

**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 №7 по дисциплине основы программной  
инженерии**

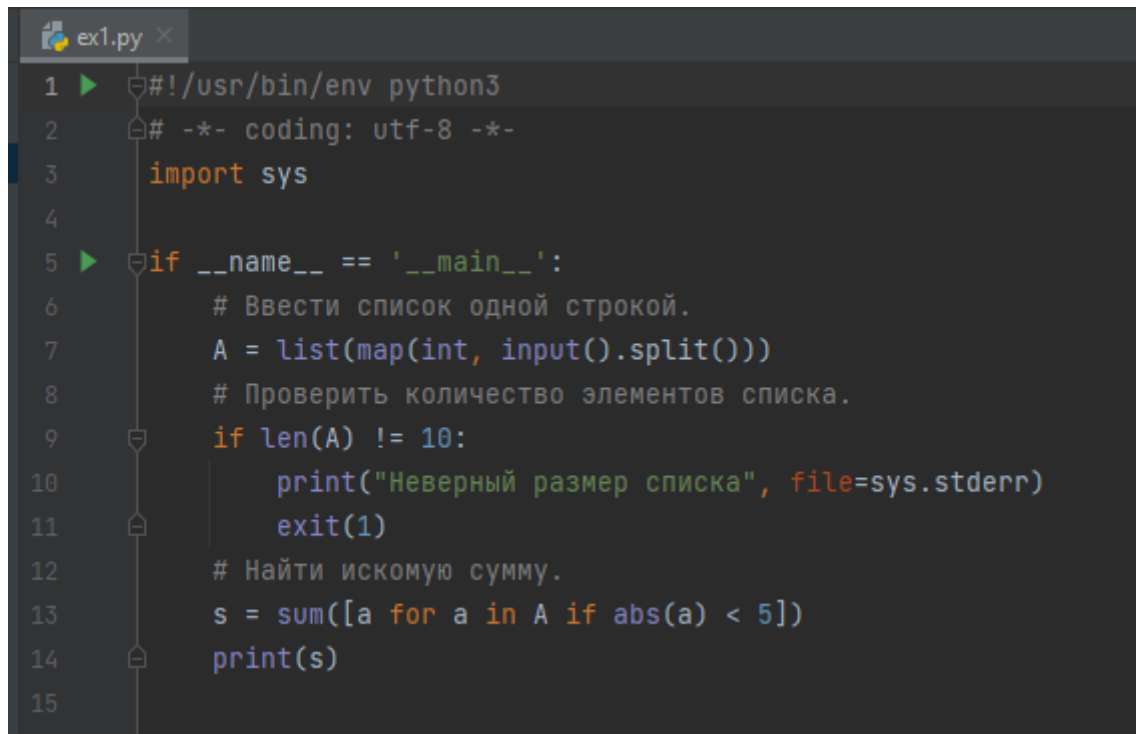
Выполнил:  
Выходцев Егор Дмитриевич,  
2 курс, группа ПИЖ-б-о-20-1,

Проверил:  
Доцент кафедры инфокоммуникаций,  
Воронкин Р.А.

