Мбайт – недостаточно.

При 8 битах: объем =  $\frac{248 \times 8 \times 75600}{8} = 18748800$  байт  $\approx 17,88$  Мбайт > 16 Мбайт – достаточно.

Минимальная мощность алфавита при 8 битах:  $2^7 + 1 = 129$  символов.

**Ответ: 129**

## 14 Задача 12

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

Б) **нашлось** (v).

Эта команда проверяет, встречается ли цепочка 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.

Дана программа для Редактора:

НАЧАЛО

ПОКА нашлось (31) ИЛИ нашлось (11111) ИЛИ нашлось (144)

ЕСЛИ нашлось (31)

ТО заменить (31, 4)

КОНЕЦ ЕСЛИ

ЕСЛИ нашлось (11111)

ТО заменить (11111, 33)

КОНЕЦ ЕСЛИ

ЕСЛИ нашлось (144)

ТО заменить (144, 133)

КОНЕЦ ЕСЛИ

КОНЕЦ ПОКА

КОНЕЦ

На вход приведённой выше программе поступает строка, начинающаяся с цифры «3», а затем содержащая n цифр «1» ( $5 < n < 1000$ ).

Определите значение n, при котором сумма цифр в строке, получившейся в результате выполнения программы, равна 160.

**Решение**

---

```
for n in range(6, 1000):
    s = '3'+n*'1'
    while '31' in s or '11111' in s or '144' in s:
        s = s.replace('31', '4', 1)
        s = s.replace('11111', '33', 1)
        s = s.replace('144', '133', 1)
    if sum(map(int, s)) == 160:
        print(n)
```

---

**Ответ: 135**

## 15 Задача 13

В терминологии сетей TCP/IP маска сети – это двоичное число, меньшее  $2^{32}$ ; в маске сначала (в старших разрядах) стоят единицы, а затем с некоторого места нули. Маска определяет, какая часть IP-адреса узла сети относится к адресу сети, а какая – к адресу самого узла в этой сети. Обычно маска записывается по тем же правилам, что и IP-адрес – в виде четырех байт, причем каждый байт записывается в виде десятичного числа. Адрес сети получается в результате применения поразрядной конъюнкции к заданному IP-адресу узла и маске.

Например, если IP-адрес узла равен 131.32.255.131, а маска равна 255.255.240.0, то адрес сети равен 131.32.240.0.

Сеть задана IP-адресом одного из входящих в неё узлов 186.215.243.5 и сетевой маской 255.255.192.0.

Найдите наибольший IP-адрес в данной сети, который может быть назначен компьютеру.

В ответе укажите найденный IP-адрес в десятичной системе счисления без точек и разделителей.

### Решение

Переведём IP-адрес узла и маску сети в двоичную систему счисления:

IP узла	10111010.11010111.11110011.00000101
IP маска	11111111.11111111.11000000.00000000

Узлы в сети имеют вид: 10111010.11010111.11xxxxxx.xxxxxx

Нам нужно найти наибольший IP-адрес в данной сети, который может быть назначен компьютеру. Но мы не можем назначать компьютеру широковещательный адрес.

Поэтому, последний возможный IP-адрес будет равен:

$$10111010.11010111.11111111.11111110_2 = 186.215.255.254_{10}$$

## Решение программой

---

```
from ipaddress import *
# Создаем объект сети
net = ip_network('186.215.243.5/255.255.192.0', 0)
# Берем предпоследний адрес, так как последний - широковещательный
print(str(net[-2]).replace('.', ''))
```

---

Ответ: 186215255254

### 16 Задача 14.1

Операнды арифметического выражения записаны в системе счисления с основанием 27.

$$2107x792_{27} + 565x211_{27}$$

В записи чисел переменной  $x$  обозначена неизвестная цифра из алфавита 27-ричной системы счисления. Определите наименьшее 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.

### Решение

---

```
for x in range(0, 27):
    a1 = 2 * 27 ** 7 + 1 * 27 ** 6 + 7 * 27 ** 4
    + x * 27 ** 3 + 7 * 27 ** 2 + 9 * 27 ** 1 + 2
    a2 = 5 * 27 ** 6 + 6 * 27 ** 5 + 5 * 27 ** 4
    + x * 27 ** 3 + 2 * 27 ** 2 + 1 * 27 + 1
    a = a1 + a2
    if a % 26 == 0:
        print(a//26)
        break
```

---

Ответ: 897607140

## 17 Задача 14.2

Значение арифметического выражения  $7^{150} - 7^{100} - x$ , где  $x$  - целое положительное число, не превышающее 2030, записали в пятеричной системе счисления. Определите наибольшее значение  $x$ , при котором в пятеричной записи числа, являющегося значением арифметического выражения, содержится максимальное количество нулей.

---

```
def ss(n):
    res = ""
    while n > 0:
        res = str(n % 5) + res
        n //= 5
    return res
mx = 0
ans = 0
for x in range(2030 + 1):
    s = 7 ** 150 - 7 ** 100 - x
    nulls = ss(s)
    if nulls.count("0") >= mx:
        mx = nulls.count("0")
        ans = x
print(ans)
```

---

**Ответ: 623**

## 18 Задача 15

Для какого наибольшего целого неотрицательного числа  $A$  выражение

$$(x + 3y > A) \vee (x < 18) \vee (y < 33)$$

тождественно истинно при любых целых неотрицательных  $x$  и  $y$ ?

**Решение**

---

```
def f(a):
    for x in range(1000):
        for y in range(1000):
            if not ((x + 3 * y > a) or (x < 18) or (y < 33)):
                return False
    return True
for a in range(1000, 0, -1):
    if f(a):
        print(a)
```

---

**Ответ: 116**

## 19 Задача 16

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

$$F(n) = 1 \text{ при } n = 1;$$

$$F(n) = n^2 + F(n - 1) \text{ при } n > 1.$$

Чему равно значение выражения  $F(2025) - F(2022)$ ?

### Решение

---

```
from functools import lru_cache
@lru_cache (None)
def f(n) :
    if n == 1:
        return 1
    else:
        return n ** 2 + f(n - 1):
for i in range (1, 2026):
    f(i)
print(f(2025) - f(2022))
```

---

**Ответ: 12289730**

## 20 Задача 17.1

В файле 17.14.txt содержится последовательность целых чисел от -100 000 до 100 000 включительно. Определите количество троек элементов последовательности, в которых из трёх элементов тройки пятизначными числами являются только два, а сумма элементов тройки не больше максимального элемента последовательности, оканчивающегося на 13. В ответе запишите кол-во найденных троек чисел, затем максимальную из сумм элементов таких троек. Под тройкой подразумевается три идущих подряд элемента последовательности.

