

CASE 2:

(2) УСЛОВИЕ: =>

В файле содержится последовательность целых чисел. Элементы последовательности могут принимать значения от 0 до 10 000 включительно. Определите количество "троек" чисел (под "тройкой чисел" подразумеваются три идущих подряд числа), в которой только два из трех элементов меньше максимального числа в файле, кратного 18 и являющегося палиндромом, и при этом все три элемента составляют арифметическую прогрессию (порядок чисел в "тройке" рассматривается строго как в файле, без перестановок. Вид прогрессии не важен: убывающая или возрастающая). В ответе запишите количество найденных "троек", затем медианное значение из сумм элементов всех подходящих "троек".

CODE 2:

```
def isPal(x):
    s = str(x); f = True
    for i in range((len(s)+1)//2):
        if s[i] != s[len(s)-i-1]:
            f = False
            break
    return f

p = [int(x) for x in open('17_file_2.txt')]; l = []
count = 0
mx18 = max(x for x in p if isPal(x) and x%18==0)

for a,b,c in zip(p, p[1:], p[2:]):
    if (b-a) == (c-b) and (int(a<mx18)+int(b<mx18)+int(c<mx18)) == 2:
        count += 1
        l += [a+b+c]
l.sort()
if len(l)%2==0:
    print(count, ( l[len(l)//2] + l[len(l)//2-1] ) / 2)
else:
    print(count, l[len(l)//2])

# 116 18996.0
```

EXPLANATION 2:

(2) РЕШЕНИЕ: =>

1. Заполнить список числами из файла (воспользуемся более короткой записью, характерной для python).
 2. Инициализировать все необходимые переменные (count, mx18) + доп. список для найденных троек l.
 3. Пройдясь по заполненному списку, находим макс. число, кратное 18 и являющееся палиндромом. Здесь используем генератор - объект, который сразу при создании не вычисляет значения всех своих элементов, так как хранит в памяти только последний вычисленный элемент. Вычисления с помощью генераторов называются "ленивыми" (они экономят память), из-за чего нам и импонирует данная реализация. При этом оборачиваем все это в max(), который и вычислит максимальное значение в списке. Дополнительно реализуем функцию isPal для проверки, является ли число палиндромом: число -> строка -> сдвигаем "указатели на символы строки" по краям навстречу друг другу до тех пор, пока они не будут стоять рядом / указывать на один и тот же символ. Если условие равенства символов не будет нарушено, мы вернем True, иначе меняем флаг f на False и выходим из цикла с помощью break -> return f.
 4. Проходимся по списку кортежей (из трех элементов) и производим все необходимые операции: 1) разность между [первым и вторым элементами] d1 и [вторым и третьим элементами] d2 равны (разность арифметической прогрессии = $\Delta(a) = a_2 - a_1$); 2) сумма "булей, приведенных к целочисленным переменным" == 2, так как только два элемента из трех должны быть меньше mx18. Если нашли такую "тройку", то счетчик увеличиваем на 1 и заносим сумму элементов (a+b+c) кортежа в наш список l.
 5. Сортируем l в порядке возрастания после завершения прохода списка p, используя встроенный функционал python (default: по возрастанию).
 6. Если кол-во элементов в l нечет., то берем центральный элемент, иначе складываем два соседних центральных элемента и полученную сумму делим на 2 (полусумма).
 7. Записываем ответ.
- * Медианой ряда чисел называется число, стоящее посередине упорядоченного по возрастанию ряда чисел (если количество чисел нечетное). Если же количество чисел в ряду четно, то медианой ряда является полусумма двух стоящих посередине чисел упорядоченного по возрастанию ряда.