

Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт перспективной инженерии
Департамент цифровых, робототехнических систем и электроники

ОТЧЕТ
ПО ЛАБОРАТОРНОЙ РАБОТЕ №7
дисциплины
«Основы кроссплатформенного программирования»
Вариант 4

Выполнил:
Борцов Богдан Михайлович
2 курс, группа ИТС-б-о-23-1,
11.03.02 «Инфокоммуникационные
технологии и системы связи», очная
форма обучения

(подпись)

Проверил:
Доцент департамента цифровых,
робототехнических систем и
электроники Воронкин Р.А.

(подпись)

Отчет защищен с оценкой _____ Дата защиты _____

Ставрополь, 2024 г.

Тема: Работа со списками в языке Python

Цель: приобретение навыков по работе со списками при написании программ с помощью языка программирования версии 3.x.

Порядок выполнения работы:

1. Изучил теоретический материал.
2. Приступил к выполнению заданий.
3. Создание репозитория
4. Клонирование репозитория
5. Проработал примеры

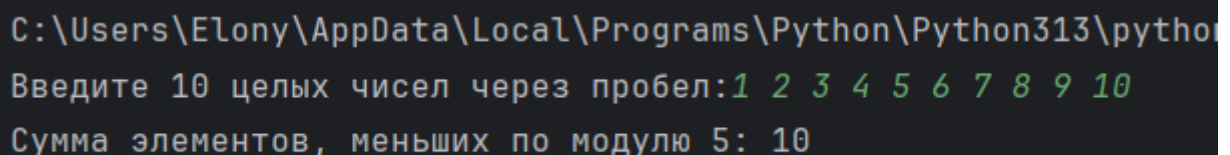
```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-

if __name__ == '__main__':
    # Ввод списка из 10 элементов
    A = list(map(int, input("Введите 10 целых чисел через
пробел:").split()))

    # Проверка, что введено именно 10 элементов
    if len(A) != 10:
        print("Ошибка: необходимо ввести ровно 10 целых чисел.",
file=sys.stderr)
        exit(1)

    # Вычисление суммы элементов, меньших по модулю 5
    total_sum = sum(item for item in A if abs(item) < 5)

    # Вывод суммы
    print("Сумма элементов, меньших по модулю 5:", total_sum)
```



```
C:\Users\Elony\AppData\Local\Programs\Python\Python313\python
Введите 10 целых чисел через пробел:1 2 3 4 5 6 7 8 9 10
Сумма элементов, меньших по модулю 5: 10
```

Рисунок 1. Пример 1

```
def count_positive_between_min_max(lst):
```

```

if not lst:
    return 0 # Если список пустой, возвращаем 0

min_value = min(lst)
max_value = max(lst)

# Находим индексы минимального и максимального элементов
min_index = lst.index(min_value)
max_index = lst.index(max_value)

# Убедимся, что индексы корректные (первый индекс - меньший)
start_index = min(min_index, max_index) + 1
end_index = max(min_index, max_index)

# Считаем количество положительных элементов между ними
positive_count = sum(1 for x in lst[start_index:end_index] if x > 0)

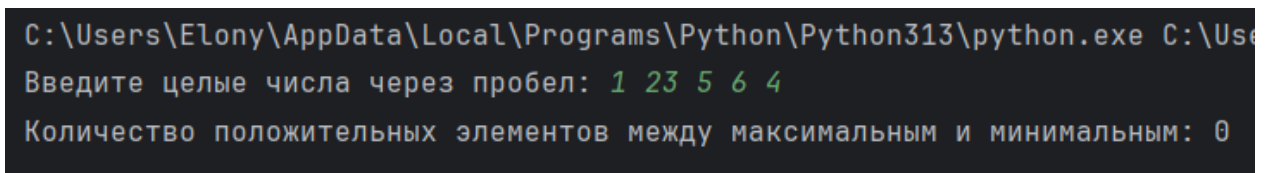
return positive_count

if __name__ == '__main__':
    # Ввод списка целых чисел
    A = list(map(int, input("Введите целые числа через пробел: ").split()))

    # Получаем количество положительных элементов между минимальным и
    максимальным
    result = count_positive_between_min_max(A)

    print("Количество положительных элементов между максимальным и
    минимальным:", result)

```



```

C:\Users\Efony\AppData\Local\Programs\Python\Python313\python.exe C:\Use
Введите целые числа через пробел: 1 23 5 6 4
Количество положительных элементов между максимальным и минимальным: 0

```

Рисунок 2. Пример 2

6. Приступил к выполнению индивидуального задания

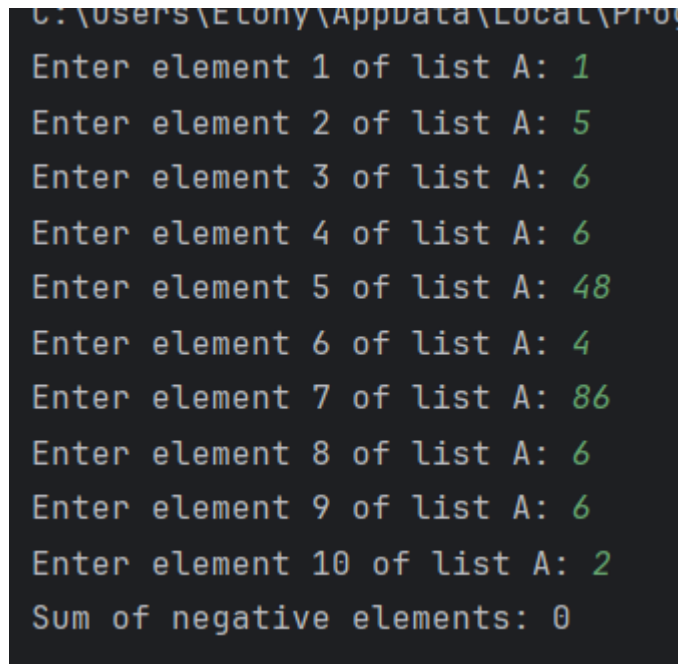
```

# Ввод списка A из 10 элементов
A = [int(input(f'Enter element {i+1} of list A: ')) for i in range(10)]

# Нахождение суммы отрицательных элементов
negative_sum = sum(x for x in A if x < 0)