## Решение

---

```
f = open('...')
a = [int(x) for x in f.readlines()]
max_13 = max([x for x in a if abs(x) % 100 == 13])
cnt = 0 # количество подходящих троек
max_sum = -10000 # максимальная из сумм троек
for i in range(len(a) - 2):
    n = a[i:i + 3] # список из чисел тройки
    t = [10_000 <= abs(x) < 100_000 for x in n] # проверяем пятизначность
    if len(t) == 2 and sum(n) <= max_13:
        cnt += 1, max_sum = max(max_sum, sum(n))
print(cnt, max_sum)
```

---

## 21 Задача 17.2

В файле 17.txt содержится последовательность целых чисел от -100 000 до 100 000 включительно. Определите количество пар элементов последовательности, в которых только один элемент является двузначным, а сумма элементов больше максимального элемента последовательности, оканчивающегося на 19. В ответе запишите без пробела кол-во найденных пар чисел, затем максимальную из их сумм. Под парой подразумевается два идущих подряд элемента последовательности.

## Решение

---

```
f = open("17.14.txt")
a = [int(x) for x in f]
res = []
max_19 = max([x for x in a if abs(x) % 100 == 19])
for i in range (len(a) -1):
    if (int(len(str(abs(a[i]))) == 2) != int(len(str(abs(a[i + 1]))) == 2))
        and (sum(a[i:i + 2]) <= max_19):
        res.append(a[i] + a[i + 1])
print(len(res), max(res))
```

---

## 22 Задача 18

Исполнитель Робот стоит в левом верхнем углу поля, разлинованного на клетки. Он может перемещаться по клеткам, выполняя за одно перемещение одну из двух команд: **вправо** или **вверх**. По команде вправо Робот перемещается в соседнюю правую клетку; по команде вниз – в соседнюю нижнюю. 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 сверху ограничены стенами, Робот не может продолжать движение, поэтому накопленная сумма считается итоговой. Таких конечных клеток на поле может быть несколько, включая правую верхнюю 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 нижней клетки в конечную клетку маршрута.

Исходные данные записаны в файле в виде электронной таблицы, каждая ячейка которой соответствует клетке поля. В ответе запишите два числа – сначала максимальную сумму, которую может собрать Робот, затем – минимальную.

### Решение

Переносим форматирование, вставляем исходное значение.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
1	77	68	28	32	50	17	89	22	48	42	13	43	92	9	95	24	56	30	82	78	
2	12	5	10	77	93	20	35	44	49	28	34	12	8	6	35	98	54	46	87	56	
3	100	90	54	41	96	1	8	81	100	55	13	25	2	37	47	42	48	58	70	51	
4	26	38	29	87	8	93	46	21	47	21	56	6	81	58	79	40	15	16	35	87	
5	93	82	73	13	58	58	29	14	62	73	48	92	49	71	12	49	73	43	52	68	
6	50	20	97	3	51	20	16	57	70	10	18	76	86	41	74	48	11	9	2	85	
7	49	34	97	60	41	93	57	14	70	85	4	56	97	48	39	94	72	94	64	1	
8	84	79	63	10	60	62	5	70	49	65	57	17	13	99	68	64	72	84	36	38	
9	63	6	28	26	44	19	76	88	66	78	1	88	98	68	90	12	22	3	96	27	
10	68	38	44	11	28	100	4	73	27	5	52	7	94	92	39	82	59	94	30	48	
11	74	23	42	7	95	95	43	43	2	21	20	31	18	54	54	22	26	18	75	5	
12	2	97	97	24	11	54	49	78	46	15	18	2	61	33	68	10	42	31	26	27	
13	79	59	43	88	29	56	64	43	89	28	7	16	54	77	81	79	25	36	20	86	
14	87	71	3	60	17	2	9	26	64	3	81	61	38	51	43	47	86	50	49	91	
15	81	37	3	30	80	47	17	2	23	83	15	26	88	93	84	6	45	31	15	53	
16	86	37	7	41	35	68	11	36	58	58	5	90	10	63	30	4	55	78	32	87	
17	91	54	11	98	73	95	1	35	79	68	73	57	67	26	76	20	43	62	49	90	
18	79	21	59	62	57	98	21	68	47	76	66	28	46	3	70	12	2	40	90	56	
19	55	44	59	56	66	62	81	94	17	62	38	86	67	64	1	21	42	86	27	73	
20	43	24	49	89	5	58	88	100	19	82	55	43	33	80	85	23	31	32	12	38	
21																					
22																					
23																					
24																					
25																					
26																					
27																					
28																					
29																					
30																					
31																					
32																					
33																					
34																					
35																					
36																					
37																					
38																					
39																					
40																					
41																					
42																					
43	43																				
44																					

В левый столбец динамически к сумме прибавляем значение из соответствующей ячейки

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
1	77	68	26	32	50	17	69	22	48	42	13	43	92	9	95	24	56	90	82	78
2	12	5	10	77	93	20	35	44	49	28	34	12	8	6	35	98	54	46	87	56
3	100	90	54	41	96	1	8	81	100	55	13	25	2	37	47	42	48	58	70	51
4	26	38	29	67	6	93	46	21	47	21	56	6	81	58	79	40	15	16	35	87
5	93	82	73	13	58	58	29	14	62	73	48	92	49	71	12	49	73	43	52	68
6	50	20	97	3	51	20	16	57	70	10	18	76	86	41	74	48	12	9	2	85
7	49	34	97	60	41	93	57	14	70	85	4	56	97	48	39	94	72	94	64	1
8	84	79	63	10	60	62	5	70	49	65	57	17	13	99	68	64	72	84	36	38
9	63	6	28	26	44	19	76	88	66	78	1	88	98	68	90	12	22	3	96	27
10	68	38	44	11	28	100	4	73	27	5	52	7	94	92	39	82	59	94	30	48
11	74	23	42	7	95	63	45	2	21	20	31	18	54	54	22	26	19	75	5	30
12	2	97	97	24	11	54	49	78	46	15	18	2	61	33	68	19	42	31	26	27
13	79	59	43	88	29	56	64	43	89	28	7	16	54	77	81	79	25	36	20	86
14	87	71	3	60	17	2	9	26	64	3	81	61	38	51	43	47	86	50	49	91
15	81	37	3	30	80	47	17	2	23	83	15	26	88	93	84	6	45	31	15	53
16	86	37	7	41	35	68	11	36	58	58	5	90	10	63	30	4	55	78	32	87
17	91	54	11	98	73	95	1	75	79	68	73	57	67	24	76	20	43	62	49	90
18	79	21	59	62	57	98	21	68	47	76	66	28	46	3	70	12	2	40	90	56
19	55	44	59	56	66	62	81	94	17	62	38	86	67	64	1	21	42	86	27	73
20	43	24	49	89	5	58	88	100	19	82	55	43	33	89	85	23	31	32	12	38
21																				
22																				
23	1299																			
24	1222																			
25	1210																			
26	1110																			
27	1084																			
28	991																			
29	941																			
30	892																			
31	808																			
32	745																			
33	677																			
34	603																			
35	601																			
36	522																			
37	435																			
38	354																			
39	268																			
40	177																			
41	=A42+A19																			
42	43																			
43																				
44																				

