

ЕГЭ-17 (14 минут)^a

Обработка числовой последовательности



Содержание

17.1 Создание списков на основе заданного списка	3
Задания для самостоятельной работы	4
Задания для дополнительной отработки	7
17.2 Примеры решения задач. Часть I (простые задачи)	10
Задания для самостоятельной работы	13
Задания для дополнительной отработки	14
17.3 Примеры решения задач. Часть II (предварительный простой шаг)	15
Задания для самостоятельной работы	16
Задания для дополнительной отработки	16
17.4 Примеры решения задач. Часть III («двухэтапные» решения)	17
Задания для самостоятельной работы	18
Задания для дополнительной отработки	18
17.5 Примеры решения задач. Часть IV (пары чисел, простой предварительный шаг)	20
Задания для самостоятельной работы	21
Задания для дополнительной отработки	22
17.6 Примеры решения задач. Часть V (пары чисел, «двухэтапное» решение)	23
Задания для самостоятельной работы	25
Ответы	27
Приложение 1	30
Основные понятия	30
Работа со списком целиком	31
Работа с фрагментами списка	33
Перебор элементов списка	34
Приложение 2	36
Целочисленные операции	36
Цифры натурального числа	36
Цифры целого отрицательного числа	37
Части натурального числа	37

^aАвтор: Е. В. Ширяева, <https://t.me/EVShiryaeva>, 2025. <https://t.me/EVShiryaeva>. Нулевой курс мехмата ЮФУ (группы). Проект «Отличный код» (мини-группы, индивидуальные занятия, https://t.me/project_OK).

Цифры натурального числа, записанного в r -ной системе счисления	38
--	----

17.1 Создание списков на основе заданного списка

Синтаксис генератора списка

[eval for x in a]

eval — выражение для получения элемента списка; x — переменная для элемента списка; a — исходный список (задан некоторым способом).

Пример использования генератора списка

```
a = [34, -90, 56, 19, 21, 34, -78, 0]

# Создать новый список, в котором все элементы будут иметь противоположный знак
a1 = [-x for x in a]

# вывод нового списка на экран
print(a1) # [-34, 90, -56, -19, -21, -34, 78, 0]
```

Синтаксис генератора списка с условием отбора condition

[eval for x in a if condition]

eval — выражение для получения элемента списка; x — переменная для элемента списка; a — исходный список; condition — условие отбора.

Примеры использования генераторов списков с условием

```
a = [34, -90, 56, 19, 21, 34, -78, 0]

# Создать список с положительными элементами
a2 = [x for x in a if x > 0]
print(a2) # [34, 56, 19, 21, 34]

# Создать список с чётными элементами
a3 = [x for x in a if x % 2 == 0]
print(a3) # [34, -90, 56, 34, -78, 0]

# Создать список с положительными элементами, кратными трём
a4 = [x for x in a if x > 0 and x % 3 == 0]
print(a4) # [21]

# Создать список с положительными элементами, двоичная запись которых
# содержит нечётное количество единиц
a5 = [x for x in a if x > 0 and bin(x).count('1') % 2 != 0]
print(a5) # [56, 19, 21]
```

Задания для самостоятельной работы

№ 17.1 (→). Для заданного списка *a* запишите генераторы списков для отбора значений, удовлетворяющих указанным в заданиях условиям.

 После операторов печати в коде стоят ожидаемые ответы для сравнения.

Допишите генераторы списков

```
a = [3, 24, -56, 10, 201, 234, -78, 0, 113, -45, 61]

# Создать список с удвоенными элементами списка a
a1 = [Здесь должен стоять Ваш код]
print(a1) # [6, 48, -112, 20, 402, 468, -156, 0, 226, -90, 122]

# Создать список из последних цифр каждого элемента списка
a2 = []
print(a2) # [3, 4, 6, 0, 1, 4, 8, 0, 3, 5, 1]

# Создать список умножением каждого элемента списка a на его I элемент
a3 = []
print(a3) # [9, 72, -168, 30, 603, 702, -234, 0, 339, -135, 183]
```

№ 17.2 (→). Для заданного списка *a* запишите генераторы списков для отбора значений, удовлетворяющих указанным в заданиях условиям.

 После операторов печати в коде стоят ожидаемые ответы для сравнения.

Допишите генераторы списков

```
a = [3, 24, 8, 1234, 9, 0, 127, 88]

# Создать список, в котором все элементы списка a переведены в 8-ную с.с.
a1 = [Здесь должен стоять Ваш код]
print(a1) # ['3', '30', '10', '2322', '11', '0', '177', '130']

# Создать список, в котором все элементы списка a переведены в 16-ную с.с.
a2 = []
print(a2) # ['3', '18', '8', '4d2', '9', '0', '7f', '58']

# Создать список с количеством цифр в 2-ной записи каждого элемента списка a
a3 = []
print(a3) # [2, 5, 4, 11, 4, 1, 7, 7]
```

№ 17.3 (→). Дан список строк, представляющих собой двоичные числа. Для данного списка запишите генераторы списков для отбора значений, удовлетворяющих указанным в заданиях условиям.

☝ После операторов печати в коде стоят ожидаемые ответы для сравнения.

Допишите генераторы списков

```
a = ['1011', '101', '11101', '1101011', '1001', '111000110010']

# Создать список, в котором все элементы списка a заменены длиной двоичного числа
a1 = [Здесь должен стоять Ваш код]
print(a1) # [4, 3, 5, 7, 4, 12]

# Создать список, в котором все элементы списка a переведены из 2-ной с.с. в 10-ную
a2 = []
print(a2) # [11, 5, 29, 107, 9, 3634]

# Создать список, в котором все элементы списка a переведены из 2-ной с.с. в 8-ную
a3 = []
print(a3) # ['13', '5', '35', '153', '11', '7062']

# Создать список, в котором все элементы списка a переведены из 2-ной с.с. в 16-ную
a4 = []
print(a4) # ['b', '5', '1d', '6b', '9', 'e32']
```

№ 17.4 (→). Для заданного списка *a* запишите генераторы списков для отбора значений, удовлетворяющих указанным в заданиях условиям.

☝ После операторов печати в коде стоят ожидаемые ответы для сравнения.

Допишите генераторы списков

```
a = [3, 24, -56, 10, 201, 234, -78, 0, 113, -45, 61]

# Создать список с двузначными элементами списка a
a1 = [Здесь должен стоять Ваш код]
print(a1) # [24, -56, 10, -78, -45, 61]

# Создать список с неположительными элементами списка a, увеличенными на 100
a2 = []
print(a2) # [44, 22, 100, 55]

# Создать список с отрицательными элементами списка a, поделенными нацело на 3
a3 = []
print(a3) # [-19, -26, -15]
```

№ 17.5 (→). Для заданного списка строк *a* запишите генераторы списков для отбора значений, удовлетворяющих указанным в заданиях условиям.

💬 После операторов печати в коде стоят ожидаемые ответы для сравнения.

Допишите генераторы списков

```
a = ['194', '26', '11101', '1101013', '1001', '78', '37']

# Создать список из элементов списка a с длиной не более 4 символов
a1 = [Здесь должен стоять Ваш код]
print(a1) # ['194', '26', '1001', '78', '37']

# Создать список из элементов списка a, которые
#                                     НЕ могут представлять собой восьмеричную запись
a2 = []
print(a2) # ['194', '78']

# Создать список из элементов списка a, которые представляют собой
#                                     правильную запись двоичного числа
a3 = []
print(a3) # ['11101', '1001']
```

Задания для дополнительной отработки

№ 17.6 (→). Для заданного списка *a* запишите генераторы списков для отбора значений, удовлетворяющих указанным в заданиях условиям.

