

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное
образовательное учреждение высшего образования
«Северо-Кавказский федеральный университет»**

Кафедра инфокоммуникаций

Отчет по лабораторной работе №2.4

по дисциплине «Основы программной инженерии»

Выполнил студент группы ПИЖ-б-о-20-1

Турклиев В.Н. « »_____20_г.

Подпись студента_____

Работа защищена « »_____20_г.

**Проверил Воронкин Р.А. _____
(подпись)**

ВЫПОЛНЕНИЕ

Пример 1.

```
1  ▶  #!/usr/bin/env python3
2  # -*- coding: utf-8 -*-
3
4  import sys
5
6
7  ▶  if __name__ == '__main__':
8      # Ввести список одной строкой.
9      A = list(map(int, input().split()))
10     # Проверить количество элементов списка.
11     if len(A) != 10:
12         print("Неверный размер списка", file=sys.stderr)
13         exit(1)
14
15     # Найти искомую сумму.
16     s = 0
17     for item in A:
18         if abs(item) < 5:
19             s += item
20
21     print(s)
22
23     if __name__ == '__main__':
```

1 x

C:\Users\Vova\AppData\Local\Programs\Python\Python39\python.e

4 5 -2 3 6 8 9 1 9 10

6

Process finished with exit code 0

Пример 2.

```
if __name__ == '__main__':
    # Ввести список одной строкой.
    a = list(map(int, input().split()))
    # Если список пуст, завершить программу.
    if not a:
        print("Заданный список пуст", file=sys.stderr)
        exit(1)

    # Определить индексы минимального и максимального элементов.
    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

    # Проверить индексы и обменять их местами.
    if i_min > i_max:
        i_min, i_max = i_max, i_min

    # Посчитать количество положительных элементов.
    count = 0
    for item in a[i_min+1:i_max]:
        if item > 0:
            count += 1

    print(count)

if __name__ == '__main__':
```

2 x

C:\Users\Vova\AppData\Local\Programs\Python\Python39\python.exe "C:/U

1 2 5 -3 4 6 7 10 7 5

3

Индивидуальное задание 1.

```
elements = [1.989, 0.117, -1.363, 1.098, 0.039, 0.949, -0.026, 1.039, -0.842, 1.339]
sumOfOdd = 0
for i in range(len(elements)):
    if (i + 1) % 2 != 0:
        sumOfOdd += elements[i]
print(f'Сумма элементов с нечетными номерами: {round(sumOfOdd, 3)}')

indices = []
for i in elements:
    if i < 0:
        indices.append(elements.index(i))

print(f'Сумма элементов между первым и последним отрицательными значениями:'
      f' {sum(elements[(indices[0] + 1):(indices[-1] + 1)])}')
```

ind1 x

```
C:\Users\Vova\AppData\Local\Programs\Python\Python39\python.exe "C:/Users/Vova/Desktop/уче
Сумма элементов с нечетными номерами: -0.203
Сумма элементов между первым и последним отрицательными значениями: 3.099

Process finished with exit code 0
```

Индивидуальное задание 2.

```
algebra = [
    [4, 3, 2, 4, 3, 3, 4, 4],
    [4, 2, 4, 5, 4, 2, 4, 3],
    [5, 5, 4, 5, 3, 3, 4, 3],
    [5, 5, 3, 4, 2, 4, 4, 3],
    [2, 5, 4, 2, 3, 4, 5, 5],
    [5, 5, 3, 5, 4, 3, 3, 3]
]

geometry = [
    [3, 3, 3, 5, 2, 5, 3, 2],
    [3, 4, 2, 2, 3, 4, 5, 5],
    [5, 5, 5, 4, 3, 5, 3, 5],
    [2, 2, 3, 2, 3, 3, 3, 5],
    [2, 3, 2, 3, 4, 5, 3, 5],
    [4, 4, 4, 3, 3, 3, 4, 5]
]

physics = [
    [3, 5, 2, 3, 5, 3, 5, 4],
    [3, 4, 2, 5, 5, 5, 2, 4],
    [5, 3, 3, 5, 4, 5, 3, 4],
    [5, 2, 3, 3, 5, 4, 4, 4],
    [5, 5, 3, 3, 4, 3, 3, 2],
    [5, 3, 5, 3, 4, 3, 3, 4]
]

average = 0
numberOfRations = 0
for i in algebra:
    average += sum(i)
    numberOfRations += len(i)
print(f'Средняя оценка по алгебре среди учеников: {average / numberOfRations}')

noDeuce = 0
for i in range(6):
    if not (2 in algebra[i] or 2 in geometry[i] or 2 in physics[i]):
        noDeuce += 1
print(f'Учеников, не имеющих двойки по предметам: {noDeuce}')
```

ind2 x

Средняя оценка по алгебре среди учеников: 3.7083333333333335

Учеников, не имеющих двойки по предметам: 2

Process finished with exit code 0

Ссылка - <https://github.com/vegas007gof/lab7>

Ответы на контрольные вопросы

1. Список (list) – это Для создания списка нужно заключить элементы в квадратные скобки.
2. В оперативной памяти списки хранятся в виде ссылок, в которых хранятся ссылки на другие элементы.
3. Перебрать элементы списка можно с помощью цикла.
4. Сложение и умножение.
5. Для проверки принадлежности элемента списку существует оператор `in`
6. `s.count('')`
7. `s.insert(1,'')` – Вставить после заданного индекса, `s.append('')` – Вставить в конец списка.
8. `s.sort()`
9. Для удаления можно использовать:
 - a. `s.pop(index)`
 - b. `s.remove('element')`
 - c. `del s[0]`, `del s[1:3]`
10. Абстракция списков или списковое включение, является частью синтаксиса языка, которая предоставляет простой способ построения списков.
11. Срезы списков работают одинаково со срезами строк: `s[start:stop:step]`
12. Функции агрегации:
 - a. `len(s)` – число элементов в списке
 - b. `min(s)` – минимальный элемент списка
 - c. `max(s)` – максимальный элемент списка
 - d. `sum(s)` – сумма элементов списка
13. `es = s.copy()`
14. `s.sort()` сортирует исходный список, `sorted` возвращает отсортированный список.
15. структура данных для хранения объектов различных типов