

Strim_9_progr

Ссылка на видео-разбор с таймингом: https://vk.com/video-205546952_456241211?t=4m42s

Задача № 1 (4984)

В файле электронной таблицы [9-154.xls](#) в каждой строке содержатся пять натуральных чисел. Сколько среди них строк, в которых квадрат произведения максимального и минимального чисел в строке больше утроенного произведения трех оставшихся?

Решение

Как считать числа из файла в программу.

Ссылка на видео-разбор с таймингом: https://vk.com/video-205546952_456241211?t=2m50s

Сначала нам нужно подготовить данные так, чтобы наша программа могла их понять. Дело в том, что программа сама по себе не умеет работать с электронными таблицами, такими как Excel. Поэтому мы должны превратить эти сложные файлы в простые текстовые документы, где каждая строка — это отдельная запись. Такой формат проще читать и обрабатывать.

После этого мы можем спокойно открыть этот текстовый файл и считать оттуда нужные данные.

Программа будет последовательно просматривать каждую строку и выполнять необходимые вычисления. Таким образом, мы сможем получить нужную информацию и использовать её дальше.

Будем работать с файлом под названием «9_154», файл можно скачать по ссылке в описании.

Открываем этот файл и выделяем все числа в столбцах от А до Е. Ничего менять не надо, просто выделяем всё подряд. Теперь нажимаем Ctrl+C, чтобы скопировать эти данные.

Далее открываем программу Блокнот. Вставляем туда наши данные и сохраняем файл на рабочий стол с понятным именем, например, «9_154.txt». Можно сохранить его в папке с вашим проектом, если вы работаете в PyCharm.

Теперь переходим к работе с этим текстовым файлом в Python.

Обратите внимание, что при сохранении файла важно указать его полное название с расширением .txt, даже если система скрывает расширения файлов. Например, если вы сохраняете файл как «9-154», убедитесь, что добавили к нему расширение .txt.

Когда вы сохраняете свою программу, старайтесь сохранять её в той же папке, где находится текстовый файл. Это упростит доступ к данным и поможет избежать ошибок с путями к файлам.

Например, можно сохранить программу в той же директории, где хранится файл «9-154.txt».

Важно помнить, что оба файла должны находиться в одной папке. Это необязательно должен быть рабочий стол, главное — чтобы они были в одном месте.

Ссылка на видео-разбор с таймингом: https://vk.com/video-205546952_456241211?t=16m45s

Чтобы открыть файл и начать его читать, используем специальную функцию `open()`. Она позволяет открыть файл по указанному пути и начать с ним работать. Создадим переменную `f` для хранения нашего открытого файла:

```
f = open('9-154.txt')
```

Эта команда откроет файл «9-154.txt» в режиме чтения. Теперь мы готовы приступить к дальнейшим действиям с данными из этого файла.

Файл открывается построчно, то есть каждая строка текста считывается отдельно. Для этого используется цикл `for`, который проходит по всем строкам файла:

```
f = open('9-154.txt')
# Обработка каждой строки
for s in f:
```

Первым шагом мы разделим строку на отдельные элементы. Для этого воспользуемся методом `split()`, который разбивает строку на части по указанным разделителям. В нашем случае разделителем являются пробельные символы, такие как пробелы или табуляции.

Применив метод `s.split()` к строке, мы получим список из пяти элементов, соответствующих числу столбцов в исходной таблице. Однако стоит учесть, что каждый элемент списка пока остается строкой, хотя визуально они выглядят как числа. Это связано с тем, что метод `split()` возвращает результат в виде строковых объектов.

Этот цикл последовательно обрабатывает каждую строку файла. При запуске программы выводятся строки по порядку: первая, вторая, третья и так далее. Важно отметить, что каждая такая строка представляет собой единый текстовый фрагмент, а не отдельные числа.

Перед дальнейшей обработкой необходимо разделить эту строку на отдельные элементы и преобразовать их в числовой формат, поскольку в текущем виде строка не подходит для математических операций.

На втором этапе необходимо преобразовать полученные элементы списка из строкового типа в числовой. Для этого применяется функция `int()`, которая преобразует строковые представления чисел в целые значения.

