

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:
Прокопов Дмитрий
Владиславович,
2 курс, группа ПИЖ-б-о-21-1,

Проверил:
Доцент кафедры инфокоммуникаций,
Воронкин Р.А.

Ставрополь, 2022 г

Ход работы

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

4 12 53 7 33 1 98 6 31 6
5

Рис. 7.1 – выполнение 1-го примера

Рис. 7.2 – 2-й пример

```
#!/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)
```

```
C:\Users\dmidt\PycharmProjects\7lb\venv\
Заданный список пуст
Process finished with exit code 1
```

Рис. 7.3 – вывод если список пуст

```
C:\Users\dmidt\PycharmProjects\7lb\venv\Scripts\python
42 62 7 74 21 44 76 88 21 9
4
Process finished with exit code 0
```

Рис. 7.4 – выполнение 2-го примера

```
4 if __name__ == '__main__':
5     A = list(map(int, input("Введите оценки по алгебре\n").split()))
6     G = list(map(int, input("Введите оценки по геометрии\n").split()))
7     F = list(map(int, input("Введите оценки по физике\n").split()))
8
9     a = g = f = 0
10    for i in range(len(A)):
11        if A[i] > G[i] and A[i] > F[i]:
12            a += 1
13        if G[i] > A[i] and G[i] > F[i]:
14            g += 1
15        if F[i] > A[i] and F[i] > G[i]:
16            f += 1
17
18    print("Кол-во оценок по алгебре лучше:", a)
19    print("Кол-во оценок по геометрии лучше:", g)
20    print("Кол-во оценок по физике лучше:", f)
21
22    if a > g and a > f:
23        print("Лучшие оценки по алгебре")
24    if g > a and g > f:
25        print("Лучшие оценки по геометрии")
26    if f > a and f > g:
27        print("Лучшие оценки по физике")
```

Рис. 7.5 – 1-е индивидуальное задание

```

Введите оценки по алгебре
4 5 3 5 4 3 5
Введите оценки по геометрии
4 3 3 4 5 3 4
Введите оценки по физике
5 5 3 4 5 4 5
Кол-во оценок по алгебре лучше: 1
Кол-во оценок по геометрии лучше: 0
Кол-во оценок по физике лучше: 2
Лучшие оценки по физике

```

Рис. 7.6 – выполнение программы

```

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

import sys

if __name__ == '__main__':
    A = list(map(int, input("Введите 10 элементов\n").split()))
    if len(A) != 10:
        print("Неверный размер массива", file=sys.stderr)
        exit(1)

    if min(A) < 0:
        pr = 0
        for i in range(len(A)):
            if A[i] < 0:
                pr += A[i]
        print(pr)
    else:
        print(0)

```

Рис.7.7 – 2-я индивидуальная работа

```

C:\Users\dmidt\PycharmProject
Введите 10 элементов
5 4 -2 34 -5 4 -3 72 9 0
-10

```

Рис. 7.8 – выполнение 2-ой инд. работы
Ответы на вопросы:

1. Список (list) – структура данных для хранения объектов различных типов
2. Для создания списка нужно заключить элементы в квадратные скобки
3. В оперативной памяти списки хранятся в виде ссылок, в которых хранятся ссылки на другие элементы

4. Перебрать элементы списка можно с помощью цикла
5. Арифметические операции: сложение и умножение
6. Для проверки принадлежности элемента списку существует оператор `in`
7. `s.count()` – проверка на элемент
8. `s.insert(1,)` – ставить после заданного индекса, `s.append()` – вставить в конец списка
9. `s.sort()` – сортировка
10. Для удаления можно использовать:
 `s.pop(index)`
 `s.remove("element")`
 `del s[0]`, `del s[1:3]`
11. Списковое включение, является частью синтаксиса языка, которая предоставляет простой способ построения списков
12. Срезы списков работают одинаково со срезами строк `s[start:stop:step]`
13. Функции агрегации:
 `len(s)` – число элементов в списке
 `min(s)` – минимальный элемент списка
 `max(s)` – максимальный элемент списка
 `sum(s)` – сумма элементов списка
14. `i-s.copy()` -копирование списка
15. `s.sort()` сортирует исходный список, `sorted` возвращает отсортированный список