Для подсчета максимума берем максимум из двух сумм, прибавляем значение из соответствующей ячейки, растягиваем на всю таблицу.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
1	77	68	26	32	50	17	69	22	48	42	13	43	92	9	95	24	56	90	82	78
2	12	5	10	77	93	20	35	44	49	28	34	12	8	6	35	98	54	46	87	56
3	100	90	54	41	96	1	8	81	100	55	13	25	2	37	47	42	48	58	70	51
4	26	38	29	67	6	93	46	21	47	21	56	6	81	58	79	40	15	16	35	87
5	93	82	73	13	58	58	29	14	62	73	48	92	49	71	12	49	73	43	52	68
6	50	20	97	3	51	20	16	57	70	10	18	76	86	41	74	48	12	9	2	85
7	49	34	97	60	41	93	57	14	70	85	4	56	97	48	39	94	72	94	64	1
8	84	79	63	10	60	62	5	70	49	65	57	17	13	99	68	64	72	84	36	38
9	63	6	28	26	44	19	76	88	66	78	1	88	98	68	90	12	22	3	96	27
10	68	38	44	11	28	100	4	73	27	5	52	7	94	92	39	82	59	94	30	48
11	74	23	42	7	95	63	45	2	21	20	31	18	54	54	22	26	19	75	5	30
12	2	97	97	24	11	54	49	78	46	15	18	2	61	33	68	19	42	31	26	27
13	79	59	43	88	29	56	64	43	89	28	7	16	54	77	81	79	25	36	20	86
14	87	71	3	60	17	2	9	26	64	3	81	61	38	51	43	47	86	50	49	91
15	81	37	3	30	80	47	17	2	23	83	15	26	88	93	84	6	45	31	15	53
16	86	37	7	41	35	68	11	36	58	58	5	90	10	63	30	4	55	78	32	87
17	91	54	11	98	73	95	1	75	79	68	73	57	67	24	76	20	43	62	49	90
18	79	21	59	62	57	98	21	68	47	76	66	28	46	3	70	12	2	40	90	56
19	55	44	59	56	66	62	81	94	17	62	38	86	67	64	1	21	42	86	27	73
20	43	24	49	89	5	58	88	100	19	82	55	43	33	89	85	23	31	32	12	38
21																				
22																				
23	1299																			
24	1222																			
25	1210																			
26	1110																			
27	1084																			
28	991																			
29	941																			
30	892																			
31	808																			
32	745																			
33	677																			
34	603																			
35	601																			
36	522																			
37	435																			
38	354																			
39	268																			
40	177																			
41	98	67																		
42	43	67																		
43																				
44																				



## 23 Задача 19.1

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 один камень или убрать из кучи три камня. Например, имея кучу из 30 камней, за один ход можно получить кучу из 29 или 27 камней. Игра завершается в тот момент, когда количество камней в куче становится не более 11.

Победителем считается игрок, сделавший последний ход, то есть первым получивший позицию, в которой в куче будет 11 или менее камней. В начальный момент в куче было  $S$  камней,  $S > 11$ .

Будем говорить, что игрок имеет выигрышную стратегию, если он может выиграть при любых ходах противника. Описать стратегию игрока — значит описать, какой ход он должен сделать в любой ситуации, которая ему может встретиться при различной игре противника. В описание выигрышной стратегии не следует включать ходы играющего по этой стратегии игрока, не являющиеся для него безусловно выигрышными, т. е. не являющиеся выигрышными независимо от игры противника.

Известно, что Ваня выиграл своим первым ходом после неудачного первого хода Пети. Укажите максимальное значение  $S$ , когда такая ситуация возможна.

### Решение

---

```
from functools import lru_cache
@lru_cache(None)
def game(x):
    if x <= 11: # условие выигрыша
        return 0
    moves = [game(x - 1), game(x - 3)] # перечисляем ходы
    n = [i for i in moves if i <= 0]
    if n: return -max(n) + 1
    return -max(moves)
for i in range(12, 100):
    if game(i - 1) == 1 or game(i - 3) == 1: # Если Петя поддаётся
        print(i)
```

---

Ответ: 17

## 24 Задача 20.1

Для игры, описанной в предыдущем задании, найдите два таких значения  $S$ , при котором у Петя 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 препинания.

### Решение

---

```
from functools import lru_cache
@lru_cache(None)
def game(x):
    if x <= 11: # условие выигрыша
        return 0
    moves = [game(x - 1), game(x - 3)] # перечисляем ходы
    n = [i for i in moves if i <= 0]
    if n:
        return -max(n) + 1
    return -max(moves)
for i in range(12, 100):
    if game(i) == 2:
        print(i)
```

---

Ответ: 16 18

## 25 Задача 21.1

Для игры, описанной ранее, найдите такое максимальное 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 такого значения нет, в ответ запишите 0.

## Решение

---

```
from functools import lru_cache
@lru_cache(None)
def game(x):
    if x <= 11: # условие выигрыша
        return 0
    moves = [game(x - 1), game(x - 3)] # перечисляем ходы
    n = [i for i in moves if i <= 0]
    if n:
        return -max(n) + 1
    return -max(moves)
for i in range(12, 100):
    if game(i) == -2:
        print(i)
```

---

Ответ: 19

### 26 Задача 19.2

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 6 камней;
- уменьшить количество камней в куче в 3 раза (количество камней, полученное при делении, округляется до меньшего).

Например, из кучи в 20 камней за один ход можно получить кучу из 17, 14 или 6 камней.

Игра завершается, когда количество камней в куче становится не более 25. Победителем считается игрок, сделавший последний ход, то есть первым получивший кучу из 25 или меньше камней. В начальный момент в куче было  $S$  камней,  $S \geq 26$ .

Укажите минимальное значение  $S$ , при котором Петя не может выиграть за один ход, но при любом ходе Пети Ваня Будем говорить, что игрок имеет выигрышную стратегию, если он может выиграть при любых ходах противника. может выиграть своим первым ходом.

