

**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**

Выполнил:

Кожухов Филипп Денисович,
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.

ВЫПОЛНЕНИЕ:

```
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 7.1 - Код примера 1

```
"C:\Users\CMDR Inferion\AppData\Local\Microsoft\WindowsApps\
1 2 3 4 5 6 7 8 9 0
10
Process finished with exit code 0
|
```

Рисунок 7.2 - Вывод программы

```
"C:\Users\CMDR Inferion\AppData\Local\Microsoft\WindowsApps\
1 2 3 4 5 6 7 8 8 9 9 2 2
Неверный размер списка
Process finished with exit code 1
|
```

Рисунок 7.3 - Результат неправильного ввода

```

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 7.4 - Код примера 2

```

"C:\Users\CMDR Inferion\AppData\Local\Microsoft\WindowsAp
1 2 -5 -6 -7 1 2 7 9 10
4

Process finished with exit code 0

```

Рисунок 7.5 - Вывод программы

ИНДИВИДУАЛЬНЫЕ ЗАДАНИЯ

Вариант 5

```
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 7.6 - Код задания 1

```
"C:\Users\CMDR Inferion\AppData\Local\Microsoft\WindowsApps\python3.8.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
|
```

Рисунок 7.7 - Вывод программы

```

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

```

Рисунок 7.8 - Код задания 2

```

"C:\Users\CMDR Inferion\AppData\Local\Microsoft\WindowsApps\python3.8.exe" "C:/Users/CMDR Infer
Enter the elements of the list a:
1 425.4 -2 43 95 3 5 59 7 8
10
The max element of this list is: 425.4
The sum of numbers before the last positive element is: 636.4
Enter the a number of the border:3
Enter the b number of the border:20
[1.0, 425.4, -2.0, 43.0, 95.0, 59.0, 0, 0, 0, 0]

```

Рисунок 7.9 - Вывод программы

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('<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.