

## Информатика. Дальний восток. 10.06. 6:45.

### Содержание

1	Задача 2	3
2	Задача 5	3
3	Задача 6	4
4	Задача 7	7
5	Задача 8	7
6	Задача 9	8
7	Задача 11	8
8	Задача 11	8
9	Задача 10	9
10	Задача 12	9
11	Задача 13	10
12	Задача 14	11
13	Задача 15	12
14	Задача 17	13
15	Задача 22	13
16	Задача 24	14

17	Задача 19	15
18	Задача 20	15
19	Задача 21	15
20	Задача 25	16
21	Задача 26	17
22	Задача 27	20

## 1 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  $w, x, z$ .

				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)
```

---

Ответ:

## 2 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. Например, для исходного числа  $12 = 110_3$ , результатом является число  $11010_3 = 111$ , а для исходного числа  $4 = 11_3$ , остаток от деления на 3 равен 1,  $1 - 5 = 5$ , в троичной записи  $5 = 123_3$ , результат:  $11123_3 = 41$ . 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
```

---

## 3    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  $p$  (где  $p$  - целое число), вызывающая передвижение Черепахи на  $p$  единиц в том направлении, куда указывает её голова; Назад  $p$  (где  $p$  - 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] 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(12*k)
    right(90)
    forward(18*k)
    right(90)
pu()
forward(7*k)
right(90)
forward(7*k)
left(90)
pd()
for j in range(2):
    forward (13*k)
    right(90)
    forward(14*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: 385

## 4    Задача 7

Фотографии размером  $1920 \times 1080$ . В палитре фотографии содержится  $2^{22}$  цветов. Для экономии трафика фотографию сжали до  $1280 \times 1024$  пикселей. Глубина кодирования цвета 21 бит. Найдите, сколько Кбайт было сэкономлено при передаче 120 фотографий.

### Решение

Объем  $I_1$  фотографий до сжатия:  $1920 * 1080 * 21 * 120$  бит.

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

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

**Ответ: 265050**

## 5    Задача 8

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**

## **6    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.

## **7    Задача 11**

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**

Для 1 серийного номера нужно:  $\frac{14 \cdot 1024 \cdot 1024 \cdot 1024}{2984523} = 5036,78$  байт.

Для хранения одного символа нужно:  $\frac{5036,78 \cdot 8}{3410} = 11,8$  бита  $\sim 12$ .

Следовательно, минимальная мощность алфавита – 2049 символов.

**Ответ: 2049**

## **8    Задача 11**

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 запи-



шите только целое число.

### Решение

$$\frac{248 \cdot i}{8} * 75600 \geq 16 * 2^{20}$$

$$\frac{248 \cdot i}{8} \geq 221, \dots$$

Решая неравенство, получаем, что мощность алфавита:  $2^7 + 1 = 129$

**Ответ: 129**

## 9    Задача 10

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

## 10    Задача 12

Исполнитель Редактор получает на вход строку цифр и преобразовывает её. Редактор может выполнять две команды, в обеих командах  $v$  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**

## 11 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**

**12    Задача 14**

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 +
    a2 = 5 * 27 ** 5 + 6 * 27 ** 5 + 5 * 27 ** 4 + x * 27 ** 3 + 2 * 27 ** 2 +
    a = a1 + a2
    if a % 26 == 0:
        print(a//26)
        break
```

---

Ответ: 825862605

## 13    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(1, 1000):
        for y in range(1, 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)
        break
```

---

Ответ: 116

## 14 Задача 17

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))
```

---

## 15 Задача 22

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.

## 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.

## 16 Задача 24

Необходимо найти подстроку максимальной длины, которая начинается с символа D и содержит его ровно один раз, и содержит 50 цифр.

## Решение

---