## Решение

---

```
from functools import lru_cache
@lru_cache(None)

def game(x):
    if x <= 25: # условие выигрыша
        return 0
    moves = [game(x - 3), game(x - 6)] # перечисляем ходы
    if x > 0:
        moves.append(game(x // 3))
    n = [i for i in moves if i <= 0]
    if n:
        return -max(n) + 1
    return -max(moves)

for i in range(26, 100):
    if game(i) == -1:
        print(i)
```

---

Ответ: 78

### 27 Задача 20.2

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.

## Решение

---

```
from functools import lru_cache
@lru_cache(None)

def game(x):
    if x <= 25: # условие выигрыша
        return 0
    moves = [game(x - 3), game(x - 6)] # перечисляем ходы
    if x > 0:
        moves.append(game(x // 3))
    n = [i for i in moves if i <= 0]
    if n:
        return -max(n) + 1
    return -max(moves)

for i in range(26, 100):
    if game(i) == 2:
        print(i)
```

---

Ответ: 81 82

### 28 Задача 21.2

Для игры, описанной в задании 19, найдите минимальное значение  $S$ , при котором одновременно выполняются 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.

## Решение

---

```
from functools import lru_cache
@lru_cache(None)
def game(x):
    if x <= 25: # условие выигрыша
        return 0
    moves = [game(x - 3), game(x - 6)] # перечисляем ходы
    if x > 0:
        moves.append(game(x // 3))
    n = [i for i in moves if i <= 0]
    if n:
        return -max(n) + 1
    return -max(moves)
for i in range(26, 100):
    if game(i) == -2:
        print(i)
```

---

Ответ: 87

### 29 Задача 22.1

В файле 22.xls содержится информация о совокупности 25 вычислительных процессов, которые могут выполняться параллельно или последовательно. Будем говорить, что процесс В зависит от процесса А, если для выполнения процесса В необходимы результаты выполнения процесса А. 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, сколько процессов выполнялось одновременно в 14 мс, если время начала каждого процесса минимально.

*В 1-й день практически у всех был такой прототип, только с разными мс.*

## Решение

Для независимых процессов, начиная со столбца D, будем проставлять единицы вправо. Количество единиц равно количеству мс выполнения каждого процесса. Для удобства выберем ширину столбцов 2-3.

Затем переходим к зависимым процессам и начинаем также проставлять единицы, но уже после выполнения независимых процессов.

В последней после таблицы строке столбца D прописываем формулу СУММ(D2:D13) и растягиваем её вправо. Полученные значения – количество выполняемых одновременно процессов. Остается посчитать сколько процессов выполнялось одновременно в 14 мс.

## 30 Задача 22.2

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.

## Решение

Выделим столбец C, перейдём во вкладку Данные, раздел «Текст по столбцам» и разделим наши данные, указав символом-разделителем точку с запятой.

В ячейку O2 впишем формулу:  $=B2+МАКС(I2 : N2)$  и растянем её вниз.

В ячейку I2 впишем формулу:  $=ВПР(C2;A:O;15;0)$  и растянем её вправо до столбца N, вниз до конца таблицы. В ячейки A14 и O14 поставим значение 0.

В ячейку P2 впишем формулу:  $=МАКС(O2 : O13)$ .

## 31 Задача 23

Исполнитель преобразует число на экране. У исполнителя есть две команды, которые обозначены латинскими буквами:

А. Вычти 3

В. Найди целую часть от деления на 3

Программа для исполнителя – это последовательность команд.

Сколько существует программ, для которых при исходном числе 81 результатом является число 3, и при этом траектория вычислений содержит число 27?

**Решение**

---

```
def f(a, b):  
    if a < b:  
        return 0  
    if a == b:  
        return 1  
    return f(a - 3, b) + f(int(a / 3), b)  
print(f(81, 27) * f(27, 3))
```

---

**Ответ: 10**

## 32 Задача 24.1

В файле 24.txt дана строка, содержащая заглавные буквы латинского алфавита и цифры. Определите максимальную длину последовательности, которая начинается с буквы D и содержит 50 цифр, причем в последовательности нет букв D, за исключением начальной.

### Решение регулярками

---

```
from re import findall
f = open('24.txt')
s = f.readline()
reg = 'D(?:[A-CE-Z]*[0-9]){50}[A-CE-Z]*'
print(len(max(findall(reg, s), key=len)))
```

---

### Решение проходом по строке

---

```
f = open('24.txt')
s = f.readline()
ans = 0
for i in range(len(s)):
    if s[i] == 'D': # начинаем подстроку с D
        cnt_digits = 0
        for j in range(i + 1, len(s)):
            if s[j] in '0123456789': # встретили цифру
                cnt_digits += 1
            if cnt_digits == 50:
                # если цифр 50, берём всё до текущего символа (включительно)
                ans = max(ans, j - i + 1)
            elif cnt_digits > 50:
                # если более 50 цифр, берём всё до текущего символа (невкл)
                ans = max(ans, j - i)
                break
            if s[j] == "D": break
print(ans)
```

---

## 33 Задача 24.2

В файле 24.txt дана строка, содержащая заглавные буквы латинского алфавита и цифры. Определите максимальную длину последовательности, которая начинается с нечетной цифры и содержит 30 букв F, причем в последовательности нет нечетных цифр, за исключением начальной.

### Решение регулярками

---

```
from re import *
f = open('24.txt')
s = f.readline()
reg = '[13579](?:[02468A-EG-Z]*F){30}[02468A-EG-Z]*'
res = findall(reg, s)
print(max(res, key=len))
```

---

### Решение проходом по строке

---

```
f = open('24.txt')
s = f.readline()
ans = 0
for i in range(len(s)):
    if s[i] in '13579': # начинаем подстроку с нечётной цифры
        cnt_f = 0
        for j in range(i + 1, len(s)):
            if s[j] == 'F': # встретили F
                cnt_f += 1
            if cnt_f > 30:
                # если более 30 F, берём всё до текущего символа (невкл)
                ans = max(ans, j - i)
                break
            if s[j] in '13579':
                break
print(ans)
```

---

## 34 Задача 25

Определите первые 5 чисел, больших 2.900.000, которые являются произведением двух различных простых чисел, каждое из которых содержит в себе ровно один 0. В ответе в первом столбце таблицы запишите найденные числа в порядке возрастания, а во втором соответствующие им их максимальные множители.

