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

Институт цифрового развития
Кафедра инфокоммуникаций

ОТЧЕТ
ПО ПРАКТИЧЕСКОЙ РАБОТЕ №2.4
дисциплины «Программирование на Python»
Вариант №2

Выполнила:
Беседина Инга Олеговна
2 курс, группа ИВТ-б-о-22-1,
09.03.01 «Информатика и вычислительная техника», очная форма обучения

_____
(подпись)

Руководитель практики:
Воронкин Р. А., канд. технических наук, доцент, доцент кафедры инфокоммуникаций

_____
(подпись)

Отчет защищен с оценкой _____ Дата защиты_____

Ставрополь, 2023 г.

Тема: Работа со списками в языке Python

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

Ход работы

Пример 1. *Ввести список A из 10 элементов, найти сумму элементов, меньших по модулю 5, и вывести ее на экран.*

```
#!/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([i for i in A if abs(i) < 5])
    print(s)
```

```
PS C:\Rep\Python_LR_2_4\Project>
mple1.py
3 4 6 1 2 6 7 3 9 8
13
```

Рисунок 1. Результат работы программы

Пример 2. *Написать программу, которая для целочисленного списка определяет, сколько положительных элементов располагается между его максимальным и минимальным элементами.*

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


import sys
```

```python
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 = len([i for i in a[i_min+1:i_max] if i > 0])
    print(count)
```

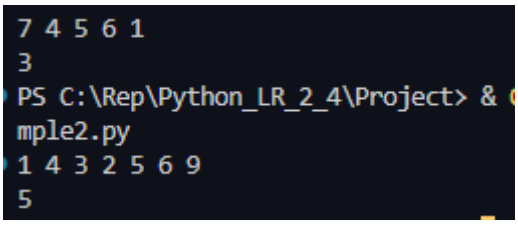

Рисунок 2. Результат работы программы

Ввести список *A* из 10 элементов, найти произведение положительных элементов и вывести его на экран.

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


import sys


if __name__ == '__main__':
    A = list(map(int, input().split()))
```

```
    if len(A) != 10:
        print("Неверный размер списка", file=sys.stderr)
        exit(1)

    m = 1
    for i in A:
        if i > 0:
            m *= i

    print(m)
```

```
-2 3 4 -5 -7 2 -6 -9 -12 2
48
```

Рисунок 3. Результат работы программы

2. В списке, состоящем из вещественных элементов, вычислить:

1. сумму положительных элементов списка;
2. произведение элементов списка, расположенных между максимальным по модулю и минимальным по модулю элементами.

Упорядочить элементы списка по убыванию.

```
#!/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)

    s = sum([i for i in a if i > 0])

    a_min = a_max = abs(a[0])
    i_min = i_max = 0
    for i, item in enumerate(a):
        if abs(item) < a_min:
            i_min, a_min = i, abs(item)
        if abs(item) >= a_max:
            i_max, a_max = i, abs(item)
```

```
    if i_min > i_max:
        i_min, i_max = i_max, i_min

    m = 1
    for item in a[i_min+1:i_max]:
        m *= item

    print(f"Сумма положительных элементов списка: {s}")
    print("Произведение элементов списка, расположенных между"
          f"максимальным и минимальным элементами: {m}")

    a.sort()
    print(f"Элементы списка по возрастанию: {a}")
```

```
-4 -8 2 3 4 1 5
Сумма положительных элементов списка: 15
Произведение элементов списка, расположенных между максимальным и минимальным элементами: 24
Элементы списка по возрастанию: [-8, -4, 1, 2, 3, 4, 5]
```

Рисунок 3. Результат работы программы

Контрольные вопросы:

1. Список (list) – это структура данных для хранения объектов различных типов. Размер списка не статичен, его можно изменять. Список по своей природе является изменяемым типом данных. Переменная, определяемая как список, содержит ссылку на структуру в памяти, которая в свою очередь хранит ссылки на какие-либо другие объекты или структуры

2. Для создания списка нужно заключить элементы в квадратные скобки: my_list = [1, 2, 3, 4, 5]

3. Список является изменяемым типом данных. При его создании в памяти резервируется область, которую можно условно назвать некоторым “контейнером”, в котором хранятся ссылки на другие элементы данных в памяти. В отличии от таких типов данных как число или строка, содержимое “контейнера” списка можно менять

4. Перебрать элементы списка можно с помощью цикла for

5. Для объединения списков можно использовать оператор сложения (+); список можно повторить с помощью оператора умножения (*)

6. Для того, чтобы проверить, есть ли заданный элемент в списке Python необходимо использовать оператор in

7. Метод count можно использовать для определения числа сколько раз данный элемент встречается в списке

8. Метод append можно использовать для добавления элемента в список. Метод insert можно использовать, чтобы вставить элемент в список

9. Для сортировки списка нужно использовать метод sort

10. Удалить элемент можно, написав его индекс в методе pop. Элемент можно удалить с помощью метода remove. Оператор del можно использовать для тех же целей. Можно удалить все элементы из списка с помощью метода clear

11. List Comprehensions чаще всего на русский язык переводят как абстракция списков или списковое включение, является частью синтаксиса языка, которая предоставляет простой способ построения списков. a = [i for i in range(int(input()))]

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. Для работы со списками Python предоставляет следующие функции: len(L) - получить число элементов в списке L . min(L) - получить минимальный элемент списка L . max(L) - получить максимальный элемент списка L . sum(L) - получить сумму элементов списка L , если список L содержит только числовые значения

14. Для создания копии списка необходимо использовать либо метод copy, либо использовать оператор среза (список[:])

15. list.sort() сортирует список на месте, изменяя его индексы и возвращая None, тогда как sorted() возвращает новый отсортированный список, оставляя исходный список неизменным. Другое отличие состоит в

том, что sorted() принимает любые итерации, в то время как list.sort() является методом класса списка и может использоваться только со списками.

Вывод: В ходе выполнения лабораторной работы были приобретены навыков по работе со списками при написании программ с помощью языка программирования Python версии 3.x.