Чтобы применить функцию `int()` ко всем элементам списка одновременно, удобно воспользоваться конструкцией, называемой "списковое включение". Создадим новый список `A`, который будет содержать результаты преобразования каждого элемента:

```
f = open('9-154.txt')
for s in f:
    a = [int(x) for x in s.split()]
```

В результате выполнения этой операции список `a` будет содержать пять целых чисел, полученных из исходной строки.

Для выполнения данного задания необходимо упорядочить числа в порядке возрастания. Это позволит легко проверить требуемое условие, так как минимальное значение окажется первым элементом списка, максимальное — последним, а промежуточные числа будут расположены между ними. Чтобы расположить числа в порядке возрастания, нужно воспользоваться методом `sorted()`. Этот метод принимает список и возвращает новый список, элементы которого упорядочены по возрастанию. Для сортировки наших чисел достаточно обернуть выражение в вызов метода `sorted()` следующим образом:

```
f = open('9-154.txt')
for s in f:
    #a = [int(x) for x in s.split()]
    a = sorted([int(x) for x in s.split()])
```

Так мы получим отсортированный список, где числа идут в порядке увеличения.

Теперь, когда у нас есть отсортированный список чисел, мы можем обратиться к каждому из них по отдельности и проверить выполнение условий задачи. Для начала найдем минимальное и максимальное числа. Минимальным числом будет первый элемент списка `a[0]`, а максимальным — последний `a[4]`.

Затем рассчитаем квадрат произведения этих двух чисел. Полученное значение должно быть больше утроенного произведения остальных трёх чисел.

$(a[0] * a[4])^2 > 3 * a[1] * a[2] * a[3]$:

Если условие выполняется, необходимо подсчитать количество строк, удовлетворяющих этому условию. Для этого сначала создается счётчик, который изначально равен нулю. Каждый раз, когда встречается подходящая строка, счётчик увеличивается на единицу.

После завершения цикла по всем строкам файла значение счётчика выводится на экран, и таким образом определяется общее количество подходящих строк.

```
f = open('9-154.txt')
k = 0
for s in f:
    a = sorted([int(x) for x in s.split()])
# Проверяя выполнение этого условия, мы решим задачу
    if (a[0]*a[4])**2 > 3*a[1]*a[2]*a[3]:
        k = k + 1
print(k)
```

Результат работы программы:

2511

Ответ: 2511

Задача №2 (5339)

В файле электронной таблицы [9-159.xls](#) в каждой строке содержатся пять натуральных чисел. Определите количество строк таблицы, в которых среднее значение всех пяти чисел меньше среднего значения между максимальным и минимальным из них.

Ссылка на видео-разбор с таймингом: https://vk.com/video-205546952_456241211?t=16m45s

Решение

Копируем содержимое прилагаемого файла, вставляем его в блокнот и сохраняем на рабочем столе или любой другой удобной директории.

Откроем в IDLE файл (9-159.txt).

Обнулим переменную `k`, которая будет использоваться для подсчета строк, соответствующих условиям.

Прочитаем каждую строку файла использованием цикла `for` и разобьём её на отдельные числа с помощью метода `split()` и сразу отсортируем список чисел с помощью встроенной функции `sorted()`.

Проверка условия.

Сравним два средних значения: среднее арифметическое пяти чисел и среднее между минимальным и максимальным значениями. Если среднее арифметическое пяти чисел меньше среднего между минимальным и максимальным числами, увеличим счётчик на единицу.

Выведем итоговое количество строк, удовлетворяющих условиям задачи.

```
# Открытие файла
f = open('9-159.txt')
k = 0
# Чтение файла построчно
for s in f:
    a = sorted([int(x) for x in s.split()])
# Среднее арифметическое всех чисел считается как sum(a)/len(a)
# среднее между минимальным и максимальным числами как (a[0]+a[4])/2
    if sum(a)/len(a) < (a[0]+a[4])/2:
        k += 1
print(k)
```

Результат работы программы:

2203

Ответ: 2203

Задача № 3 (5798)

В файле электронной таблицы [9-181.xls](#) в каждой строке содержатся четыре числа. Определите, какое количество четвёрок удовлетворяют условию, что наибольшее число является делителем произведения остальных.