### Решение

---

```
def prime(x):
    for i in range(2, int(x**0.5) + 1):
        if x % i == 0:
            return False
    return x > 1
k = 0
for x in range(2_900_000, 5_000_000):
    d = set()
    for i in range(2, int(x**0.5) + 1):
        if x % i == 0:
            d.add(i)
            d.add(x // i)
    b = list(d)
    if len(d) == 2 and str(b[0]).count('0') == 1 and str(b[1]).count('0') == 1
       and b[0] * b[1] == x and prime(b[0]) and prime(b[1]):
        print(x, max(b))
        k += 1
    if k == 5:
        break
```

---

### Ответ:

2900021 27103  
2900449 27107  
2900519 4091  
2900663 27109  
2901809 5701

## 35 Задача 26.1

В магазине для упаковки подарков есть  $N$  кубических коробок красного цвета и  $M$  кубических коробок зеленого цвета ( $N > M$ ). Самой интересной считается упаковка подарка по принципу матрешки – подарок упаковывается в одну из коробок, та в свою очередь в другую коробку и т. д., при этом цвет коробок чередуется. Одну коробку можно поместить в другую, если длина её стороны хотя бы на 5 единиц меньше длины стороны другой коробки. Определите наибольшее количество коробок, которое можно использовать для упаковки одного подарка, и максимально возможную длину стороны самой маленькой коробки, где будет находиться подарок. Размер подарка позволяет поместить его в самую маленькую коробку.

**Входные данные:** В первой строке входного файла находятся число  $N$  — количество коробок красного цвета в магазине (натуральное число, не превышающее 10 000) и через пробел число  $M$  — количество коробок зеленого цвета в магазине (натуральное число, не превышающее 10 000). В следующих  $N$  строках находятся значения длин сторон коробок красного цвета (все числа натуральные, не превышающие 10 000) и через знак табуляции значения длин сторон коробок зеленого цвета (все числа натуральные, не превышающие 10 000), каждая пара таких значений — в отдельной строке; в последних  $N - M$  строках второе число опускается, и числа, соответствующие длинам сторон коробок красного цвета, идут каждое в отдельной строке.

Запишите в ответе два целых числа: сначала наибольшее количество коробок, которое можно использовать для упаковки одного подарка, затем максимально возможную длину стороны самой маленькой коробки в таком наборе.

Типовой пример организации данных во входном файле

5 4

39 55

40 42

44 44

40 55

50

Пример входного файла приведён для случая пяти коробок красного цвета и четырёх коробок синего цвета, когда минимальная допустимая разница между длинами

сторон коробок, подходящих для упаковки «матрёшкой», составляет 3 единицы. При таких исходных данных ответом будет являться 4 40.

### Решение

#### Таблицами:

Добавляем обозначение цвета, пусть 0 – красный, 1 – синий

	A	B	C	D
1	5		4	
2	39	0	55	1
3	40	0	42	1
4	44	0	44	1
5	40	0	55	1
6	50	0		
7				

Переносим одно под другое, сортируем по убыванию столбец с матрёшками.

	A	B
1	55	1
2	55	1
3	50	0
4	44	0
5	44	1
6	42	1
7	40	0
8	40	0
9	39	0
10		
11		

В первую строку берем самую большую коробку с цветом.

Во вторую строку записываем формулу для размера:

=ЕСЛИ(C1-A2>=5; ЕСЛИ(ИЛИ(И(D1=1; B2=0); И(D1=0; B2=1))); A2; C1); C1)

Что происходит в формуле:

1) C1-A2>=5

Проверяем, достаточно ли велика разница между предыдущей коробкой (C1) и текущей (A2). Если разница >=5, то коробка может поместиться

2) Если разница достаточная, проверяем цвет:

ИЛИ(И(D1=1; B2=0); И(D1=0; B2=1))

И(D1=1; B2=0) - предыдущая синяя (1) И текущая красная (0)

И(D1=0; B2=1) - предыдущая красная (0) И текущая синяя (1)

ИЛИ - любое из этих условий = цвета чередуются

3) Результат:

Если разница >=5 И цвета чередуются -> берем новую коробку A2

Если разница >=5 НО цвета НЕ чередуются -> оставляем старую C1

Если разница  $<5$  -> оставляем старую  $C1$

В столбец  $D$  записываем формулу для цвета:

$$=ЕСЛИ(C1=C2; D1; B2)$$

$C1=C2$  - проверяем, изменился ли размер коробки

Если НЕ изменился ( $C1=C2$ ) -> цвет остается тот же  $D1$

Если изменился (взяли новую коробку) -> цвет новый  $B2$

	A	B	C	D	E	F	G	H	I
1	55	1	55	1					
2	55	1	C1; C1	1					
3	50	0	50	0					
4	44	0	50	0					
5	44	1	44	1					
6	42	1	44	1					
7	40	0	44	1					
8	40	0	44	1					
9	39	0	39	0					
10									
11									

Чтобы посчитать кол-во коробок, посмотрим: если размер коробки не меняется – значит, и кол-во коробок не увеличивается. Динамически посчитаем кол-во коробок:

$$=ЕСЛИ(C2=C1; E1; E1+1)$$

	A	B	C	D	E	F
1	55	1	55	1	1	
2	55	1	55	1	1	
3	50	0	50	0	2	
4	44	0	50	0	2	
5	44	1	44	1	3	
6	42	1	44	1	3	
7	40	0	44	1	3	
8	40	0	44	1	3	
9	39	0	39	0	4	
10						

Тогда кол-во будет равно максимальному, последнему значению, для подсчета наименьшей нужно посмотреть на последнюю взятую возможную коробку: на примере на коробки со значениями 4 в столбце  $E$ : если бы их было несколько – взяли бы среди них ту, что имеет наибольшее значение.

Порогой:

---

```
f = open("26.txt")
first_line = f.readline().strip()
n, m = map(int, first_line.split())

red_boxes = []
green_boxes = []

for i in range(m):
    line = f.readline().strip()
    parts = line.split()
    red_boxes.append(int(parts[0]))
    green_boxes.append(int(parts[1]))
for i in range(n - m):
    line = f.readline()
    red_boxes.append(int(line))

all_boxes = []

# красные коробки (цвет = 0)
for size in red_boxes:
    all_boxes.append((size, 0))

# зеленые коробки (цвет = 1)
for size in green_boxes:
    all_boxes.append((size, 1))

# сортируем по убыванию размера
all_boxes.sort(reverse=True)

# находим максимальную матришку
max_boxes = 0
max_min_size = 0
```

```

# пробуем начать с каждой коробки
for start in range(len(all_boxes)):
    # начинаем новую последовательность
    matryoshka = [all_boxes[start]]
    current_size = all_boxes[start][0]
    current_color = all_boxes[start][1]

    # ищем подходящие коробки для продолжения
    for i in range(start + 1, len(all_boxes)):
        next_size, next_color = all_boxes[i]

        # проверяем условия:
        # 1. разница в размерах >= 5
        # 2. цвета чередуются
        if current_size - next_size >= 5 and next_color != current_color:
            matryoshka.append(all_boxes[i])
            current_size = next_size
            current_color = next_color

    # обновляем максимум
    if len(matryoshka) > max_boxes:
        max_boxes = len(matryoshka)
        max_min_size = matryoshka[-1][0]
    elif len(matryoshka) == max_boxes and matryoshka[-1][0] > max_min_size:
        max_min_size = matryoshka[-1][0]

print(max_boxes, max_min_size)

```

---

## 36 Задача 26.2

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 представлены в файле следующим образом. Первая строка входного файла содержит натуральное число  $N$  ( $1 \leq N \leq 1000$ ) – 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 раннее время начала последнего мероприятия (в минутах от начала суток).