💬 После операторов печати в коде стоят ожидаемые ответы для сравнения.

_____ Допишите генераторы списков _____

```
a = [-34, -90, 56, 19, 21, 134, -78, 0, 11, -345, 61, 84]

# Создать список из элементов списка a, умноженных на свою последнюю цифру
a1 = [Здесь должен стоять Ваш код]
print(a1) # [-136, 0, 336, 171, 21, 536, -624, 0, 11, -1725, 61, 336]

# Создать список из элементов списка a, умноженных на последнюю цифру I элемента
a2 = []
print(a2) # [-136, -360, 224, 76, 84, 536, -312, 0, 44, -1380, 244, 336]

# Создать список из элементов списка a, увеличенных на значение последнего элемента
a3 = []
print(a3) # [50, ...] Проверьте свой результат самостоятельно
```

№ 17.7 (→). Для заданного списка *a* запишите генераторы списков для отбора значений, удовлетворяющих указанным в заданиях условиям.

💬 После операторов печати в коде стоят ожидаемые ответы для сравнения.

_____ Допишите генераторы списков _____

```
a = [30, 7, 32, 2050, 19, 63, 98, 13]

# Создать список, в котором все элементы списка a записаны в 2-ной с.с.
a1 = [Здесь должен стоять Ваш код]
print(a1) # ['11110', '111', '100000', '100000000010', '10011', ...]

# Создать список, в котором все элементы списка a записаны в 8-ной с.с.
a2 = []
print(a2) # Проверьте свой результат самостоятельно

# Создать список из элементов списка a заменой каждого элемента
#                               на количество цифр в его 16-ной записи
a3 = []
print(a3) # Проверьте свой результат самостоятельно
```

№ 17.8 (→). Дан список строк, представляющих собой двоичные числа. Для данного списка запишите генераторы списков для отбора значений, удовлетворяющих указанным в заданиях условиям.

Допишите генераторы списков

```
a = ['10101', '10100', '100110', '1100110', '10011', '11001101101']

# Создать список, в котором все элементы списка a переведены из 2-ной с.с. в 10-ную
a1 = [Здесь должен стоять Ваш код]
print(a1) # Проверьте свой результат самостоятельно

# Создать список, в котором все элементы списка a переведены из 2-ной с.с. в 8-ную
a2 = []
print(a2) # Проверьте свой результат самостоятельно

# Создать список, в котором все элементы списка a переведены из 2-ной с.с. в 16-ную
a3 = []
print(a3) # Проверьте свой результат самостоятельно

# Создать список, в котором все элементы списка a заменены кол-вом пар 11 в строке
a4 = []
print(a4) # Проверьте свой результат самостоятельно
```

№ 17.9 (→). Для заданного списка строк *a* запишите генераторы списков для отбора значений, удовлетворяющих указанным в заданиях условиям.

Допишите генераторы списков

```
a = ['100', '194', '26', '111', '1013', '10101', '98', '37', '8']

# Записать в новый список элементы списка a, которые
#                                     НЕ могут представлять собой 8-ую запись
a1 = [Здесь должен стоять Ваш код]
print(a1) # Проверьте свой результат самостоятельно

# Записать в новый список все элементы списка a с длиной не более двух символов
a2 = []
print(a2) # Проверьте свой результат самостоятельно

# Записать в новый список элементы списка a, которые представляют собой 2-ую запись
a3 = []
print(a3) # Проверьте свой результат самостоятельно
```

№ 17.10 (→). Для заданного списка *a* запишите генераторы списков для отбора значений, удовлетворяющих указанным в заданиях условиям.

 После операторов печати в коде стоят ожидаемые ответы для сравнения.

Допишите генераторы списков

```
a = [-34, -90, 56, 19, 21, 134, -78, 0, 11, -345, 61, 84]

# Создать список с неположительными элементами списка a
a1 = [Здесь должен стоять Ваш код]
print(a1) # Проверьте свой результат самостоятельно

# Создать список с элементами списка a, кратными трем
a2 = []
print(a2) # Проверьте свой результат самостоятельно

# Создать список с положительными чётными элементами списка a
a3 = []
print(a3) # Проверьте свой результат самостоятельно

# Создать список с положительными элементами, двоичная запись которых
# содержит нечётное количество значащих нулей
a4 = []
print(a4) # [56, 134, 11, 61]

# Создать список с положительными элементами списка a, запись которых в 8-ной с.с.
# содержит хотя бы одну двойку
a5 = []
print(a5) # [19, 21, 134, 84]
```

17.2 Примеры решения задач. Часть I (простые задачи)

Пример 17.1. В файле 17_ev1.txt содержится последовательность целых чисел. Элементы последовательности могут принимать целые значения от 0 до 10 000 включительно. Рассматривается множество элементов последовательности, которые удовлетворяют следующим условиям:

- остаток от деления на 14 равен 3;
- последняя цифра числа больше 5.

Найдите количество таких чисел и их сумму.

Решение. Условие «остаток от деления на 14 равен 3»: $x \% 14 == 3$. Условие «последняя цифра числа больше 5»: $x \% 10 > 5$. Должны выполняться оба условия (используется операция and).

Решение-1. Алгоритмы поиска количества и суммы

```
with open('17_ev1.txt') as ff:
    a = [int(x) for x in ff.read().split()]

cnt, sm = 0, 0
for x in a:
    if x % 14 == 3 and x % 10 > 5:
        cnt += 1
        sm += x

print(cnt, sm) # 59 293505 :)
```

Решение-2. Создание списка подходящих чисел

```
with open('17_ev1.txt') as ff:
    a = [int(x) for x in ff.read().split()]

t = []
for x in a:
    if x % 14 == 3 and x % 10 > 5:
        t += [x]

print(len(t), sum(t)) # 59 293505 :)
```

Решение-3. Использование генератора списка

```
with open('17_ev1.txt') as ff:
    a = [int(x) for x in ff.read().split()]

t = [x for x in a if x % 14 == 3 and x % 10 > 5]

print(len(t), sum(t)) # 59 293505 :)
```

Пример 17.2. В файле 17_ev1.txt содержится последовательность целых чисел. Элементы последовательности могут принимать целые значения от 0 до 10 000 включительно. Рассматривается множество элементов последовательности, двоичная запись которых содержит нулей больше, чем единиц.

Найдите количество таких чисел и максимальное из них.

Решение-1. Алгоритмы поиска количества и максимума

```
with open('17_ev1.txt') as ff:
    a = [int(x) for x in ff.read().split()]

cnt = 0
mx = 0 - 1 # [0; 10_000] берём нереально малое число

for x in a:
    x2 = bin(x)[2:]
    if x2.count('0') > x2.count('1'):
        cnt += 1
        if mx < x:
            mx = x

print(cnt, mx) # 746 9777 :)
```

Решение-2. Создание списка подходящих чисел

```
with open('17_ev1.txt') as ff:
    a = [int(x) for x in ff.read().split()]

t = []
for x in a:
    x2 = bin(x)[2:]
    if x2.count('0') > x2.count('1'):
        t += [x]

print(len(t), max(t)) # 746 9777 :)
```

Пример 17.3. В файле 17_ev2.txt содержится последовательность целых чисел. Элементы последовательности могут принимать целые значения от $-10\,000$ до $10\,000$ включительно. Рассматривается множество элементов последовательности, которые удовлетворяют следующим условиям:

- число оканчивается тройкой;
- сумма цифр числа кратна трём.