Ставрополь, 2021 г

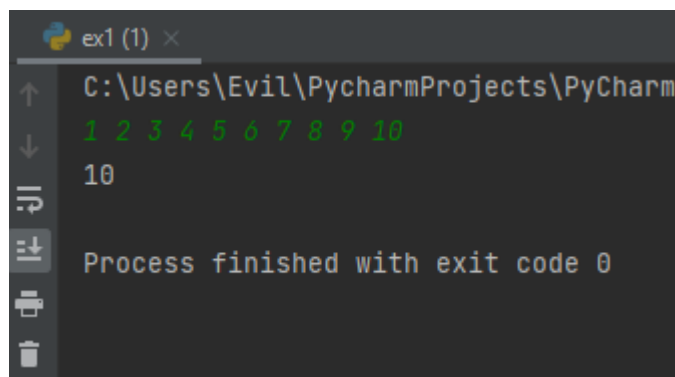
## 1. Работа со списками в языке Python

### 1.1 Пример 1 (рис. 1, 2, 3).



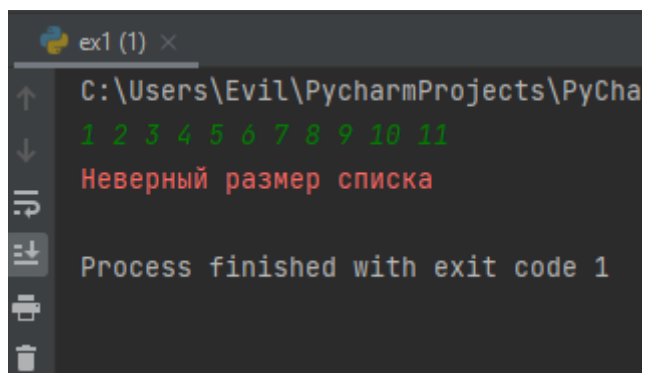
```
ex1.py x
1  ▶  #!/usr/bin/env python3
2      # -*- coding: utf-8 -*-
3      import sys
4
5  ▶  if __name__ == '__main__':
6      # Ввести список одной строкой.
7      A = list(map(int, input().split()))
8      # Проверить количество элементов списка.
9      if len(A) != 10:
10         print("Неверный размер списка", file=sys.stderr)
11         exit(1)
12     # Найти искомую сумму.
13     s = sum([a for a in A if abs(a) < 5])
14     print(s)
15
```

Рисунок 1 – Код программы



```
ex1 (1) x
C:\Users\Evil\PycharmProjects\PyCharm
1 2 3 4 5 6 7 8 9 10
10
Process finished with exit code 0
```

Рисунок 2 – Пример вывода программы



The image shows a terminal window titled 'ex1 (1)'. The output of the program is as follows:

```
C:\Users\Evil\PycharmProjects\PyCha
1 2 3 4 5 6 7 8 9 10 11
Неверный размер списка
Process finished with exit code 1
```

The numbers 1 through 11 are printed in green. The error message 'Неверный размер списка' (Incorrect list size) is printed in red. The final line indicates the process finished with exit code 1.

Рисунок 3 – Вывод программы при неверном количестве элементов в списке

1.2 Пример 2 (рис. 4, 5, 6).

```
ex1.py x ex2.py x
1 ▶ #!/usr/bin/env python3
2 # -*- coding: utf-8 -*-
3 import sys
4
5 ▶ if __name__ == '__main__':
6     # Ввести список одной строкой.
7     a = list(map(int, input().split()))
8     # Если список пуст, завершить программу.
9     if not a:
10         print("Заданный список пуст", file=sys.stderr)
11         exit(1)
12     # Определить индексы минимального и максимального элементов.
13     a_min = a_max = a[0]
14     i_min = i_max = 0
15     for i, item in enumerate(a):
16         if item < a_min:
17             i_min, a_min = i, item
18         if item >= a_max:
19             i_max, a_max = i, item
20     # Проверить индексы и обменять их местами.
21     if i_min > i_max:
22         i_min, i_max = i_max, i_min
23     # Посчитать количество положительных элементов.
24     count = 0
25     for item in a[i_min+1:i_max]:
26         if item > 0:
27             count += 1
28     print(count)
29
```

Рисунок 4 – Код программы

```
ex2 x
C:\Users\Evil\PycharmProjects\PyCharm
1 2 -5 -6 -7 1 2 7 9 10
4
Process finished with exit code 0
```