Пример входного файла:

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. Последнее мероприятие в конференц-зала начнётся самое раннее на 131-й минуте, если состоятся мероприятия по заявкам 2, 4, 5. Ответ: 3 131.

## Решение

### Решение при помощи электронных таблиц:

Сначала переносим данные в Excel, удалим первую строчку, а потом при помощи настраиваемой сортировки отсортируем в первую очередь по столбцу В, а потом по столбцу А - всё по возрастанию. Нужно сортировать по времени окончания мероприятия, так как чем раньше одно кончится, тем раньше сможет начаться следующее.

В ячейку С1 запишем значение ячейки В1, в ячейку С2 запишем следующую формулу и растянем её до конца таблицы:

$$=ЕСЛИ(А2>=С1;В2;С1)$$

Таким образом у нас в столбце С будет храниться время окончания последнего (на данный момент) мероприятия. Каждое новое мероприятие мы проверяем - может ли оно начаться после предыдущего последнего, и если да - сохраняем уже новое время окончания, иначе дублируем старое.

В столбце D мы посчитаем количество прошедших мероприятий. В ячейку D1 запишем 1, а в ячейку D2 запишем следующую формулу и растянем её до конца таблицы:

$$=ЕСЛИ(С2<>С1;D1+1;D1)$$

Если сменилось время окончания - увеличиваем счётчик, иначе дублируем предыдущий.

Максимальное число в столбце D и будет количеством мероприятий.

Минимальное допустимое время начала последнего мероприятия найдём при помощи следующей формулы, в качестве диапазона выбрав отрезок столбца А напротив максимального числа в столбце С:

$$=МИНЕСЛИ(А903:А966;А903:А966;">="&С902)$$

*Не обязательно, что число именно такое, приведено в качестве примера.*

## Решение при помощи программы:

---

```
f = open('3_26_conf.txt')
n = int(f.readline())

confs = [list(map(int, i.split())) for i in f]

# Сортируем по времени окончания, так как
# чем раньше одно кончится, тем раньше сможет начаться следующее
confs.sort(key=lambda x: (x[1], x[0]))

# Время окончания предпоследнего мероприятия
prev_end = 0
# Время окончания последнего мероприятия
last_end = confs[0][1]
# Все возможные времена начала последнего мероприятия
last_starts = [confs[0][0]]
cnt = 1

for start, end in confs:
    # Может начаться новое - начинаем, не может - добавляем в список возможных
    if start >= last_end:
        cnt += 1
        # Смещаем последнее в предпоследнее и обновляем последнее
        prev_end = last_end
        last_end = end
        # Обнуляем список с началами
        last_starts = [start]
    else:
        last_starts.append(start)

# вывод количества и минимального времени начала рейса,
# который может быть последним
print(cnt, min([i for i in last_starts if i >= prev_end]))
```

---

## 37 Задача 26.3

На производстве штучных изделий  $N$  деталей должны быть отшлифованы и окрашены. Для каждой детали известно время ее шлифовки и время окрашивания. Детали пронумерованы начиная с нуля. Параллельная обработка деталей не предусмотрена. На ленте транспортера имеется  $N$  мест для каждой из  $N$  деталей. На ленте транспортера детали располагают по следующему алгоритму:

- все  $2N$  чисел, обозначающих время окрашивания и шлифовки для  $N$  деталей, упорядочивают по возрастанию;
- если максимальное число в этом упорядоченном списке – это время окрашивания конкретной детали, то деталь размещают на ленте транспортера на первое свободное место от ее начала;
- если максимальное число – это время шлифовки, то деталь размещают на первое свободное место от конца ленты транспортера;
- если число обозначает время окрашивания или шлифовки уже рассмотренной детали, то его не принимают во внимание.

Этот алгоритм применяется последовательно для размещения всех  $N$  деталей.

Определите сколько деталей будет окрашено и какой номер будет иметь последняя отшлифованная деталь.

Входные данные представлены в файле следующим образом. Первая строка входного файла содержит натуральное число  $N$  ( $1 \leq N \leq 10000$ ) – количество деталей. Следующие  $N$  строк содержат пары чисел, обозначающих соответственно время шлифовки и время окрашивания конкретной детали (все числа натуральные, различные).

Запишите в ответе два натуральных числа через пробел: сначала количество деталей, которые будут окрашены, затем номер последней отшлифованной детали.

## Решение

---

```
f = open('26_12.txt')
n = int(f.readline())
a = [list(map(int,i.split())) for i in f]
t = [] #список для хранения времени, типа операции и
# порядкового номера каждой детали
for i in range(n):
    if a[i][0] > a[i][1]:
        t.append([a[i][0], 1, i]) #шлифовка
    else:
        t.append([a[i][1], 2, i]) #окрашивание
t.sort()
line_start = [] #списки для хранения деталей, которые размещаются в начале
line_end = [] #списки для хранения деталей, которые размещаются в конце
for i in t: #заполнение списков в зависимости от типа операции
    if i[1] == 2:
        line_start.append(i)
    else:
        line_end.append(i)
print(len(line_start))
print(line_end[-1][2])
```

---

## 38 Задача 26.4

На вокзале работает автоматическая камера хранения, в которой есть  $K$  ячеек для временного хранения багажа. Все ячейки пронумерованы, начиная с единицы. Стоимость размещения багажа в любой ячейке составляет  $m$  рублей (фиксированная плата, не зависящая от времени хранения). Известно время, в которое каждый пассажир хочет сдать багаж, и время, когда он планирует его забрать. При поступлении заявки багаж размещается в свободной ячейке с наименьшим номером. Для размещения или извлечения багажа требуется 1 минута. Со следующей минуты ячейка становится доступной для использования другим пассажиром. Если пассажир прибыл, но свободных ячеек нет, он уходит без оплаты.