Найдите количество таких чисел и минимальное из них.

Решение. При проверке условия «число оканчивается тройкой» надо обязательно учитывать, что числа могут быть отрицательными.

Условие «сумма цифр числа кратна трём» может быть заменено более простым «число кратно трём» (Критерий делимости на 3: целое число делится на 3 без остатка, если сумма входящих в его состав цифр делится на 3).

Решение-1. Алгоритмы поиска количества и минимума

```
with open('17_ev2.txt') as ff:
    a = [int(x) for x in ff.read().split()]

cnt = 0
mn = 10_000 + 1 # [-10_000; 10_000]    выходим за правую границу отрезка
for x in a:
    if abs(x) % 10 == 3 and x % 3 == 0:
        cnt += 1
        if mn > x:
            mn = x

print(cnt, mn) # 58 -6843 :)
```

Решение-2. Создание списка подходящих чисел

```
with open('17_ev2.txt') as ff:
    a = [int(x) for x in ff.read().split()]

t = []
for x in a:
    if abs(x) % 10 == 3 and x % 3 == 0:
        t += [x]

print(len(t), min(t)) # 58 -6843 :)
```

Решение-3. Создание списка подходящих чисел. Использование генератора списка

```
with open('17_ev2.txt') as ff:
    a = [int(x) for x in ff.read().split()]

t = [x for x in a if abs(x) % 10 == 3 and x % 3 == 0]

print(len(t), min(t)) # 58 -6843 :)
```

Пример 17.4. В файле 17_ev2.txt содержится последовательность целых чисел. Элементы последовательности могут принимать целые значения от $-10\,000$ до $10\,000$ включительно. Рассматривается множество элементов последовательности, которые удовлетворяют только одному из условий:

- число однозначное или двузначное;
- последние две цифры числа упорядочены по возрастанию (например, в числе 4459 последние две цифры упорядочены по возрастанию).

Найдите количество таких чисел и целую часть от модуля их среднего арифметического.

Решение. «Число x однозначное или двузначное», значит $x \in [-99; 99]$ (или $x \in (-100; 100)$).

Проверка «удовлетворяет только одному из условий»: (условие 1) $\neq$ (условие 2).

Решение

```
with open('17_ev2.txt') as ff:
    a = [int(x) for x in ff.read().split()]

t = []
for x in a:
    xx = abs(x)
    if (-100 < x < 100) != (xx // 10 % 10 < xx % 10):
        t += [x]

res2 = int(abs(sum(t)/len(t))) # целая часть от модуля среднего арифметического
print(len(t), res2) # 911 3851 :)
```

Задания для самостоятельной работы

№ 17.11 (→). В файле 17_ev1.txt содержится последовательность целых чисел. Элементы последовательности могут принимать целые значения от 0 до 10 000 включительно. Рассматривается множество элементов последовательности, которые кратны трём и пяти, но не кратны семи.

Найдите количество таких чисел и их сумму.

№ 17.12 (→). В файле 17_ev1.txt содержится последовательность целых чисел. Элементы последовательности могут принимать целые значения от 0 до 10 000 включительно. Рассматривается множество элементов последовательности, которые удовлетворяют следующим условиям:

- число трёхзначное;
- сумма цифр числа кратна семи.

Найдите количество таких чисел и максимальное из них.

№ 17.13 (→). В файле 17_ev2.txt содержится последовательность целых чисел. Элементы последовательности могут принимать целые значения от $-10\,000$ до $10\,000$ включительно. Рассматривается множество элементов последовательности, которые удовлетворяют только одному из условий:

- число отрицательное;
- число положительное трёхзначное и его восьмеричная запись оканчивается нулём.

Найдите количество таких чисел и целую часть от модуля их среднего арифметического.

№ 17.14 (→). В файле 17_ev2.txt содержится последовательность целых чисел. Элементы последовательности могут принимать целые значения от $-10\,000$ до $10\,000$ включительно. Рассматривается множество элементов последовательности, которые удовлетворяют хотя бы одному из условий:

- число двузначное и кратное семи;
- число кратно 119 и 19.

Найдите количество таких чисел и последние четыре цифры их произведения.

№ 17.15 (→). В файле 17_ev2.txt содержится последовательность целых чисел. Элементы последовательности могут принимать целые значения от $-10\,000$ до $10\,000$ включительно. Рассматривается множество элементов последовательности, которые являются четырёхзначными числами, оканчивающимися на 12. Найдите количество таких чисел, а также сумму минимального и максимального из таких чисел.

Задания для дополнительной отработки

№ 17.16 (→). В файле 17_ev1.txt содержится последовательность целых чисел. Элементы последовательности могут принимать целые значения от 0 до $10\,000$ включительно. Рассматривается множество элементов последовательности, которые кратны четырём, но не кратны трём и семи. Найдите количество таких чисел и максимальное из них.

№ 17.17 (→). В файле 17_ev1.txt содержится последовательность целых чисел. Элементы последовательности могут принимать целые значения от 0 до $10\,000$ включительно. Рассматривается множество элементов последовательности, для которых запись в двоичной системе оканчивается на 11, а запись в семеричной системе счисления оканчивается на 5. Найдите количество таких чисел и минимальное из них.

№ 17.18 (→). В файле 17_ev1.txt содержится последовательность целых чисел. Элементы последовательности могут принимать целые значения от 0 до $10\,000$ включительно. Рассматривается множество элементов последовательности, которые удовлетворяют следующим условиям:

- число трёхзначное;
- сумма цифр числа делится на пять с остатком три.

Найдите количество таких чисел и минимальное из них.

№ 17.19 (→). В файле 17_ev2.txt содержится последовательность целых чисел. Элементы последовательности могут принимать целые значения от $-10\,000$ до $10\,000$ включительно. Рассматривается множество элементов последовательности, которые удовлетворяют хотя бы одному из условий:

- число отрицательное;
- число положительное двузначное и его шестнадцатеричная запись оканчивается тройкой.

Найдите количество таких чисел и целую часть от модуля их среднего арифметического.

№ 17.20 (→). В файле 17_ev3.txt содержится последовательность целых чисел. Элементы последовательности могут принимать целые значения от $-10\,000$ до $10\,000$ включительно. Рассматривается множество элементов последовательности, которые являются отрицательными четырёхзначными числами, оканчивающимися на 12, но не на 212. Найдите количество таких чисел и максимальное из них.

17.3 Примеры решения задач. Часть II (предварительный простой шаг)

Пример 17.5. В файле 17_ev4.txt содержится последовательность натуральных чисел из диапазона [10; 10 000]. Определите количество нечётных элементов последовательности, которые превышают среднее арифметическое всех элементов последовательности более чем в два раза. В ответе запишите количество найденных чисел, затем минимальное из таких чисел.