```
f = open('...')
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 более 50 цифр, берём всё до текущего символа
                ans = max(ans, j - i)
                break
            if s[j] == "D":
                break
print(ans)
```

---



## 17 Задача 19

Два игрока, Петя и Ваня, играют в следующую игру. Перед игроками лежит куча камней. Игроки ходят по очереди, первый ход делает Петя. За один ход игрок может 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.

## 18 Задача 20

Для игры, описанной в предыдущем задании, найдите два таких значения  $S$ , при котором у Пети есть выигрышная стратегия, причём Петя не может выиграть за один ход и Петя может выиграть своим вторым ходом независимо от того, как будет ходить Ваня.

В ответе запишите числа в порядке возрастания без пробелов и знаков препинания.

## 19 Задача 21

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.

## 20    Задача 25

Определите первые 5 чисел, больших 2.900.000, которые являются произведением двух простых чисел и содержат 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 simple(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 simple(
        b[0]) and simple(b[1]):
        print(x, max(b))
        k += 1
    if k == 5:
        break
```

---

### Ответ:

2900021 27103



2900449 27107

2900519 4091

2900663 27109

2901809 5701

## 21    Задача 26

В магазине для упаковки подарков есть  $N$  кубических коробок красного цвета и  $M$  кубических коробок синего 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 синего 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 синего 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

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 5 единицы.

### Решение

---

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

red_boxes = []
green_boxes = []

min_n_m = min(n, m)
for i in range(min_n_m):
    line = f.readline().strip()
    parts = line.split()
    red_boxes.append(int(parts[0]))
    if len(parts) > 1 and i < m:
        green_boxes.append(int(parts[1]))
if n > m:
    for i in range(n - m):
        line = f.readline()
        red_boxes.append(int(line))
if m > n:
    for i in range(m - n):
        line = f.readline()
        green_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 матрёшку
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 >= 3 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)

```

---

## 22    Задача 27

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.

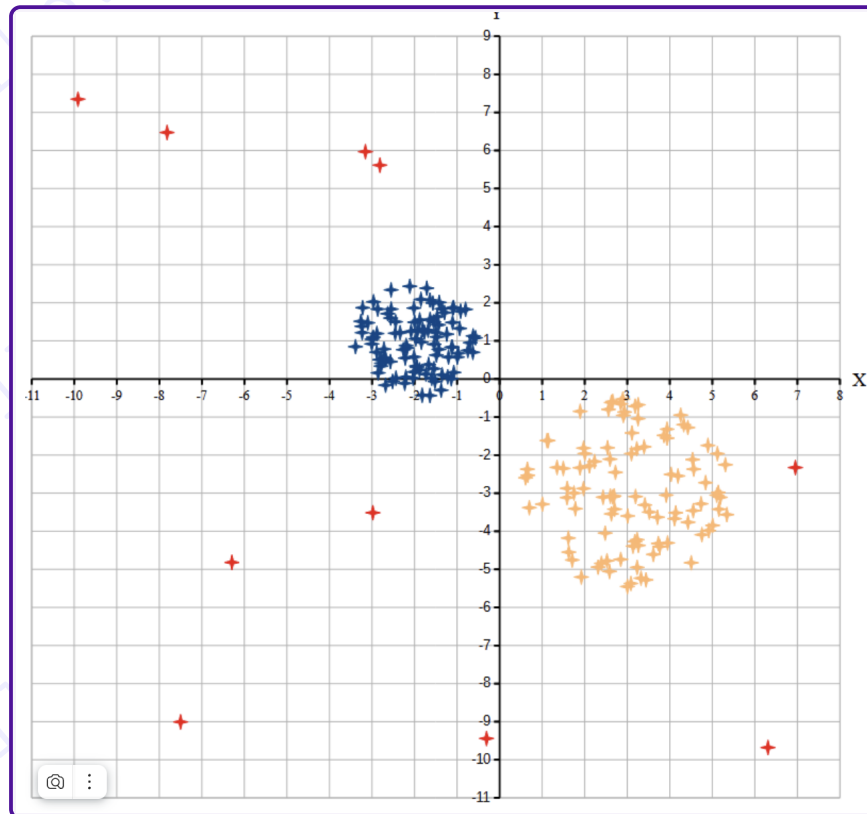
В файле А хранятся данные о звёздах двух кластеров, где  $R = 2$  для каждого кластера. В каждой строке записана информация о расположении на карте одной звезды: сначала координата  $x$ , затем координата  $y$ . Значения даны в условных единицах, которые представлены вещественными числами. Известно, 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('3A.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

---

```

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))
```

---

**Ответ:**