Определите общую выручку камеры хранения за 24 часа и номер ячейки, в которую будет помещён последний багаж. Если для размещения багажа доступно несколько ячеек, выберите свободную ячейку с наименьшим номером.

### Входные данные

В первой строке входного файла находятся два числа:  $K$  — количество ячеек в камере хранения (натуральное число, не превышающее 1000) и  $m$  — стоимость размещения багажа в рублях (натуральное число, не превышающее 10000). Во второй строке находится число  $N$  — количество пассажиров, желающих воспользоваться камерой хранения ( $N \leq 1000$ ). В следующих  $N$  строках указаны два значения: минута размещения багажа и минута, до которой планируется хранение багажа в ячейке. Отсчёт времени ведётся с начала суток (все числа неотрицательные, не превышающие 1440). Каждая строка соответствует отдельному пассажиру.

### Выходные данные

Запишите в ответе два целых числа: сначала общую выручку камеры хранения за 24 часа, затем наименьший номер ячейки, в которую будет помещён последний багаж.

### Типовой пример организации данных во входном файле

```
2 50
5
30 60
40 1110
59 60
```

61 120  
1230 1440

При таких исходных данных первый, второй, четвёртый и пятый пассажиры смогут воспользоваться ячейками. Общая выручка составит  $4 \times 50 = 200$  рублей. Последний пассажир оставит свой багаж в первой ячейке (так как первая и вторая ячейки будут свободны). Ответ: 200 1.

### Решение

---

```
f = open("26.txt")
k, m = map(int, f.readline().split())
n = int(f.readline())

a = [list(map(int, i.split())) for i in f]
a.sort()

cells = [0] * k # 0 означает, что ячейка свободна

cnt = 0
last_cell = -1

for start_time, end_time in a:
    for i in range(k):
        if cells[i] < start_time: # ячейка свободна
            cells[i] = end_time
            cnt += 1
            last_cell = i + 1
            break

# выручка
revenue = cnt * m

print(revenue, last_cell)
```

---

Ответ: 7242 3

## 39 Задача 26.5

Логистическая компания использует контейнеры с максимальной вместимостью  $M$  кг. Для отправки  $N$  грузов контейнеры заполняют грузами в порядке убывания веса, пока общая масса грузов не превысит максимальную вместимость  $M$ . Затем процедуру повторяют для следующего контейнера, пока все грузы не будут размещены. Необходимо определить количество контейнеров для отправки всех грузов и общую загрузку предпоследнего контейнера.

### Входные данные

Входные данные представлены в файле следующим образом. В первой строке записаны два числа – количество грузов  $N$  и максимальная вместимость контейнера  $M$ . В каждой из следующих  $N$  строк записано по одному натуральному числу, обозначающему вес груза.

### Пример входного файла

```
6 100
30
10
40
50
10
20
```

В первый контейнер будут помещены грузы весом 50, 40 и 10 кг (общий вес 100 кг), во второй контейнер – грузы весом 30, 20 и 10 кг (общий вес 60 кг). Ответ: 2 100.

### Решение

```
with open("26.txt") as file:
    n, m = map(int, file.readline().split())
    weights = sorted([int(line) for line in file], reverse=True)

    container_count = 0
    res = []
    while weights:
```

```
current_weight = 0
current_ans = []
for weight in weights[:]:
    if current_weight + weight <= m:
        current_weight += weight
        weights.remove(weight)
        current_ans.append(weight)
container_count += 1
res.append(current_ans)

print(container_count, sum(res[-2]))
```

---

**Ответ: 86 149**

## 40 Задача 27.1

Учёный решил провести кластеризацию некоторого множества звёзд по их расположению на карте звёздного неба. Кластер звёзд – это набор звёзд (точек) на графике, лежащий внутри круга радиусом  $R$ . Каждая звезда обязательно принадлежит только одному из кластеров. Истинный центр кластера, или центроид, – это одна из звёзд на графике, сумма расстояний от которой до всех остальных звёзд кластера минимальна. Под расстоянием понимается расстояние Евклида между двумя точками  $A(x_1, y_1)$  и  $B(x_2, y_2)$  на плоскости, которое вычисляется по формуле:

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

**Аномалиями** назовём точки, находящиеся на расстоянии более одной условной единицы от точек кластеров. При расчётах аномалии учитывать не нужно.

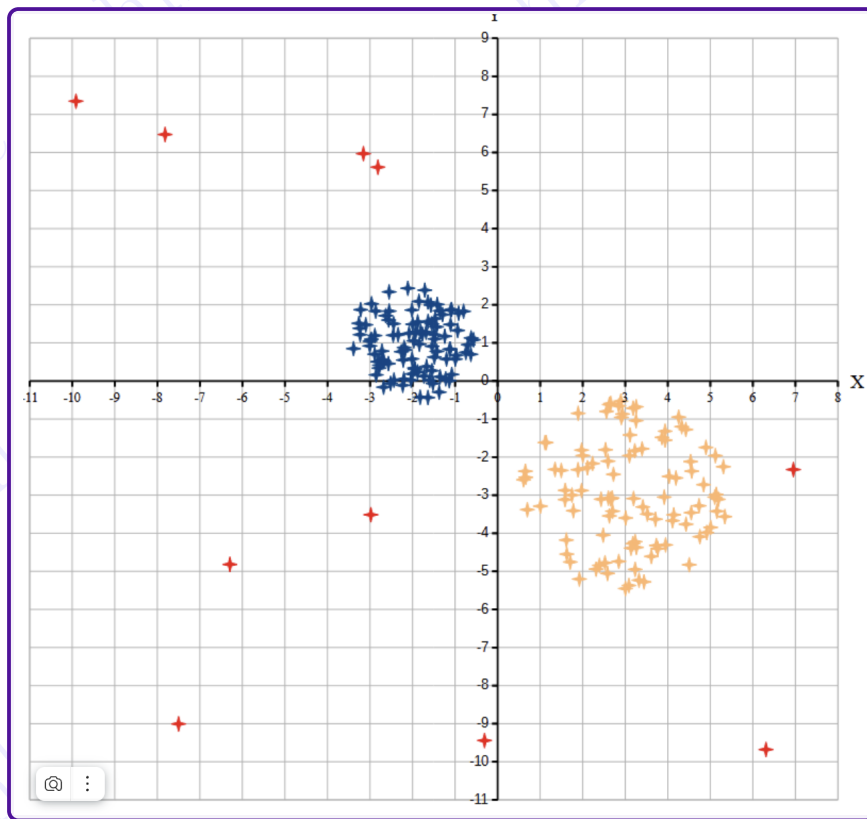
В файле А хранятся данные о звёздах двух кластеров, где  $R = 2$  для каждого кластера. В каждой строке записана информация о расположении на карте одной звезды: сначала координата  $x$ , затем координата  $y$ . Значения даны в условных единицах, которые представлены вещественными числами. Известно, что количество звёзд не превышает 1000.