Решение-1. Алгоритмы поиска количества и минимума

```
with open('17_ev4.txt') as ff:
    a = [int(x) for x in ff.read().split()]

cnt = 0
mn = max(a) + 1
mean = sum(a) / len(a) # среднее арифметическое (посчитали однократно)

for x in a:
    if x % 2 != 0 and x > 2*mean:
        cnt += 1
        if mn > x:
            mn = x

print(cnt, mn) # 38 9535 :)
```

Решение-2. Создание списка подходящих чисел

```
with open('17_ev4.txt') as ff:
    a = [int(x) for x in ff.read().split()]

mean = sum(a) / len(a) # среднее арифметическое (посчитали однократно)
t = []
for x in a:
    if x % 2 != 0 and x > 2*mean:
        t += [x]

print(len(t), min(t)) # 38 9535 :)
```

Решение-3. Создание списка подходящих чисел. Использование генератора списка

```
with open('17_ev4.txt') as ff:
    a = [int(x) for x in ff.read().split()]

mean = sum(a) / len(a) # среднее арифметическое (посчитали однократно)
t = [x for x in a if x % 2 != 0 and x > 2*mean]

print(len(t), min(t)) # 38 9535 :)
```

Задания для самостоятельной работы

№ 17.21 (→). В файле 17_ev2.txt содержится последовательность целых чисел. Элементы последовательности могут принимать целые значения от $-10\,000$ до $10\,000$ включительно. Рассматривается множество положительных элементов последовательности, не превышающих среднее арифметическое всех элементов последовательности. Гарантируется, что в последовательности есть хотя бы одно положительное число, удовлетворяющее условию задачи. Найдите количество таких чисел и их сумму.

№ 17.22 (→). В файле 17_ev5.txt содержится последовательность натуральных чисел. Элементы последовательности не превышают $10\,000$. Рассматривается множество элементов последовательности, кратных минимальному элементу последовательности. Найдите количество таких чисел и максимальное из них.

№ 17.23 (→). В файле 17_ev2.txt содержится последовательность целых чисел. Элементы последовательности могут принимать целые значения от $-10\,000$ до $10\,000$ включительно. Рассматривается множество положительных элементов последовательности, которые не превышают целую часть выражения $\sqrt{m}$, где m — значение максимального элемента последовательности (гарантируется, что такой элемент больше нуля). Найдите количество таких чисел и их сумму.

Задания для дополнительной отработки

№ 17.24 (→). В файле 17_ev5.txt содержится последовательность натуральных чисел. Элементы последовательности не превышают $10\,000$. Рассматривается множество элементов последовательности, которые делят максимальный элемент последовательности с остатком пять. Найдите количество таких чисел и их сумму.

№ 17.25 (→). В файле 17_ev2.txt содержится последовательность целых чисел. Элементы последовательности могут принимать целые значения от $-10\,000$ до $10\,000$ включительно. Рассматривается множество элементов последовательности, кратных сумме цифр минимального элемента последовательности. Найдите количество таких чисел и максимальное из них.

№ 17.26 (→). В файле 17_ev2.txt содержится последовательность целых чисел. Элементы последовательности могут принимать целые значения от $-10\,000$ до $10\,000$ включительно. Рассматривается множество чётных элементов последовательности, превышающих среднее арифметическое всех элементов последовательности. Найдите количество таких чисел и последние четыре цифры их суммы.

17.4 Примеры решения задач. Часть III («двухэтапные» решения)

Пример 17.6. В файле 17_ev2.txt содержится последовательность целых чисел. Элементы последовательности могут принимать целые значения от $-10\,000$ до $10\,000$ включительно. Рассматривается множество элементов последовательности, которые удовлетворяют условиям:

- число трёхзначное;
- число кратно минимальному двузначному числу последовательности, кратному семи.

Найдите количество таких чисел и максимальное из них.

Решение-1. Алгоритмы (рекомендуется знать)

```
with open('17_ev2.txt') as ff:
    a = [int(x) for x in ff.read().split()]

# I этап (вспомогательный): ищем мин. двузначное число, кратное семи
mn7 = max(a) + 1
for x in a:
    if 10 <= abs(x) <= 99 and x % 7 == 0:
        if x < mn7:
            mn7 = x
#print(mn7) # 14, печать для контроля

# II этап: решение основной задачи
mx = -10_000 - 1 # [-10_000; 10_000]
cnt = 0
for x in a:
    if 99 < abs(x) < 1000 and x % mn7 == 0:
        cnt += 1
        if x > mx: mx = x

print(cnt, mx) # 19 910 :)
```

Решение-2. (рекомендуется для экзамена)

```
with open('17_ev2.txt') as ff:
    a = [int(x) for x in ff.read().split()]

# I этап (вспомогательный): создаём список двузначных чисел, кратных семи
a7 = [x for x in a if 10 <= abs(x) <= 99 and x % 7 == 0]
mina7 = min(a7) # определяем минимальное из отобранных чисел
#print(mn7) # 14, печать для контроля

# II этап: решение основной задачи
t = []
for x in a:
    if 99 < abs(x) < 1000 and x % mina7 == 0:
        t += [x]
print(len(t), max(t)) # 19 910 :)
```

Решение-3. Только генераторы (не для экзамена, для опытных)

```
with open('17_ev2.txt') as ff:
    a = [int(x) for x in ff.read().split()]

mina7 = min(x for x in a if 10 <= abs(x) <= 99 and x % 7 == 0)
#print(mina7) # 14

t = [x for x in a if 99 < abs(x) < 1000 and x % mina7 == 0]
print(len(t), max(t)) # 19 910 :)
```

Задания для самостоятельной работы

№ 17.27 (→). В файле 17_ev2.txt содержится последовательность целых чисел. Элементы последовательности могут принимать целые значения от $-10\,000$ до $10\,000$ включительно. Рассматривается множество элементов последовательности, кратных максимальному нечётному двузначному числу последовательности. Найдите количество таких чисел и минимальное из них.

№ 17.28 (→). В файле 17_ev1.txt содержится последовательность целых чисел. Элементы последовательности могут принимать целые значения от 0 до $10\,000$ включительно. Рассматривается множество элементов последовательности, которые удовлетворяют следующим условиям:

- запись в двоичной системе оканчивается на 11;
- число не превышает минимальное число последовательности, которое оканчивается на 12.

Найдите количество таких чисел и произведение их минимального и максимального значений.

№ 17.29 (→). В файле 17_ev2.txt содержится последовательность целых чисел. Элементы последовательности могут принимать целые значения от $-10\,000$ до $10\,000$ включительно. Рассматривается множество положительных элементов последовательности, не превышающих среднее арифметическое элементов последовательности, кратных трём. Найдите количество таких чисел и максимальное из них.

Задания для дополнительной отработки

№ 17.30 (→). В файле 17_ev2.txt содержится последовательность целых чисел. Элементы последовательности могут принимать целые значения от $-10\,000$ до $10\,000$ включительно. Рассматривается множество элементов последовательности, кратных минимальному нечётному трёхзначному числу последовательности. Найдите количество таких чисел и максимальное из них.

№ 17.31 (→). В файле 17_ev1.txt содержится последовательность целых чисел. Элементы последовательности могут принимать целые значения от 0 до $10\,000$ включительно. Рассматривается множество элементов последовательности, которые удовлетворяют следующим условиям:

- запись в восьмеричной системе оканчивается на 00;
- число превышает минимальное число последовательности, которое в двоичной системе счисления оканчивается на 101.

Найдите количество таких чисел и сумму их минимального и максимального значений.

№ 17.32 (→). В файле 17_ev2.txt содержится последовательность целых чисел. Элементы последовательности могут принимать целые значения от $-10\,000$ до $10\,000$ включительно. Рассматривается множество элементов последовательности, кратных минимальному положительному элементу последовательности, кратному двенадцати. Найдите количество таких чисел и максимальное из них.

