

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное
образовательное учреждение высшего образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»**

Кафедра инфокоммуникаций

Основы кроссплатформенного программирования

Отчет по лабораторной работе №2.4

Работа со списками в языке Python

Выполнил студент группы

ИВТ-б-о-21-1

Мальцев Н.А. « » _____ 20__ г.

Подпись студента _____

Работа защищена « » _____ 20__ г.

Проверил доцент

Кафедры инфокоммуникаций, старший
преподаватель

Воронкин Р.А.

(подпись)

Ставрополь 2022

Работа со списками в языке Python.

Цель работы: приобретение навыков по работе со списками при написании программ с помощью языка программирования Python версии 3.10.

Порядок выполнения работы:

1) Проработка примеров.

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

Код программы:

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

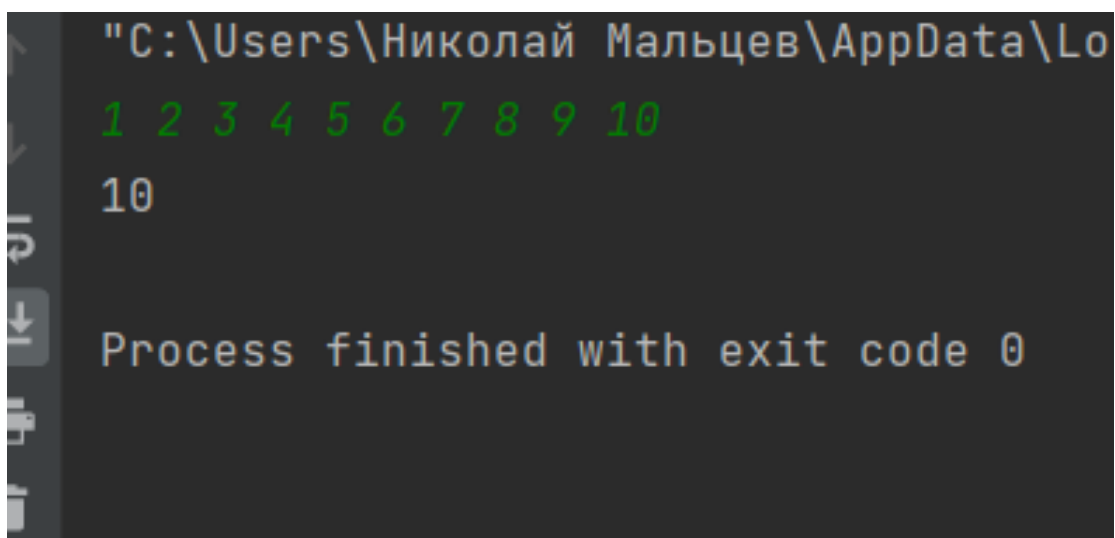
import sys

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

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

    s = 0
    for item in A:
        if abs(item) < 5:
            s += item
    print(s)
```

Результат работы программы:



```
"C:\Users\Николай Мальцев\AppData\Lo
1 2 3 4 5 6 7 8 9 10
10
Process finished with exit code 0
```

Рисунок 1. Результат работы программы из примера 1

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

Код программы:

```
#!/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)

    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

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

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

Результат программы:

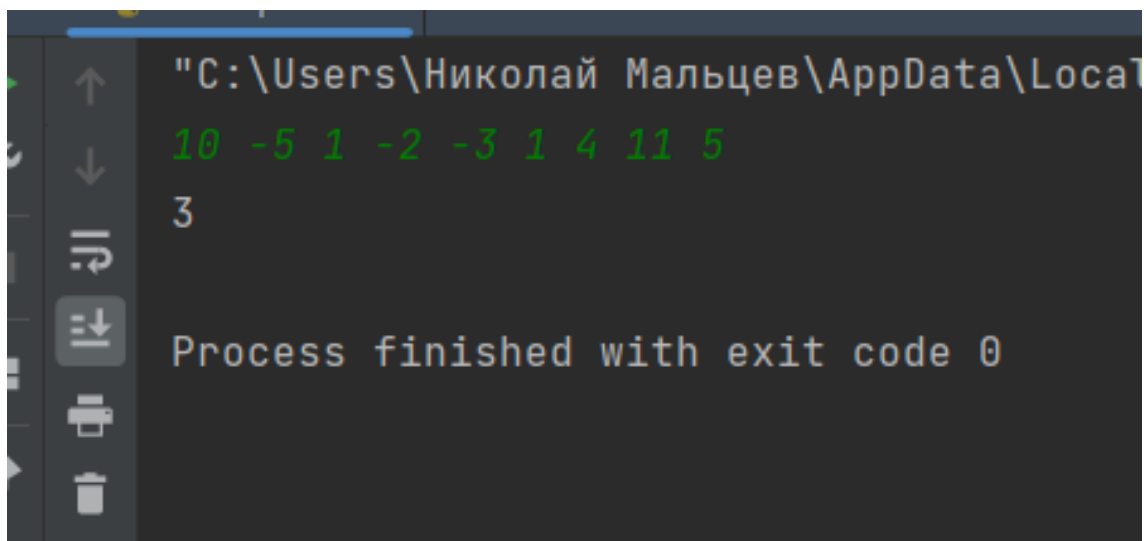


Рисунок 2. Результат выполнения программы из примера 2

2) Выполнение индивидуальных заданий (вариант 9).