Ссылка на видео-разбор с таймингом: https://vk.com/video-205546952_456241211?t=35m55s

Решение

Копируем содержимое прилагаемого файла, вставляем его в блокнот и сохраняем на рабочем столе или любой другой удобной директории.

Обнулим переменную k, которая будет использоваться для подсчета строк, соответствующих условиям.

Откроем файл (9-181.txt) и прочитаем каждую строку файла использованием цикла for и разобьём её на отдельные числа с помощью метода split() и сразу отсортируем список чисел с помощью встроенной функции sorted().

Проверка условия

Если произведение первых трёх чисел из строки делится на четвёртое число без остатка, то увеличиваем счётчик на единицу. В конце выводим количество строк, удовлетворяющих условию.

```
k = 0
# Чтение файла построчно
for s in open('9-181.txt'):
    a = sorted([int(x) for x in s.split()])
    if (a[0]*a[1]*a[2]) % a[3] == 0:
        k += 1
print(k)
```

Результат работы программы:

49

Ответ: 49

Задача № 4 (5522)

В файле электронной таблицы [9-170.xls](#) в каждой строке содержатся шесть неотрицательных целых чисел. Определите количество строк таблицы, для которых выполнены оба условия:

- в строке только одно число повторяется трижды (ровно 3 раза), остальные числа не повторяются;
- утроенная сумма неповторяющихся чисел строки не больше произведения повторяющихся чисел.

Ссылка на видео-разбор с таймингом: https://vk.com/video-205546952_456241211?t=43m35s

Решение

Копируем содержимое прилагаемого файла, вставляем его в блокнот и сохраняем на рабочем столе или любой другой удобной директории.

Обнулим переменную k, которая будет использоваться для подсчета строк, соответствующих условиям.

Откроем файл (9-170.txt) и прочитаем каждую строку файла использованием цикла for и разобьём её на отдельные числа с помощью метода split() и сразу отсортируем список чисел с помощью встроенной функции sorted().

Создадим список a1, который включает все неповторяющиеся числа из списка a. Метод count(x) определяет, сколько раз число x встречается в списке a. Если число встречается ровно один раз, оно добавляется в список a1.

Список a3 заполняется числами, которые повторяются ровно три раза в списке a.

Проверка условий:

1. Длина списка a1 должна быть равной 3, что означает, что в строке есть три неповторяющихся числа.
2. Длина списка a3 также должна быть 3, что указывает на наличие одного числа, которое повторяется три раза.
3. Утроенная сумма неповторяющихся чисел не должна превышать куб числа, которое повторяется три раза.

Если все условия выполняются, увеличивается счётчик на единицу. В конце выводится количество строк, которые соответствуют всем необходимым условиям.

```
k = 0
for s in open('9-170.txt'):
    a = [int(x) for x in s.split()]
# Определение неповторяющихся чисел
    a1 = [x for x in a if a.count(x)==1]
# Определение чисел, которые повторяются в списке ровно три раза
    a3 = [x for x in a if a.count(x)==3]
#проверка условий
    if len(a1)==3 and len(a3)==3 and 3*sum(a1) <= a3[0]**3:
        k += 1
print(k)
```

Результат работы программы:

134

Ответ: 134

Задача №5 (5526)

В файле электронной таблицы [9-170.xls](#) в каждой строке содержатся шесть неотрицательных целых чисел. Определите количество строк таблицы, для которых выполнены оба условия:

- в строке только одно число повторяется дважды (ровно 2 раза), остальные числа не повторяются;
- сумма максимального и минимального среди неповторяющихся чисел строки не больше суммы повторяющихся чисел.

Ссылка на видео-разбор с таймингом: https://vk.com/video-205546952_456241211?t=1h1m30s

Решение

Копируем содержимое прилагаемого файла, вставляем его в блокнот и сохраняем на рабочем столе или любой другой удобной директории.

Обнулим переменную k, которая будет использоваться для подсчета строк, соответствующих условиям.

Откроем файл (9-170.txt) и прочитаем каждую строку файла использованием цикла for и разобьём её на отдельные числа с помощью метода split().

Создадим отсортированный список a1, включающий все неповторяющиеся числа из списка a. Метод count(x) определяет, сколько раз число x встречается в списке a. Если число встречается ровно один раз, оно добавляется в список a1.

