

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ  
ФЕДЕРАЦИИ**

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ  
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ**

**ВЫСШЕГО ОБРАЗОВАНИЯ**

**«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ» ИНСТИТУТ  
ЦИФРОВОГО РАЗВИТИЯ**

**Отчет о лабораторной работе №2.4 по дисциплине основы программной  
инженерии**

Выполнила:  
Лобанова Елизавета  
Евгеньевна,  
2 курс, группа ПИЖ-б-о-20-1

Проверил:  
Доцент кафедры прикладной  
математики и  
компьютерной безопасности,  
Воронкин Р.А.

Отчет защищен с оценкой \_\_\_\_\_ Дата защиты \_\_\_\_\_

Ставрополь, 2021 г.

## Ход работы

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
import sys

if __name__ == '__main__':
    # Ввести список одной строкой.
    A = list(map(int, input().split()))
    # Проверить количество элементов списка.
    if len(A) != 10:
        print("Неверный размер списка", file=sys.stderr)
        exit(1)
    # Найти искомую сумму.
    s = sum([a for a in A if abs(a) < 5])
    print(s)
```

Пример 1

```
1 2 1 3 4 23 45 12 36 78
11

Process finished with exit code 0
```

Результат 1 программы (1)

```
5 4 2 8 1
Неверный размер списка

Process finished with exit code 1
```

Результат 1 программы (2)

```

#!/usr/bin/env python3
# -*- coding: utf-8 -*-
import sys

if __name__ == '__main__':
    a = list(map(int, input().split()))
    if not a:
        print("Заданный список пуст", file=sys.stderr)
        exit(1)
    a_min = a_max = a[0]
    i_min = i_max = 0
    for i, item in enumerate(a):
        if item < a_min:
            i_min, a_min = i, item
        if item >= a_max:
            i_max, a_max = i, item
    if i_min > i_max:
        i_min, i_max = i_max, i_min

    count = 0
    for item in a[i_min+1:i_max]:
        if item > 0:
            count += 1
    print(count)

```

Пример 2

```
32 456 56 12 0 -9 2 -98 91
```

```
3
```

```
Process finished with exit code 0
```

Результат 2 программы (1)

Заданный список пуст

Process finished with exit code 1

Результат 2 программы (2)

Вариант 12

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-

import sys

if __name__ == '__main__':
    s = list(map(int, input().split()))
    count = 0
    if len(s) != 10:
        print("Неверный размер списка", file=sys.stderr)
        exit(1)
    m = sum([a for a in s if (2 < a < 20) and (a % 8 == 0)])
    for a in s:
        if (2 < a < 20) and (a % 8 == 0):
            count += 1
    print(f"Сумма элементов больших 2, меньших 20 и кратных 8 = {m}"
          f"\nКоличество элементов больших 2, меньших 20 и кратных 8 = {count}")
```

Индивидуальная программа 1

56 74 23 56 0

Неверный размер списка

Process finished with exit code 1

Результат работы индивидуальной программы 1 (1)

8 76 43 24 8 16 4 -9 -8 16

Сумма элементов больших 2, меньших 20 и кратных 8 = 48

Количество элементов больших 2, меньших 20 и кратных 8 = 4

Process finished with exit code 0

Результат работы индивидуальной программы 1 (2)

```

#!/usr/bin/env python3
# -*- coding: utf-8 -*-
import sys
if __name__ == '__main__':
    s = list(map(float, input().split()))
    A = float(input("Введите A, первый элемент задаваемого диапазона:"))
    B = float(input("Введите B, последний элемент задаваемого диапазона:"))
    tr = ([el for el in s if A <= el <= B or B <= el <= A])
    print(f"Количество элементов списка, лежащих в диапазоне от A до B = {len(tr)}")
    summ = 0
    s_max = s[0]
    i_max = 0
    for i, item in enumerate(s):
        if item >= s_max:
            i_max, s_max = i, item
    print(f"Максимальный элемент = {s_max}")
    summ = sum([elem for ind, elem in enumerate(s) if ind > i_max])
    print(f"Сумма элементов, расположенные после максимального элемента = {summ}")
    s2 = [abs(elem) for elem in s]
    s2.sort(reverse=True)
    print(s2)