№ 17.33 (→). В файле 17_ev6.txt содержится последовательность натуральных чисел. Её элементы могут принимать целые значения от 1 до $100\,000$ включительно. Определите количество элементов последовательности, которые являются четырёхзначными числами, сумма цифр которых кратна минимальному чётному элементу последовательности. В ответе запишите количество найденных чисел, затем максимальную из сумм цифр таких чисел.

17.5 Примеры решения задач. Часть IV (пары чисел, простой предварительный шаг)

Пример 17.7. В файле 17_ev1.txt содержится последовательность целых чисел. Элементы последовательности могут принимать целые значения от 0 до 10 000. Определите количество пар элементов последовательности, в которых хотя бы одно число больше среднего арифметического всех чисел последовательности. В ответе запишите количество найденных пар, затем максимальную из сумм элементов таких пар. В данной задаче под парой подразумевается два идущих подряд элемента последовательности.

Решение-1. Алгоритмы (рекомендуется знать)

```
with open('17_ev1.txt') as ff:
    a = [int(x) for x in ff.read().split()]

# I этап (вспомогательный): ищем среднее арифметическое
sm, cnt = 0, 0
for x in a:
    cnt += 1
    sm += x
mean = sm / cnt
print(mean) # 4767.125 для контроля

# II этап: решение основной задачи
mx = -1 # [0; 10_000] выходим за левую границу отрезка
cnt = 0
for i in range(len(a)-1): # цикл по индексам
    if a[i] > mean or a[i+1] > mean:
        cnt += 1
        if a[i] + a[i+1] > mx:
            mx = a[i] + a[i+1]
print(cnt, mx) # 1382 19333 :)
```

Решение-2 (рекомендуется для экзамена)

```
with open('17_ev1.txt') as ff:
    a = [int(x) for x in ff.read().split()]

# I этап (вспомогательный): ищем среднее арифметическое
mean = sum(a) / len(a)

# II этап: решение основной задачи
t = []
for i in range(len(a)-1):
    if a[i] > mean or a[i+1] > mean:
        t += [ a[i] + a[i+1] ] # ! Сохраняем сумму элементов пары

print(len(t), max(t)) # 1382 19333 :)
```

Задания для самостоятельной работы

№ 17.34 (→). В файле 17_ev1.txt содержится последовательность целых чисел. Элементы последовательности могут принимать целые значения от 0 до 10 000. Определите количество пар элементов последовательности, в которых оба числа больше среднего арифметического всех чисел последовательности. В ответе запишите количество найденных пар, затем минимальную из сумм элементов таких пар. В данной задаче под парой подразумевается два идущих подряд элемента последовательности.

№ 17.35 (→). В файле 17_ev1.txt содержится последовательность целых чисел. Элементы последовательности могут принимать целые значения от 0 до 10 000 включительно. Определите количество пар, в которых хотя бы один из двух элементов больше, чем среднее арифметическое всех чисел в файле, а их сумма делится на 13. В ответе запишите два числа: сначала количество найденных пар, а затем минимальную из сумм элементов таких пар. В данной задаче под парой подразумевается два идущих подряд элемента последовательности.

№ 17.36 (→). В файле 17_ev2.txt содержится последовательность целых чисел. Элементы последовательности могут принимать целые значения от $-10\,000$ до $10\,000$ включительно. Определите количество пар, в которых хотя бы один из двух элементов больше, чем среднее арифметическое всех чисел в файле, а их сумма оканчивается на 13. В ответе запишите два числа: сначала количество найденных пар, а затем максимальную из сумм элементов таких пар. В данной задаче под парой подразумевается два идущих подряд элемента последовательности.

№ 17.37 (→). В файле 17_ev2.txt содержится последовательность целых чисел. Элементы последовательности могут принимать целые значения от $-10\,000$ до $10\,000$ включительно. Определите количество пар, в которых только один из двух элементов больше, чем среднее арифметическое всех чисел в файле, а их сумма оканчивается на 7. В ответе запишите два числа: сначала количество найденных пар, а затем минимальную из сумм элементов таких пар. В данной задаче под парой подразумевается два идущих подряд элемента последовательности.

№ 17.38 (→). В файле 17_ev6.txt содержится последовательность натуральных чисел, не превышающих 100 000. Определите количество пар, в которых остаток от деления на три хотя бы одного числа из пары равен остатку от деления на три минимального элемента всей последовательности. В ответе запишите два числа: сначала количество найденных пар, затем максимальную из сумм элементов таких пар. В данной задаче под парой подразумевается два идущих подряд элемента последовательности.

№ 17.39 (→). В файле 17_ev6.txt содержится последовательность натуральных чисел, не превышающих 100 000. Определите количество троек, для которых остаток от деления на 7 только одного числа из тройки равен остатку от деления на 11 максимального элемента всей последовательности. В ответе запишите два числа: сначала количество найденных троек, затем минимальную из сумм элементов таких троек. В данной задаче под тройкой подразумевается три идущих подряд элемента последовательности.

Задания для дополнительной отработки

№ 17.40 (→). В файле 17_ev1.txt содержится последовательность целых чисел. Элементы последовательности могут принимать целые значения от 0 до 10 000. Определите количество пар элементов последовательности, в которых оба числа больше среднего арифметического всех чисел последовательности. В ответе запишите количество найденных пар, затем сумму из минимальных элементов таких пар. В данной задаче под парой подразумевается два идущих подряд элемента последовательности.

№ 17.41 (→). В файле 17_ev1.txt содержится последовательность целых чисел. Элементы последовательности могут принимать целые значения от 0 до 10 000. Определите количество троек элементов последовательности, в которых все три числа больше среднего арифметического всех чисел последовательности. В ответе запишите количество найденных троек, затем минимальную из сумм элементов таких троек. В данной задаче под тройкой подразумевается три идущих подряд элемента последовательности.

№ 17.42 (→). В файле 17_ev6.txt содержится последовательность натуральных чисел, не превышающих 100 000. Определите количество пар, в которых только один из двух элементов больше, чем среднее арифметическое всех чисел в файле, а их сумма оканчивается на 19. В ответе запишите два числа: сначала количество найденных пар, а затем максимальную из сумм элементов таких пар. В данной задаче под парой подразумевается два идущих подряд элемента последовательности.

№ 17.43 (→). В файле 17_ev6.txt содержится последовательность натуральных чисел, не превышающих 100 000. Определите количество пар, в которых оба элемента оканчиваются на 3, а их сумма не превышает среднего арифметического всех чисел в файле. В ответе запишите два числа: сначала количество найденных пар, а затем минимальную из сумм элементов таких пар. В данной задаче под парой подразумевается два идущих подряд элемента последовательности.

№ 17.44 (→). В файле 17_ev6.txt содержится последовательность натуральных чисел, не превышающих 100 000. Определите количество троек, для которых остатки от деления на 9 ровно двух чисел из тройки равны остатку от деления на 11 максимального элемента всей последовательности. В ответе запишите два числа: сначала количество найденных троек, затем сумму минимальных элементов таких троек. В данной задаче под тройкой подразумевается три идущих подряд элемента последовательности.

№ 17.45 (→). В файле 17_ev2.txt содержится последовательность целых чисел, не превышающих по модулю 10 000. Определите количество троек элементов последовательности, для которых выполняются оба условия:

- среди чисел тройки есть хотя бы одно четырёхзначное число;
- в тройке чисел не более одного числа, которое заканчивается на ту же цифру, что и максимальный элемент всей последовательности.

