

Задание 17. Обработка числовой последовательности на языке Python 3.9

ФМШ МАИ Занятие № 2
Дмитриев Андрей

Задача 1

Рассматривается множество целых чисел, принадлежащих числовому отрезку [331; 8751], которые удовлетворяют следующим условиям:

- количество цифр в десятичной и шестнадцатеричной записях одинаковое;
- кратны 5, но не 25.

Найдите количество таких чисел и максимальное из них.

```
print('mai_zad1.txt')
count=0
max=0
for i in range(331,8751+1):
    b=len(str(i))
    c=len(hex(i)[2:])
    if b==c and i%5==0 and
i%25!=0:
        count+=1
        max=i
print(count, max)
```

Задача 2

Назовём натуральное число подходящим, если у него больше 17 различных делителей (включая единицу и само число). Определите количество подходящих чисел, принадлежащих отрезку $[10\ 001; 50\ 000]$, а также наименьшее из таких чисел. В ответе запишите два целых числа: сначала количество, затем наименьшее число.

```
print(' mai_zad2.txt ')
def F(n):
    delitel=2
    countdel=2
    while delitel ** 2 < n:
        if n % delitel == 0:
            countdel += 2
            if countdel > 17:
                return 1
        delitel += 1
count=0
min=50001
for i in range(10001,50000+1):
    if F(i)==1:
        count += 1
        if i < min:
            min = i
print(count, min)
```

Задача 3

Назовём натуральное число подходящим, если у него ровно 3 различных простых делителя. Например, число 180 подходящее (его простые делители – 2, 3 и 5), а число 12 – нет (у него только два различных простых делителя).

Определите количество подходящих чисел, принадлежащих [10 001; 50 000], а также наименьшее из таких чисел. В ответе запишите два целых числа: сначала количество, затем наименьшее число.

```
print('mai_zad3')
def F(n):
    delitel=2
    while delitel ** 2 <= n:
        if n % delitel == 0:
            return False
        delitel += 1
    return True
def G(n):
    count = 0
    delitel=2
    while delitel ** 2 <= n:
        if n % delitel == 0:
            if F(delitel):
                count +=1
            if F(n // delitel):
                count +=1
            if count > 3:
                break
        delitel += 1
    if delitel ** 2 == n:
        count +=1
    return count

kol=0
min=50001
for i in range(10001,50000+1):
    if G(i) == 3:
        kol += 1
        if i < min:
            min = i
print(kol, min)
```

Задача 4

Файл содержит последовательность неотрицательных целых чисел, не превышающих 10 000. Назовём парой два идущих подряд элемента последовательности. Определите количество пар, в которых хотя бы один из двух элементов делится на 3 и хотя бы один из двух элементов меньше среднего арифметического всех чётных элементов последовательности. В ответе запишите два числа: сначала количество найденных пар, а затем – максимальную сумму элементов таких пар. Например, в последовательности (3 8 9 4) есть две подходящие пары: (3 8) и (9 4), в ответе для этой последовательности надо записать числа 2 и 13.

```
print('mai_zad4')
spisok = []
with open('17.txt') as f:
    for n in f:
        spisok.append(int(n))
kol = 0
summ = 0
for i in range(len(spisok)):
    if spisok[i] % 2 == 0:
        kol += 1
        summ += spisok[i]
sred = summ // kol
count = 0
maxi = 0
for i in range(len(spisok) - 1):
    if (spisok[i] % 3 == 0 or spisok[i + 1] % 3
    == 0) and (spisok[i] < sred or spisok[i + 1] <
    sred):
        count += 1
        maxi = max(maxi, spisok[i] + spisok[i +
    1])
print(count, maxi)
```

Задача 5

В файле содержится последовательность целых чисел. Элементы последовательности могут принимать целые значения от $-10\,000$ до $10\,000$ включительно. Определите и запишите в ответе сначала количество пар элементов последовательности, в которых хотя бы одно число делится на 3, затем максимальную из сумм элементов таких пар. В данной задаче под парой подразумевается два идущих подряд элемента последовательности. Например, для последовательности из пяти элементов: 6; 2; 9; -3; 6 — ответ: 4 11.

Будем последовательно считывать числа из файла. Для каждой пары (двух подряд идущих элементов) будем проверять, делится ли хотя бы одно число из пары на 3. При успешном выполнении условия будем увеличивать значения счётчика count и проверять, больше ли сумма элементов пары текущей максимальной суммы. Если сумма элементов пары больше текущей максимальной суммы, будем обновлять значение переменной max.

```
f=open('17.txt')

count=0
sm=-20001

n1=int(f.readline())

for s in f.readlines():
    n2=int(s)

    if n1%3==0 or n2%3==0:
        count=count+1
        sm=max(n1+n2, sm)

    n1=n2

print(count)
print(sm)
```

Задача 6

В файле содержится последовательность из 10 000 целых положительных чисел. Каждое число не превышает 10 000. Определите и запишите в ответе сначала количество пар элементов последовательности, разность которых четна и хотя бы одно из чисел делится на 31, затем максимальную из сумм элементов таких пар. В данной задаче под парой подразумевается два различных элемента последовательности. Порядок элементов в паре не важен.

```
with open('17.txt', 'r') as f:
    numbers = [int(x) for x in f.read().split('\n') if x]

count = 0
maximum = -1
for i in range(len(numbers)-1):
    for j in range(i + 1, len(numbers)):
        if abs(numbers[i] - numbers[j]) % 2 == 0 and
            (numbers[i] % 31 == 0 or numbers[j] % 31 == 0):
            count += 1
            maximum = max(maximum, numbers[i] +
                numbers[j])
print(count)
print(maximum)
```

Задача 7

В файле содержится последовательность из 10 000 натуральных чисел. Каждое число не превышает 10 000. Определите и запишите в ответе сначала количество пар элементов последовательности, у которых различные остатки от деления на $d = 160$ и хотя бы одно из чисел делится на $p = 7$, затем максимальную из сумм элементов таких пар. В данной задаче под парой подразумевается два различных элемента последовательности. Порядок элементов в паре не важен.

```
a = [int(input()) for _ in range(int(input()))]
p = []
e = []
gh = dict()
for i in range(len(a)):
    for j in range(len(a)):
        if (a[i] % 160 != a[j] % 160) and (a[i] % 7 == 0
or a[j] % 7 == 0):
            e.append(a[i])
            e.append(a[j])
h = []
if len(e) == 0:
    print('00')

for i in range(0, len(e), 2):
    if i != len(e):
        h.append(e[i] + e[i + 1])
        k = str(e[i]) + ' ' + str(e[i + 1])
        p.append(k)
        gh[k] = e[i] + e[i + 1]
fg = max(h)
t = 0
for i in gh:
    if gh[i] == fg:
        t = i
print(t)
```