Пример 1. Составить программу, выдающую индексы заданного элемента или сообщаящую, что такого элемента в списке нет.

Код программы:

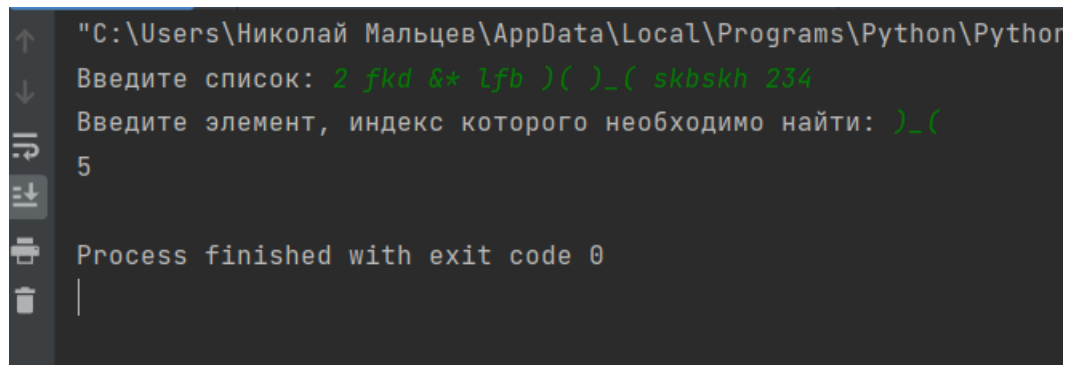
```
#!/usr/bin/env python3
# __coding: utf-8 __

import sys

if __name__ == '__main__':
    #Ввод списка одной строкой.
    a = list(input("Введите список: ").split())
    #Ошибка, если список пуст.
    if not a:
        print("Заданный список пуст", file=sys.stderr)
        exit(1)

    elem = input("Введите элемент, индекс которого необходимо найти: ")
    if elem in a:
        print(a.index(elem))
    else:
        print("Данный элемент отсутствует в списке")
```

Результат работы программы:



```
"C:\Users\Николай Мальцев\AppData\Local\Programs\Python\Python
Введите список: 2 fkd &* lfb )( )_( skbskh 234
Введите элемент, индекс которого необходимо найти: )(
5
Process finished with exit code 0
```

Рисунок 3. Результат работы программы к индивидуальному заданию 1

Пример 2. В списке, состоящем из целых элементов, вычислить:

1. Минимальный по модулю элемент списка.

Код программы:

```
#!/usr/bin/env python3
# __coding: utf-8 __

import sys

if __name__ == '__main__':
    a = list(map(int, input("Введите список из целых элементов: ").split()))
    if not a:
```

```

print("Заданный список пуст", file=sys.stderr)
exit(1)

a_min = abs(a[0])
i_min = 0
for i in a:
    if abs(i) < a_min:
        a_min = abs(i)
        i_min = a.index(i)

print("Минимальный по модулю элемент: ", a[i_min])

```

Результат работы программы:

```

"C:\Users\Николай Мальцев\AppData\Local\Programs\Python\Python38\python.exe"
Введите список из целых элементов: -2 3 -4 -1 5 10 -11 2
Минимальный по модулю элемент: -1

Process finished with exit code 0

```

Рисунок 4. Результат работы программы к индивидуальному заданию 2 пункт 1

2. Сумму модулей элементов списка, расположенных после первого элемента, равного нулю.

Код программы:

```

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

import sys

if __name__ == '__main__':
    a = list(map(int, input("Введите список из целых элементов: ").split()))
    if not a:
        print("Заданный список пуст", file=sys.stderr)
        exit(1)

    ind = a.index(0)
    s = 0
    for n, i in enumerate(a):
        if n > ind:
            s += i
    print("Сумма элементов после нуля: ", s)

```

Результат работы программы:

```
"C:\Users\Николай Мальцев\AppData\Local\Programs\Python\Python310\p
Введите список из целых элементов: 1 2 3 0 -2 6 -3 5 8 -2 0 1 0 21
Сумма элементов после нуля: 34

Process finished with exit code 0
```

Рисунок 5. Результат работы программы к индивидуальному заданию 2 пункт 2

Вывод: в ходе работы были приобретены навыки работы со списками при программировании на языке Python версии 3.10.