В ответе запишите два числа: сначала количество найденных троек, затем минимальную из сумм элементов этих троек. В данной задаче под тройкой подразумевается три идущих подряд элемента последовательности.

17.6 Примеры решения задач. Часть V (пары чисел, «двухэтапное» решение)

Пример 17.8. В файле 17_ev2.txt содержится последовательность целых чисел. Элементы последовательности могут принимать целые значения от $-10\,000$ до $10\,000$. Определите количество пар элементов последовательности, в которых хотя бы одно число кратно минимальному положительному числу в последовательности, кратному 11. Гарантируется, что в последовательности есть хотя бы одно положительное число, кратное 11. В ответе запишите количество найденных пар, затем максимальную из сумм элементов таких пар. В данной задаче под парой подразумевается два идущих подряд элемента последовательности.

Решение-1. Алгоритмы (рекомендуется знать)

```
with open('17_ev2.txt') as ff:
    a = [int(x) for x in ff.read().split()]

# I этап (вспомогательный): ищем мин. число, кратное 11
mn11 = max(a) + 1
for x in a:
    if x > 0 and x % 11 == 0 and x < mn11:
        mn11 = x
print(mn11) # 33 для контроля

# II этап: решение основной задачи
mx = 2*min(a) - 1 #
cnt = 0
for i in range(len(a)-1): # цикл по индексам
    if a[i] % mn11 == 0 or a[i+1] % mn11 == 0:
        cnt += 1
        if a[i] + a[i+1] > mx:
            mx = a[i] + a[i+1]
print(cnt, mx) # 134 16602 :)
```

Решение-2. (рекомендуется для экзамена)

```
with open('17_ev2.txt') as ff:
    a = [int(x) for x in ff.read().split()]

# I этап (вспомогательный): ищем мин. число, кратное 11
a11 = [x for x in a if x > 0 and x % 11 == 0]
mn11 = min(a11) # напечатать для контроля за правильностью вычислений!

# II этап: решение основной задачи
t = []
for i in range(len(a)-1):
    if a[i] % mn11 == 0 or a[i+1] % mn11 == 0:
        t += [ a[i] + a[i+1] ] # ! Сохраняем сумму элементов пары

print(len(t), max(t)) # 134 16602 :)
```

Пример 17.9. В файле 17_ev2.txt содержится последовательность целых чисел. Элементы последовательности могут принимать целые значения от $-10\,000$ до $10\,000$. Определите количество пар элементов последовательности, в которых хотя бы одно число кратно минимальному положительному числу последовательности, кратному 19. Гарантируется, что в последовательности есть хотя бы одно положительное число, кратное 19. В ответе запишите количество найденных пар, затем сумму всех минимальных элементов таких пар. В данной задаче под парой подразумевается два идущих подряд элемента последовательности.

Решение-1. Алгоритмы (рекомендуется знать)

```
with open('17_ev2.txt') as ff:
    a = [int(x) for x in ff.read().split()]

# I этап (вспомогательный): ищем мин. число, кратное 19
mn19 = max(a) + 1
for x in a:
    if x > 0 and x % 19 == 0 and x < mn19:
        mn19 = x
print(mn19) # 133 для контроля

# II этап: решение основной задачи
sm = cnt = 0
for i in range(len(a)-1):
    if a[i] % mn19 == 0 or a[i+1] % mn19 == 0:
        cnt += 1
        if a[i] < a[i+1]:
            sm += a[i]
        else:
            sm += a[i+1]

print(cnt, sm) # 36 65023 :)
```

Решение-2. (рекомендуется для экзамена)

```
with open('17_ev2.txt') as ff:
    a = [int(x) for x in ff.read().split()]

# I этап (вспомогательный): ищем мин. число, кратное 19
a19 = [x for x in a if x > 0 and x % 19 == 0]
mn19 = min(a19) # напечатать для контроля за правильностью вычислений!

# II этап: решение основной задачи
t = []
for i in range(len(a)-1):
    if a[i] % mn19 == 0 or a[i+1] % mn19 == 0:
        t += [ min(a[i], a[i+1]) ] # ! Сохраняем мин. из пары

print(len(t), sum(t)) # 36 65023 :)
```

Задания для самостоятельной работы

№ 17.46 (→). В файле 17_ev1.txt содержится последовательность целых чисел. Элементы последовательности могут принимать целые значения от 0 до 10 000. Определите количество пар элементов последовательности, в которых оба числа больше среднего арифметического всех чисел последовательности, оканчивающихся на 13. Гарантируется, что в последовательности есть хотя бы одно число, оканчивающееся на 13. В ответе запишите количество найденных пар, затем минимальную из сумм элементов таких пар. В данной задаче под парой подразумевается два идущих подряд элемента последовательности.

№ 17.47 (→). В файле 17_ev1.txt содержится последовательность целых чисел. Элементы последовательности могут принимать целые значения от 0 до 10 000 включительно. Определите количество пар, в которых хотя бы один из двух элементов больше, чем среднее арифметическое всех чисел в файле, а их сумма делится нацело на минимальный элемент последовательности, кратный трём. Гарантируется, что в последовательности есть хотя бы одно число, кратное трём. В ответе запишите два числа: сначала количество найденных пар, а затем минимальную из сумм элементов таких пар. В данной задаче под парой подразумевается два идущих подряд элемента последовательности.

№ 17.48 (→). В файле 17_ev2.txt содержится последовательность целых чисел. Элементы последовательности могут принимать целые значения от $-10\,000$ до $10\,000$ включительно. Определите количество пар, в которых оба числа имеют один знак, а их сумма меньше минимального элемента последовательности, который оканчивается на 15. Гарантируется, что в последовательности есть хотя бы одно число, оканчивающееся на 15. В ответе запишите два числа: сначала количество найденных пар, а затем максимальную из сумм элементов таких пар. В данной задаче под парой подразумевается два идущих подряд элемента последовательности.

№ 17.49 (→). В файле 17_ev2.txt содержится последовательность целых чисел. Элементы последовательности могут принимать целые значения от $-10\,000$ до $10\,000$ включительно. Определите количество троек, в которых только один из трёх элементов положительный, а произведение чисел тройки оканчивается на ту же цифру, что и минимальный трёхзначный положительный элемент последовательности. В ответе запишите два числа: сначала количество найденных троек, а затем максимальную из сумм элементов таких троек. В данной задаче под тройкой подразумевается три идущих подряд элемента последовательности.

№ 17.50 (→). В файле 17_2024.txt содержится последовательность натуральных чисел, каждое из которых не превышает 100 000. Определите количество троек элементов последовательности, в которых ровно два из трёх элементов являются трёхзначными числами, а сумма минимального и максимального элементов тройки больше максимального элемента последовательности, оканчивающегося на 17. Гарантируется, что в последовательности есть хотя бы одно число, оканчивающееся на 17. В ответе запишите количество найденных троек чисел, затем максимальную из сумм элементов таких троек. В данной задаче под тройкой подразумевается три идущих подряд элемента последовательности.

№ 17.51 (→). В файле 17_2026.txt содержится последовательность натуральных чисел. Её элементы могут принимать целые значения от 1 до 100 000 включительно. Определите количество пар последовательности, в которых только один из элементов является двузначным числом, а произведение элементов пары кратно минимальному двузначному элементу последовательности. В ответе запишите количество найденных пар, затем максимальную из сумм элементов таких пар. В данной задаче под парой подразумевается два идущих подряд элемента последовательности.