В файле Б хранятся данные о звёздах трех кластеров, где  $R = 6$  для каждого кластера. Известно, что количество звёзд не превышает 10 000. Структура хранения информации о звездах в файле Б аналогична файлу А.

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 через пробел: сначала целую часть произведения  $|P_x| \cdot 100$  для файла А, затем  $|P_y| \cdot 100$  для файла А, далее целую часть произведения  $|P_x| \cdot 100$  для файла Б и  $|P_y| \cdot 100$  для файла Б. Возможные данные одного из файлов иллюстрированы графиком.

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



## Решение

### Файл А

---

```
f = open('1A.txt')
n = f.readline()
a = [[] for i in range(2)]
for i in range(811):
    star = list(map(float, f.readline().replace(',', '.').split()))
    if (star[1] > 0 and star[1] < 40) and (star[0] > -20 and star[0] < 20):
        a[0].append(star)
    elif star[1] < -60 and star[0] > 0:
        a[1].append(star)

sum_x = sum_y = tx = ty = 0
for i in a:
    mn = 100000050000
    for j in i:
        x1, y1 = j
        sm = 0
        for k in i:
            x2, y2 = k
            sm += ((x2-x1)**2 + (y2-y1)**2)**0.5
        if sm < mn:
            mn = sm
            tx, ty = x1, y1
    sum_x += tx
    sum_y += ty
print(int(abs(sum_x / 2) * 100))
print(int(abs(sum_y / 2) * 100))
```

---

## Файл Б

---

```
f = open('1B.txt')
n = f.readline()
a = [[] for i in range(3)]
for i in range(11003):
    star = list(map(float, f.readline().replace(',', ' ').split()))
    if star[1] < -60 and star[0] > -20:
        a[0].append(star)
    elif (star[1] < 0 and star[1] > -40) and star[0] > -40:
        a[1].append(star)
    elif star[1] < -40 and star[0] < -60:
        a[2].append(star)

sum_x = sum_y = tx = ty = 0
for i in a:
    mn = 100000050000
    for j in i:
        x1, y1 = j
        sm = 0
        for k in i:
            x2, y2 = k
            sm += ((x2 - x1) ** 2 + (y2 - y1) ** 2) ** 0.5
        if sm < mn:
            mn = sm
            tx, ty = x1, y1
    sum_x += tx
    sum_y += ty
print(int(abs(sum_x / 3) * 100))
print(int(abs(sum_y / 3) * 100))
```

---

## 41 Задача 27.2

Учёный решил провести кластеризацию некоторого множества звёзд по их расположению на карте звёздного неба. Кластер звёзд – это набор звёзд (точек) на графике, лежащий внутри круга радиусом  $R$ . Каждая звезда обязательно принадлежит только одному из кластеров. Истинный центр кластера, или центроид, – это одна из звёзд на графике, сумма расстояний от которой до всех остальных звёзд кластера минимальна. Под расстоянием понимается расстояние Евклида между двумя точками  $A(x_1, y_1)$  и  $B(x_2, y_2)$  на плоскости, которое вычисляется по формуле:

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

**Аномалиями** назовём точки, находящиеся на расстоянии более одной условной единицы от точек кластеров. При расчётах аномалии учитывать не нужно.

В файле А хранятся данные о звёздах трёх кластеров, где  $R = 2$  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, которые представлены вещественными числами. Известно, что количество звёзд не превышает 1000.

В файле Б хранятся данные о звёздах четырёх кластеров, где  $R = 6$  для каждого кластера. Известно, что количество звёзд не превышает 10 000. Структура хранения информации о звездах в файле Б аналогична файлу А.

Для файла А определите наибольшую абсциссу  $M_x$  и наибольшую ординату  $M_y$  среди координат центроидов кластеров. Для файла Б определите разность между абсциссами  $D_x$  и ординатами  $D_y$  центроидов кластера с наибольшим количеством звезд и кластера с наименьшим количеством звезд.

В ответе запишите четыре числа через пробел: сначала целую часть произведения  $|M_x| \cdot 100$  для файла А, затем  $|M_y| \cdot 100$  для файла А, далее целую часть произведения  $|D_x| \cdot 10$  для файла Б и  $|D_y| \cdot 10$  для файла Б.

## Решение

### Файл А

---

```
f = open('2A.txt')
n = f.readline()
a = [[] for i in range(3)]
for i in range(15032):
    star = list(map(float, f.readline().replace(',','').split()))
    if -40 < star[0] < 0 and 50 < star[1] < 100:
        a[0].append(star)
    elif 125 < star[0] < 165 and -50 < star[1] < 0:
        a[1].append(star)
    elif 124 < star[0] < 160 and -125 < star[1] < -75:
        a[2].append(star)

m_x = m_y = tx = ty = 0
for i in a:
    mn = 100000050000
    for j in i:
        x1, y1 = j
        sm = 0
        for k in i:
            x2, y2 = k
            sm += ((x2-x1)**2 + (y2-y1)**2)**0.5
        if sm < mn:
            mn = sm
            tx, ty = x1, y1
m_x = max(m_x, tx)
m_y = max(m_y, ty)
print(int(abs(m_x) * 100))
print(int(abs(m_y) * 100))
```

---

## Файл Б

---

```
from math import *
def centroid(cl):
    tx, ty = 0
    mn = 100000050000
    for j in cl:
        sm = 0
        for k in i:
            sm += dist(j,k)
        if sm < mn:
            mn = sm
            tx = j[0], ty = j[1]
    return [tx,ty]
f = open('2B.txt')
n = f.readline()
a = [[] for i in range(4)]
for i in range(2145):
    star = list(map(float, f.readline().replace(',','').split()))
    if -180 < star[0] < -140 and -150 < star[1] < -110:
        a[0].append(star)
    elif -180 < star[0] < -140 and -200 < star[1] < -150:
        a[1].append(star)
    elif 60 < star[0] < 90 and -55 < star[1] < -20:
        a[2].append(star)
    elif -10 < star[0] < 25 and -180 < star[1] < -140:
        a[3].append(star)
mn_cl = min(a, key = len)
mx_cl = max(a, key = len)
c_mn_cl = centroid(mn_cl)
c_mx_cl = centroid(mx_cl)
print(int(abs(c_mx_cl[0]-c_mn_cl[0]) * 10))
print(int(abs(c_mx_cl[1]-c_mn_cl[1]) * 10))
```

---