Рисунок 5 – Пример вывода программы

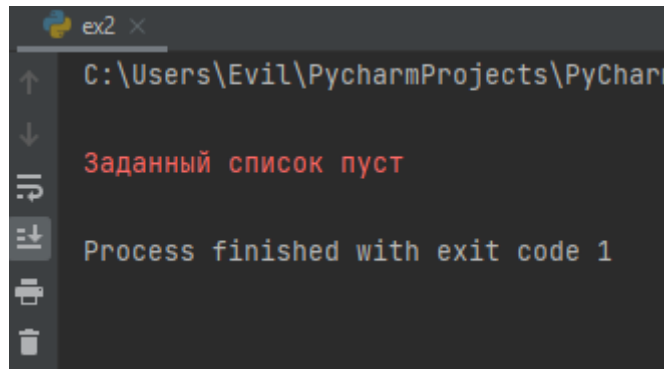


Рисунок 6 – Вывод программы при пустом списке

## Решение индивидуальных заданий

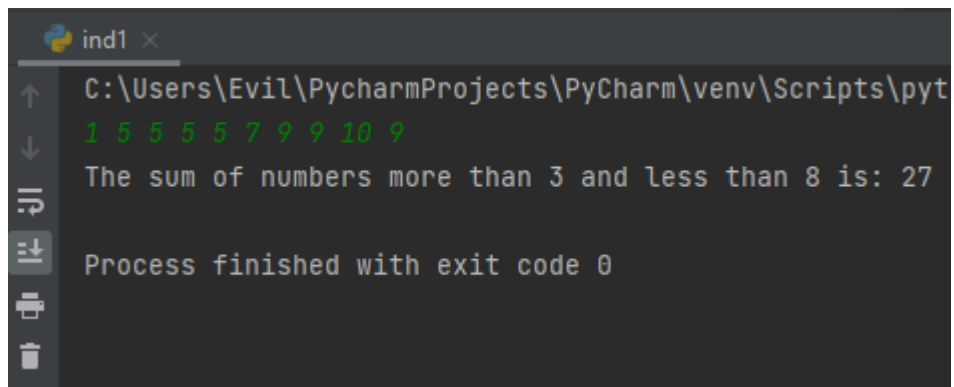
### Вариант 5

#### 1.3 Индивидуальное задание №1 (рис. 6, 7).

Условие: ввести список A из 10 элементов, найти сумму элементов, больших 3 и меньших 8 и вывести ее на экран.

```
ex1.py x ex2.py x ind1.py x
1 ▶ #!/usr/bin/env python3
2 # -*- coding: utf-8 -*-
3
4 import sys
5
6 ▶ if __name__ == '__main__':
7     A = list(map(int, input().split()))
8     if len(A) != 10:
9         print("Неверный размер списка", file=sys.stderr)
10        exit(1)
11    s = sum([a for a in A if 3 < a < 8])
12    print(f"The sum of numbers more than 3 and less than 8 is: {s}")
13
```

Рисунок 6 – Код программы

A screenshot of a PyCharm terminal window. The title bar shows a Python icon and the text 'ind1 x'. The terminal output is as follows:

```
C:\Users\Evil\PycharmProjects\PyCharm\venv\Scripts\python.exe  
1 5 5 5 5 7 9 9 10 9  
The sum of numbers more than 3 and less than 8 is: 27  
Process finished with exit code 0
```

On the left side of the terminal, there is a vertical toolbar with icons for navigating through the code (up, down, search, etc.) and a trash icon at the bottom.

Рисунок 7 – Пример работы программы

#### 1.4 Индивидуальное задание №2 (рис 8, 9).

Условие: в списке, состоящем из вещественных элементов, вычислить:

1. максимальный элемент списка;
2. сумму элементов списка, расположенных до последнего положительного элемента.

Сжать список, удалив из него все элементы, модуль которых находится в интервале  $[a, b]$ .

Освободившиеся в конце списка элементы заполнить нулями.

```
ind2.py x
1 ▶ #!/usr/bin/env python3
2 # -*- coding: utf-8 -*-
3
4 ▶ if __name__ == '__main__':
5     print("Enter the elements of the list a:")
6     A = list(map(float, input().split()))
7     length = len(A)
8     print(length)
9     print(f"The max element of this list is: {max(A)}")
10    i = 0
11    for el in A:
12        if el >= 0:
13            i = A.index(el)
14    s = sum([a for a in A if A.index(a) < i])
15    print(f"The sum of numbers before the last positive element is: {s}")
16    a = int(input("Enter the a number of the border:"))
17    b = int(input("Enter the b number of the border:"))
18    temp = []
19    for el in A:
20        if a < abs(el) < b:
21            continue
22        else:
23            temp.append(el)
24    if length > len(temp):
25        for i in range(length - len(temp)):
26            temp.append(0)
27    print(temp)
28
```

Рисунок 8 – Код программы

```
ind2 x
C:\Users\Evil\PycharmProjects\PyCharm\venv\Scripts\python.exe C:/Users/Evil/
Enter the elements of the list a:
1 226.9 -7 -200 3 8 9 7 2 10
10
The max element of this list is: 226.9
The sum of numbers before the last positive element is: 49.900000000000006
Enter the a number of the border:2
Enter the b number of the border:9
[1.0, 226.9, -200.0, 9.0, 2.0, 10.0, 0, 0, 0, 0]

Process finished with exit code 0
```

## Рисунок 9 – Пример работы программы

### 2. Ответы на контрольные 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('<значение элемента>')
```

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.