№ 17.52 (→). В файле 17_2025.txt содержится последовательность натуральных чисел. Её элементы могут принимать целые значения от 1 до 100 000 включительно. Определите количество пар последовательности, в которых остаток от деления хотя бы одного из элементов на 16 равен последним двум цифрам минимального элемента последовательности, оканчивающегося на 4. Гарантируется, что в последовательности есть хотя бы одно число, оканчивающееся на 4. В ответе запишите количество найденных пар, затем максимальную из сумм элементов таких пар. В данной задаче под парой подразумевается два идущих подряд элемента последовательности.

№ 17.53 (→). В файле 17_2023.txt содержится последовательность целых чисел. Элементы последовательности могут принимать целые значения от $-10\,000$ до $10\,000$ включительно. Определите количество пар последовательности, в которых только одно число оканчивается на 7, а сумма квадратов элементов пары не меньше квадрата максимального элемента последовательности, оканчивающегося на 6. В ответе запишите два числа: сначала количество найденных пар, затем максимальную из сумм квадратов элементов таких пар. В данной задаче под парой подразумевается два идущих подряд элемента последовательности.

Ответы

№ 17.11: 136 637965

№ 17.12: 30 984

№ 17.13: 198 4495

Если ответ не совпадает, перечитайте условие.

№ 17.14: 7 5600

№ 17.15: 18 1400

№ 17.16: 291 9776

№ 17.17: 64 159

№ 17.18: 49 152

№ 17.19: 179 5031

№ 17.20: 6 -1012

№ 17.21: 839 1619140

№ 17.22: 198 9660

№ 17.23: 19 668

№ 17.24: 2 3669

№ 17.25: 81 9660

№ 17.26: 499 4012

№ 17.27: 45 —9620

№ 17.28: 21 933

№ 17.29: 926 4260

№ 17.30: 5 8631

№ 17.31: 30 9984

№ 17.32: 50 9168

№ 17.33: 271 30

№ 17.34: 394 9592

№ 17.35: 87 5694

№ 17.36: 20 18513

№ 17.37: 111 —3217

№ 17.38: 1047 173050

№ 17.39: 601 1265

№ 17.40: 394 2766572

№ 17.41: 172 18364

№ 17.42: 18 69119

№ 17.43: 7 586

№ 17.44: 90 212391

№ 17.45: 1964 —20419

№ 17.46: 499 8906

№ 17.47: 460 4863

№ 17.48: 31 -9643

№ 17.49: 4 315

№ 17.50: 4 101024

№ 17.51: 470 9949

№ 17.52: 1216 177727

№ 17.53: 221 188785853

Приложение 1

Если эта информация для вас актуальна, значит вы ещё не готовы к экзамену. 🇨🇭)

Основные понятия 📖

Список (list) в Python — это упорядоченная коллекция элементов. Список позволяет хранить несколько значений в одной переменной.

Длина списка, т.е. количество элементов списка a находится с помощью функции `len(a)`.

Элементы в списке имеют номера. Нумерация может вестись слева направо и справа налево. Пусть n — это количество элементов в списке a , т.е. $n = \text{len}(a)$.

Слева направо

Первый элемент имеет **слева** номер 0, последний — номер $n - 1$

Справа налево

Первый элемент имеет **справа** номер -1 , последний — номер $-n$

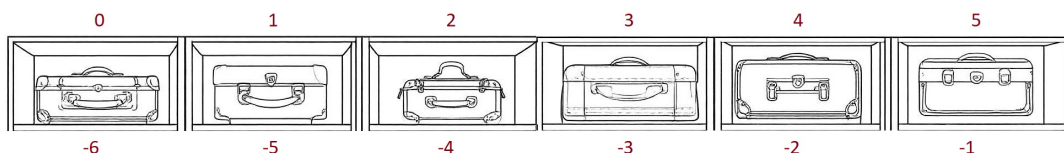


Рис. 1. Шесть ячеек с чемоданами. Список с шестью элементами

Кортеж (tuple) — неизменяемый список.

Работа со списком целиком

[Примеры работы со списком целиком](#)

```
import math # нужно только для prod

a = [4, -9, 50, 19, 21, 4, -7] # инициализация списка
b = [] # инициализация пустого списка

# вывод списка на экран
print(a) # [4, -9, 50, 19, 21, 4, -7]
print(b) # []

# вывод списка на экран с "разыменованием"
print(*a) # 4 -9 50 19 21 4 -7

# вывод длины (количество элементов) списка
print(len(a)) # 7
print(len(b)) # 0

# вывод количества вхождений числа 4 в список
print(a.count(4)) # 2

# вывод минимального числа в списке
print(min(a)) # -9

# вывод максимального числа в списке
print(max(a)) # 50

# сумма элементов в списке
print(sum(a)) # 82

# среднее арифметическое всех элементов списка
print(sum(a)/len(a)) # 11.714285714285714

# целая часть среднего арифметического всех элементов списка
print(int(sum(a)/len(a))) # 11

# произведение элементов в списке (требуется импорт из math)
print(math.prod(a)) # 20109600
```

Примеры работы со списком целиком (модификация списка)

```
a = [4, -9, 50, 19, 21, 4, -7] # инициализация списка

# переворот (инвертирование) списка
a.reverse() # или a = a[::-1]
print(a) # [-7, 4, 21, 19, 50, -9, 4]

# сортировка списка по неубыванию
a.sort() # изменяет список a
# или a = sorted(a) # действие sorted(a) не изменяет исходного списка
print(a) # [-9, -7, 4, 4, 19, 21, 50]

# сортировка списка по невозрастанию
a.sort(reverse=True)
# или a = sorted(a, reverse=True)
print(a) # [50, 21, 19, 4, 4, -7, -9]

# добавление одного элемента в КОНЕЦ списка
a += [100] # или a.append(100)
print(a) # [50, 21, 19, 4, 4, -7, -9, 100]

# добавление нескольких элементов в КОНЕЦ списка
a += [-3, -4] # или a.extend([-3, -4])
print(a) # [50, 21, 19, 4, 4, -7, -9, 100, -3, -4]

# добавление одного элемента в НАЧАЛО списка
a = [-100] + a
print(a) # [-100, 50, 21, 19, 4, 4, -7, -9, 100, -3, -4]

# добавление нескольких элементов в НАЧАЛО списка
a = [22, 55] + a
print(a) # [22, 55, -100, 50, 21, 19, 4, 4, -7, -9, 100, -3, -4]
```

Работа с фрагментами списка (срезы)

Примеры обращения к элементам списка

```
#   0   1   2   3   4   5   6   неотрицательная нумерация элементов
a = [2, 34, 6, -7, 12, -10, 13]
#  -7  -6  -5  -4  -3  -2  -1  отрицательная нумерация элементов

# первый (слева) элемент списка
print(a[0])    # 2

# последний элемент списка
print(a[-1])   # 13

# срез с третьего элемента и до конца
print(a[2:])   # [6, -7, 12, -10, 13]

# срез с первыми тремя элементами
print(a[:3])   # [2, 34, 6]

# срез всех элементов кроме первого и последнего
print(a[1:-1]) # [34, 6, -7, 12, -10]

# взять каждый второй элемент, начиная с нулевого
print(a[::2])  # [2, 6, 12, 13]

# взять каждый второй второй, начиная с первого
print(a[1::2]) # [34, -7, -10]

# инвертирование списка (переворот)
print(a[::-1]) # [13, -10, 12, -7, 6, 34, 2]
```

