Информатика. Задания на кластеризацию.

1 Задание 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 прямоугольника высотой H и шириной W. 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:

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

Входные данные

В файле A хранятся данные о звёздах **двух** кластеров, где $H = 3$, $W = 3$ для каждого кластера. В каждой строке записана информация о расположении на карте одной звезды: сначала координата x, затем координата y. Значения даны в условных единицах. Известно, что количество звёзд не превышает 1000.

В файле $Б$ хранятся данные о звёздах **трёх** кластеров, где $H = 3$, $W = 3$ для каждого кластера. Известно, что количество звёзд не превышает 10000. Структура хранения информации о звездах в файле $Б$ аналогична файлу A. Для каждого файла определите координаты центра каждого кластера, затем вычислите два числа: Px – среднее арифметическое абсцисс центров кластеров, и Py – среднее арифметическое ординат центров кластеров.

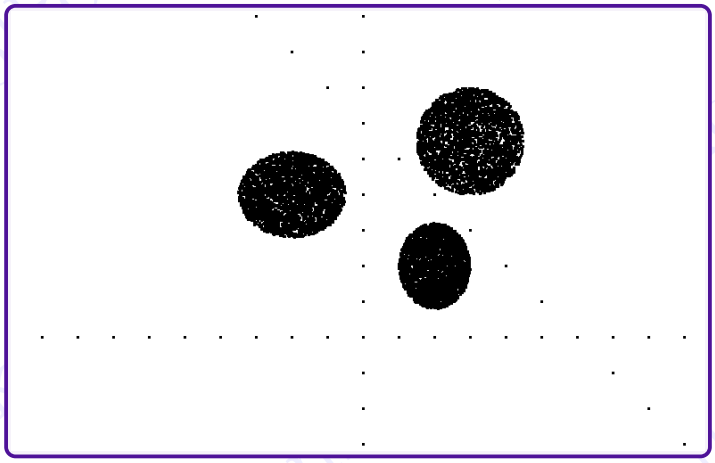
Выходные данные

В ответе запишите четыре числа: в первой строке сначала целую часть произведения $Px \times 10000$, затем целую часть произведения $Py \times 10000$ для файла A, во второй строке – аналогичные данные для файла $Б$. Возможные данные одного из файлов иллюстрированы графиком.

Решение:

Изобразим кластеры с помощью модуля turtle в python:

```
from turtle import *
f = open('27-1b.txt')
s = f.readline()
print (s)
tracer (0) # Отключение анимации
pu () # Поднятие хвоста
m = 40 # Масштаб
# Рисуем оси
for i in range(-50, 50):
    goto(i*m, 0)
    dot(4)
    goto(0, i*m)
    dot(4)
    goto(i*m, (-i+6)*m) # Подбираем значение прямой, разделяющей кластеры
    dot(4)
# Рисуем кластеры
for i in f:
    x, y = map(float, i.replace(',', '.').split())
    goto(x*m, y*m)
    dot(4)
done()
```



```
f = open('27-1b.txt')
s = f.readline()
c = [[], [], []] # Список кластеров
for i in f:
    x, y = map(float, i.replace(',', '.').split())
    if x < 0: # Если 'x' отрицательный
        c[0].append([x, y]) # Записываем координаты звезды в левый кластер
    elif y < -x+6: # Если 'y' лежит ниже прямой '-x+6'
        c[1].append([x, y]) # Записываем координаты звезды в нижний кластер
    else:
        c[2].append([x, y]) # Записываем координаты в правый верхний кластер

# Поиск центроидов
srx = sry = 0 # Искомые средн. арифметические
mx = my = 0
for k in range(3): # Для каждого кластера
    ms = 100000050000000000 # Минимальное расстояние
    for j in range(len(c[0])): # Для каждой звезды кластера
        x1, y1 = c[k][j] # Координаты звезды, от которой ищем расстояние
        s = 0 # Длина расстояния
        for i in c[k]:
            x2, y2 = i # Координаты звезды, до которой ищем расстояние
            s += ((x2-x1)**2 + (y2-y1)**2)**0.5
        if s < ms: # Если расстояние меньше минимума
            ms = s # Обновляем ms
            mx, my = x1, y1 # Запоминаем координаты предполагаемого центроида
    srx += mx # Добавляем координату 'x' найденного центроида
    sry += my # Добавляем координату 'y' найденного центроида
print(int(srx/3*10000), int(sry/3*10000))
```

Ответ: 9864 38539