В список a2 будем записывать числа, которые повторяются ровно два раза в списке a.

Здесь проверяются несколько условий:

1. Длина списка a1 должна быть равной 4, что означает, что в строке есть четыре неповторяющихся числа.
2. Длина списка a2 должна равняться 2, что указывает на наличие двух чисел, повторяющихся дважды.
3. Сумма минимального и максимального чисел из списка a1 не должна превышать сумму чисел, встречающихся дважды.

Если все эти условия соблюдаются, счётчик k увеличивается на 1, что свидетельствует о том, что данная строка соответствует требованиям задачи. В конце выводится количество строк, которые соответствуют всем необходимым условиям.

```
k = 0
for s in open('9-170.txt'):
    a = [int(x) for x in s.split()]
    a1 = sorted([x for x in a if a.count(x)==1])
    a2 = [x for x in a if a.count(x)==2]
    if len(a1)==4 and len(a2)==2 and a1[0]+a1[3] <= sum(a2):
        k += 1
print(k)
```

Результат работы программы

1159

Ответ: 1159

Задание №6 (5761)

В файле электронной таблицы [9-177.xls](#) в каждой строке содержатся восемь неотрицательных чисел. Определите количество строк таблицы, для которых выполнены оба условия:

- квадрат суммы максимального и минимального числа больше суммы квадратов других чисел;
- в строке есть хотя бы одно повторяющееся число.

Ссылка на видео-разбор с таймингом: https://vk.com/video-205546952_456241211?t=1h8m20s

Решение

Копируем содержимое прилагаемого файла, вставляем его в блокнот и сохраняем на рабочем столе или любой другой удобной директории.

Обнулим переменную *k*, которая будет использоваться для подсчета строк, соответствующих условиям.

Откроем файл (9-177.txt) и прочитаем каждую строку файла использованием цикла *for* и разобьём её на отдельные числа с помощью метода *split()* и сразу отсортируем список чисел с помощью встроенной функции *sorted()*. Создадим список *a2*, который включает числа, которые встречаются в списке *a* более одного раза. Метод *a.count(x)* считает количество вхождений числа *x* в список *a*, и если число встречается более одного раза, оно добавляется в список.

Проверка условий:

Сравнивается квадрат суммы максимального и минимального чисел с суммой квадратов остальных чисел. Т.к. строка отсортирована первое и последнее числа будут являться минимальное и максимальное соответственно числом в строке. Если длина множества уникальных чисел меньше длины списка чисел, значит, есть повторяющиеся числа.

Итоговый результат: Если оба условия выполнены, увеличивается счётчик. В конце выводится количество строк, удовлетворяющих обоим условиям.

```
k = 0
for s in open('9-177.txt'):
    a = sorted([int(x) for x in s.split()])
    a2 = [x for x in a if a.count(x)>1]
    if (a[0]+a[7])**2 > a[1]**2+a[2]**2+a[3]**2+a[4]**2+a[5]**2+a[6]**2 and len(a2)>0:
        k+=1
print(k)
```

Результат работы программы:

98

Ответ: 98

Задание № 7(6140)

В файле электронной таблицы [9-194.xls](#) в каждой строке записаны 5 натуральных чисел. Определите количество строк таблицы, для которых выполнены все условия:

- в строке нет повторяющихся чисел;
- чётных чисел больше, чем нечётных;
- сумма чётных чисел меньше, чем сумма нечётных.

Ссылка на видео-разбор с таймингом: https://vk.com/video-205546952_456241211?t=1h16m55s

Решение

Копируем содержимое прилагаемого файла, вставляем его в блокнот и сохраняем на рабочем столе или любой другой удобной директории.

Обнулим переменную *k*, которая будет использоваться для подсчета строк, соответствующих условиям.

Откроем файл (9-170.txt) и прочитаем каждую строку файла использованием цикла *for* и разобьём её на отдельные числа с помощью метода *split()*.

Создадим список a0, который включает все числа из списка a, которые делятся на 2 без остатка (чётные числа) и список a1 состоящий из всех чисел, которые не делятся на 2 нацело (нечётные числа).

Проверка условий:

1. Необходимо проверить отсутствие повторяющихся чисел в строке. Для этого можно использовать множество (set), так как оно содержит только уникальные элементы.
2. Сравнить количество чётных и нечётных чисел.
3. Вычислить и сравнить суммы чётных и нечётных чисел.