```

Индивидуальная программа 2

```

1.5 3 4 5 7 24 8 -10
Введите A, первый элемент задаваемого диапазона: 3
Введите B, последний элемент задаваемого диапазона: 5
Количество элементов списка, лежащих в диапазоне от A до B = 3
Максимальный элемент = 24.0
Сумма элементов, расположенные после максимального элемента = -2.0
[24.0, 10.0, 8.0, 7.0, 5.0, 4.0, 3.0, 1.5]

Process finished with exit code 0

```

Результат работы индивидуальной программы 2 (1)

Ответы на вопросы:

1. Список (list) – это структура данных для хранения объектов различных типов.
2. Для создания списка нужно заключить элементы в квадратные скобки.
3. При его создании в памяти резервируется область, которую можно условно назвать некоторым “контейнером”, в котором хранятся ссылки на другие элементы данных в памяти. В отличие от таких типов данных как число или строка, содержимое “контейнера” списка можно менять.

4. Читать элементы списка можно с помощью следующего цикла: `my_list = ['один', 'два', 'три', 'четыре', 'пять']` `for elem in my_list: print(elem)`
5. Списки можно объединять с помощью операции сложения, а также повторять с помощью умножения.
6. Нужно использовать оператор «in»
7. Метод `count` можно использовать для определения числа сколько раз данный элемент встречается в списке.
8. Метод `insert` можно использовать, чтобы вставить элемент в список: `my_list = [1, 2, 3, 4, 5]` `my_list.insert(1, 'Привет')` `print(my_list)` Метод `append` можно использовать для добавления элемента в список: `my_list = ['один', 'два', 'три', 'четыре', 'пять']` `my_list.append('ещё один')` `print(my_list)`
9. Для сортировки списка нужно использовать метод `sort`. Пример: `list_2.sort()`.
10. Удалить элемент можно, написав его индекс в методе `pop`: `removed = my_list.pop(2)` Если не указывать индекс, то функция удалит последний элемент. Элемент можно удалить с помощью метода `remove`. `my_list.remove("")` Оператор `del` можно использовать для тех же целей: `del my_list[2]` `del my_list[1:3]`
11. Пример использования списковых включений для ввода данных в список: `a = [i for i in range(n)]` Использование функции `map`: `a = [1, 2, 3, 4, 5, 6, 7]` `b = list(map(lambda x: x**2, a))` `for i in a: if i%2 == 0: b.append(i)` Данный алгоритм можно реализовать с помощью функции `filter`: `a = [1, 2, 3, 4, 5, 6, 7]` `b = list(filter(lambda x: x % 2 == 0, a))`
12. Слайс задается тройкой чисел, разделенных запятой: `start:stop:step`. `Start` – позиция с которой нужно начать выборку, `stop` – конечная позиция, `step` – шаг. При этом необходимо помнить, что выборка не включает элемент, определяемый `stop`.
13. `len()`, `min()`, `max()`, `sum()`.
14. `list2 = my_list` создается ссылка на данный список.
15. `sorted()` возвращает новый отсортированный список, оставляя исходный список незатронутым. `list.sort()` сортирует список на месте, изменяет индексы списка и возвращает `None` (как и все операции на месте). `sorted()` работает с любыми итерациями, а не только со списками. Строки, кортежи, словари (вы получите ключи), генераторы и т. Д., возвращающие список, содержащий все элементы, отсортированные.
  - Используйте `list.sort()`, когда вы хотите изменить список, `sorted()`, когда вы хотите вернуть новый отсортированный объект. Используйте `sorted()`, когда вы хотите отсортировать что-то, что является повторяемым, а не списком.
  - Для списков `list.sort()` быстрее, чем `sorted()`, потому что ему не нужно создавать копию. Для любой другой операции у вас нет выбора.