```

```
# Вывод результата
print(f"Sum of negative elements: {negative_sum}")
```



```
C:\Users\Elony\AppData\Local\Programs\Python\Python39\python.exe
Enter element 1 of list A: 1
Enter element 2 of list A: 5
Enter element 3 of list A: 6
Enter element 4 of list A: 6
Enter element 5 of list A: 48
Enter element 6 of list A: 4
Enter element 7 of list A: 86
Enter element 8 of list A: 6
Enter element 9 of list A: 6
Enter element 10 of list A: 2
Sum of negative elements: 0
```

Рисунок 3. Задание 1

```
# Ввод списка вещественных чисел
n = int(input("Enter the number of elements in the list: "))
lst = [float(input(f"Enter element {i+1}: ")) for i in range(n)]

# 1. Сумма элементов списка с нечётными номерами (индексы 1, 3, 5, ...)
sum_odd_index = sum(lst[i] for i in range(1, len(lst), 2))

# 2. Сумма элементов между первым и последним отрицательными
элементами
first_negative_index = next((i for i in range(len(lst)) if lst[i] < 0), -1)
last_negative_index = next((i for i in range(len(lst) - 1, -1, -1) if lst[i] < 0), -1)

if first_negative_index != -1 and last_negative_index != -1 and
first_negative_index < last_negative_index:
    sum_between_negatives = sum(lst[first_negative_index +
1:last_negative_index])
else:
    sum_between_negatives = 0 # Если отрицательные элементы отсутствуют
или их меньше двух

# 3. Сжатие списка
compressed_list = [x for x in lst if abs(x) > 1]
```

```
compressed_list += [0] * (len(lst) - len(compressed_list)) # Добавление нулей в
конец
```

```
# Вывод результатов
```

```
print(f"Sum of elements with odd indices: {sum_odd_index}")
```

```
print(f"Sum of elements between the first and last negative elements:
```

```
{sum_between_negatives}")
```

```
print(f"Compressed list: {compressed_list}")
```

```
Enter the number of elements in the list: 4
Enter element 1: 1
Enter element 2: 2
Enter element 3: 3
Enter element 4: 4
Sum of elements with odd indices: 6.0
Sum of elements between the first and last negative elements: 0
Compressed list: [2.0, 3.0, 4.0, 0]
```

Рисунок 4. Задание 2

7. Зафиксировал изменения

Ответы на контрольные вопросы:

1. Что такое списки в языке Python?

Списки в Python — это изменяемые последовательности, которые могут содержать элементы различных типов, включая числа, строки и другие объекты. Списки позволяют хранить и управлять коллекциями данных.

2. Как осуществляется создание списка в Python?

Список создается с помощью квадратных скобок [], также можно использовать функцию list.

3. Как организовано хранение списков в оперативной памяти?

Списки хранятся в виде массивов указателей на объекты. Каждый элемент списка хранит ссылку на объект, а не сам объект. Это позволяет спискам содержать элементы разных типов.

4. Каким образом можно перебрать все элементы списка?

Перебор элементов списка можно осуществить с помощью цикла for.

5. Какие существуют арифметические операции со списками?

Основные арифметические операции со списками: конкатенация—объединяет два списка. Повторение — повторяет список n раз.

6. Как проверить есть ли элемент в списке?

Через оператора in if element in my list

7. Как определить число вхождений заданного элемента в списке?

Через метод count

8. Как осуществляется добавление (вставка) элемента в список?

Через метод append для добавления в конец списка или insert для вставки по индексу.

9. Как выполнить сортировку списка?

Через sort для сортировки списка на месте или функцию sorted для создания нового отсортированного списка

10. Как удалить один или несколько элементов из списка?

Через remove для удаления первого вхождения элемента, pop(index) для удаления элемента по индексу, или del для удаления по индексу

11. Что такое списковое включение и как с его помощью осуществлять обработку списков?

Списковое включение — это способ создания нового списка на основе существующего.

12. Как осуществляется доступ к элементам списков с помощью срезов?

Срезы позволяют извлекать подпоследовательности. Синтаксис: my list[start:end], где start — начальный индекс, а end — конечный индекс.

13. Какие существуют функции агрегации для работы со списками?

Основные функции: sum — сумма элементов. min — минимальный элемент. max — максимальный элемент. len — количество элементов.

14. Как создать копию списка?

Через метод сору

15. Самостоятельно изучите функцию sorted языка Python. В чем ее отличие от метода sort списков?

Функция sorted создает новый отсортированный список и не изменяет оригинальный, а метод sort сортирует список на месте и возвращает None

Вывод: В ходе работы освоены основные операции со списками в языке Python 3.x, включая создание, сортировку, фильтрацию, добавление и удаление элементов, а также работу с вложенными списками. Реализованы примеры программ, протестированных на различных вводных данных, с результатами, представленными в отчёте. Репозиторий на GitHub использовался для хранения кода, настроен файл .gitignore для PyCharm.