Перебор элементов списка </>

Примеры печати элементов списка

```
a = [2, 34, 6, -7, 12, -10, 13]

# печать списка по элементам (в одну строку через пробел)
for x in a:
    print(x, end=' ') # 2 34 6 -7 12 -10 13

print() # пустая строка
# печать чётных элементов списка
for x in a:
    if x % 2 == 0:
        print(x, end=' ') # 2 34 6 12 -10

print()
# печать двузначных элементов списка
for x in a:
    if 10 <= abs(x) <= 99:
        print(x, end=' ') # 34 12 -10 13

print()
# печать двузначных элементов списка, меньших его последнего элемента
for x in a:
    if 10 <= abs(x) <= 99 and x < a[-1]:
        print(x, end=' ') # 12 -10
```

Печать пар элементов списка

```
a = [2, 34, 6, -7, 12, -10, 13]

# печать пар соседних элементов списка
for i in range(len(a)-1): # цикл по индексам
    print(a[i], a[i+1])
```

Результат

```
12 -10
2 34
34 6
6 -7
-7 12
12 -10
-10 13
```

Печать троек элементов списка

```
a = [2, 34, 6, -7, 12, -10, 13]

# печать троек соседних элементов списка
for i in range(len(a)-2): # цикл по индексам
    print(a[i], a[i+1], a[i+2])
```

Результат

```
2 34 6
34 6 -7
6 -7 12
-7 12 -10
12 -10 13
```

```
a = [2, 34, 6, -7, 12, -10, 13]

# Поиск количества чётных элементов списка
cnt = 0
for x in a:
    if x % 2 == 0:
        cnt += 1
print(cnt) # 5

# Поиск суммы всех элементов списка
sm = 0
for x in a:
    sm += x
print(sm) # 50

# Поиск произведения положительных элементов списка
pr = 1
for x in a:
    if x > 0:
        pr *= x
print(pr) # 63648

# Поиск среднего арифметического двузначных элементов списка
sm = 0
cnt = 0
for x in a:
    if 10 <= abs(x) <= 99:
        sm += x
        cnt += 1
average = sm / cnt
print(average) # 12.25

# Формирование нового списка с индексами отрицательных элементов
t = []
for i in range(len(a)): # цикл по индексам
    if a[i] < 0:
        t += [i]
print(t) # [3, 5]
```

Приложение 2

Если эта информация для вас актуальна, значит вы ещё не готовы к экзамену. 🇨🇭

Целочисленные операции

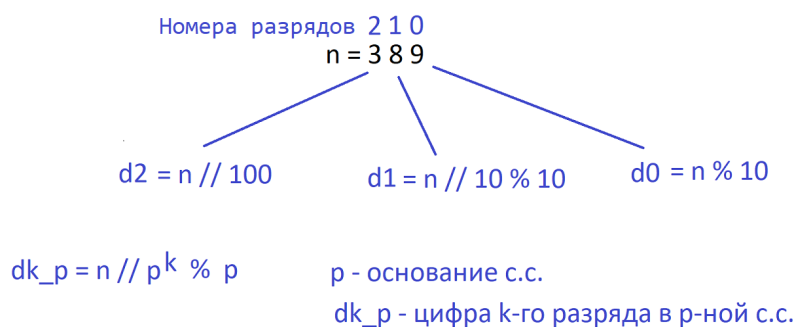
Операция	Описание	Пример	Результат
//	целочисленное деление	17 // 2	8
%	остаток от деления нацело	17 % 2	1
Функция	Описание	Пример	Результат
divmod()	результат — кортеж вида (деление нацело, остаток от деления нацело)	divmod(17, 2)	(8, 1)

Схема для запоминания действия операций «//» и «%»

$$\begin{array}{r|l} 17 & 2 \\ \hline 16 & 8 \leftarrow 17 // 2 \\ \hline 1 & \leftarrow 17 \% 2 \end{array}$$

Операции «//» и «%» используются, в частности, для отделения частей целого числа (цифр, пар цифр и т.п.).

Цифры натурального числа



Цифры 10-ной записи натурального числа

```
n = 389 # дано трёхзначное число

# цифры целого числа в 10-ной с.с. (p = 10)
d0 = n % 10      # цифра разряда единиц
d1 = n // 10 % 10 # цифра разряда десятков
d2 = n // 100    # цифра разряда сотен

print(d2, d1, d0, n) # 3 8 9 389
```

Цифры целого отрицательного числа

Цифры 10-ной записи целого отрицательного(!) числа

```
n = -389 # дано трёхзначное ОТРИЦАТЕЛЬНОЕ число

d0 = abs(n) % 10      # цифра разряда единиц
d1 = abs(n) // 10 % 10 # цифра разряда десятков
d2 = abs(n) // 100     # цифра разряда сотен

print(d2, d1, d0, n) # 3 8 9 389
```

Части натурального числа

Цифры пятизначного натурального числа

```
# 43210 <- номера разрядов
n = 38912 # дано пятизначное число

# цифры целого числа в 10-ной с.с. (p = 10)
d0 = n % 10      # цифра разряда единиц
d1 = n // 10 % 10 # цифра разряда десятков
d2 = n // 100 % 10 # цифра разряда сотен
d3 = n // 1000 % 10 # цифра разряда тысяч
d4 = n // 10000    # цифра разряда десятков тысяч

print(d4, d3, d2, d1, d0, n) # 3 8 9 1 2 38912

# -----
# фрагменты целого числа в 10-ной с.с. (p = 10)
d10 = n % 100      # двузначное число в конце числа n (номера разрядов цифр 1 и 0)
d210 = n % 1000    # трёхзначное число в конце числа n (номера разрядов 2, 1 и 0)
d3210 = n % 10000  # четырёхзначное число в конце числа n

print(d10, d210, d3210) # 12 912 8912
```

Цифры натурального числа, записанного в p -ной системе счисления

Цифры 8-ной записи натурального числа

```
n = 789 # дано 10-ное трехзначное число

# В 8-ной записи числа 789 будет 4 цифры, т.к.  $8^{**3} \leq 789 < 8^{**4}$ 
d0 = n % 8
d1 = n // 8 % 8
d2 = n // 8**2 % 8
d3 = n // 8**3 # или d3 = n // 8**3 % 8

print(d3, d2, d1, d0, oct(n)) # 1 4 2 5 0o1425
```

Цифры 16-ной записи натурального числа

```
n = 389 # дано 10-ное трехзначное число

# В 16-ной записи числа 389 будет 3 цифры, т.к.  $16^{**2} \leq 389 < 16^{**3}$ 
d0 = n % 16
d1 = n // 16 % 16
d2 = n // 16**2 # или d2 = n // 16**2 % 16

print(d2, d1, d0, hex(n)) # 1 8 5 0x185

# =====
n = 2989 # дано 10-ное четырёхзначное число

# В 16-ной записи числа 2989 будет 3 цифры, т.к.  $16^{**2} \leq 2989 < 16^{**3}$ 
d0 = n % 16
d1 = n // 16 % 16
d2 = n // 16**2 # или d2 = n // 16**2 % 16

# вместо 16-ричных цифр выводятся его десятичные значения
print(d2, d1, d0, hex(n)) # 11 10 13 0xbad

# вывод 16-ричных цифр
print(f'{d2:x} {d1:x} {d0:x}', hex(n)) # b a d 0xbad
```