Если все условия выполняются, счётчик увеличивается на единицу. В конце выводится количество строк, удовлетворяющих всем трём условиям.

```
k = 0
for s in open('9-194.txt'):
    a = [int(x) for x in s.split()]
    a0 = [x for x in a if x%2==0]
    a1 = [x for x in a if x%2!=0]
    if len(a) == len(set(a)) and len(a0)>len(a1) and sum(a0)<sum(a1):
        k += 1
print(k)
```

Результат работы программы:

76

Ответ: 76

Задание №8 (6818)

В файле электронной таблицы [9-226.xls](#) в каждой строке записаны семь натуральных чисел. Определите сумму чисел в строке таблицы с наименьшим номером, для которой выполнены оба условия:

- в строке есть два числа, которые повторяются дважды, остальные три числа различны;
- максимальное число строки не повторяется. В ответе запишите только число.

Ссылка на видео-разбор с таймингом: https://vk.com/video-205546952_456241211?t=1h25m25s

Решение

Копируем содержимое прилагаемого файла, вставляем его в блокнот и сохраняем на рабочем столе или любой другой удобной директории.

Переменная k в этой задаче будет использоваться как номер строки, которая обрабатывается в цикле, поэтому увеличение её на 1 будет происходить с началом каждого повтора.

Откроем файл (9-170.txt) и увеличивая счетчик строк, прочитаем каждую строку файла использованием цикла for и разобьём её на отдельные числа с помощью метода split().

Создадим списки a1, который включает все числа из списка a, встречающиеся в списке a ровно один раз

и список a2 состоящий из чисел, которые повторяются ровно два раза в списке a

Проверка условий

Если в строке содержится три неповторяющихся числа и четыре числа, повторяющиеся дважды, и максимальное число из списка a1 присутствует в строке, то выводится сумма чисел a.

```
k = 0
for s in open('9-226.txt'):
    k += 1
    a = [int(x) for x in s.split()]
    a1 = [x for x in a if a.count(x)==1]
    a2 = [x for x in a if a.count(x)==2]
    if len(a1)==3 and len(a2)==4 and max(a) in a1:
        print(sum(a))
```

Вывод результата:

261

Ответ: 261

Задание №9(7348)

В файле электронной таблицы [9-228.xls](#) в каждой строке записаны шесть натуральных чисел.

Определите количество строк таблицы, для которых выполнены следующие условия:

- в строке есть повторяющиеся числа;
- максимальное число в строке не повторяется;
- сумма всех повторяющихся чисел в строке меньше максимального числа этой строки.

При подсчёте суммы повторяющихся чисел каждое число учитывается столько раз, сколько оно встречается. В ответе запишите только число.

Ссылка на видео-разбор с таймингом: https://vk.com/video-205546952_456241211?t=1h37m45s

Решение

Копируем содержимое прилагаемого файла, вставляем его в блокнот и сохраняем на рабочем столе или любой другой удобной директории.

Обнулим переменную k, которая будет использоваться для подсчета строк, соответствующих условиям.

Откроем файл (9-177.txt) и прочитаем каждую строку файла использованием цикла for и разобьём её на отдельные числа с помощью метода split() и сразу отсортируем список чисел с помощью встроенной функции sorted().

Создадим список a2, который включает все числа строки, которые повторяются более одного раза.

Это нужно для проверки наличия повторяющихся чисел.

Проверка условий.

Проверим, что в строке есть повторяющиеся числа, т.е. длина списка a2 больше нуля, максимальное (a[5]) число уникально, и суммарная величина чисел, повторившихся более одного раза, не превышает максимального (a[5]). Если эти условия выполняются, то счётчик увеличивается на одну единицу. В конце выводится количество строк, удовлетворяющих всем трём условиям.

```
k = 0
for s in open('9-228.txt'):
    a = sorted([int(x) for x in s.split()])
    a2 = [x for x in a if a.count(x)>1]
    if len(a2)>0 and a.count(a[5])==1 and sum(a2) < a[5]:
        k += 1
```

```
print(k)
```

Результат работы программы:

882

Ответ: 882

Telegram: @fast_ege