Отчет №7
по дисциплине “Основы программной инженерии”
Группа: ПИЖ-б-о-20-1, Примаков В. Д

ВЫПОЛНЕНИЕ

Пример 1.

```python
#!/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 = 0
    for item in A:
        if abs(item) < 5:
            s += item

    print(s)
```

```
C:\Users\surai5a\Desktop\laba_2_4\p
4 5 -2 3 6 8 9 1 9 10
6

Process finished with exit code 0
```

Пример 2.

Отчет №7
по дисциплине “Основы программной инженерии”
Группа: ПИЖ-б-о-20-1, Примаков В. Д

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

import sys


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

    print(count)
```

```
C:\Users\surai5a\Desktop\laba_2_4\
1 2 5 -3 4 6 7 10 7 5
4

Process finished with exit code 0
```

Отчет №7
по дисциплине “Основы программной инженерии”
Группа: ПИЖ-б-о-20-1, Примаков В. Д

Индивидуальное задание 1.

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

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

import sys


print("Write list elements with spaces: ")
a = list(map(int, input().split()))
if not a:
    print("List is empty", file=sys.stderr)
    exit(1)
x = 1;
for i in a:
    if i < 0:
        x *= i
if x == 1:
    print("List doesn't contain negative numbers")
else:
    print(f"List: {a}\nProduct of negative numbers: {x}")
```

```
C:\Users\surai5a\Desktop\laba_2_4\pyProj\S
Write list elements with spaces:
1 2 3 4 -2 6 7 -5 9 -10
List: [1, 2, 3, 4, -2, 6, 7, -5, 9, -10]
Product of negative numbers: -100

Process finished with exit code 0
```

Отчет №7
по дисциплине “Основы программной инженерии”
Группа: ПИЖ-б-о-20-1, Примаков В. Д

Индивидуальное задание 2.

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

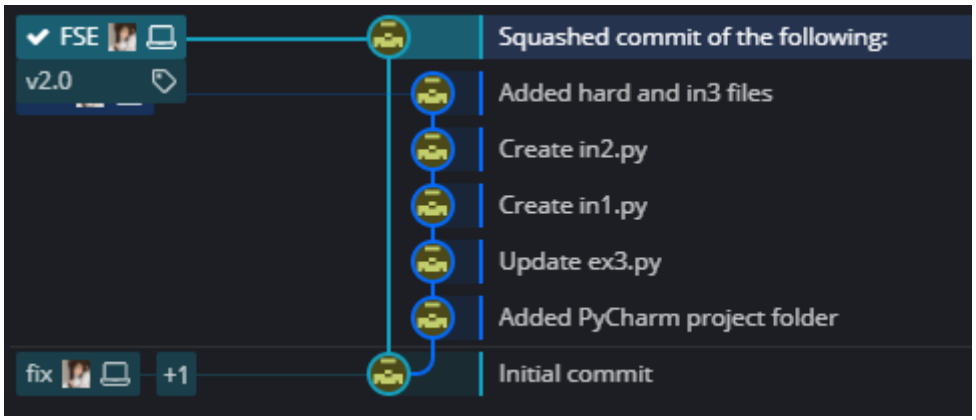
1. номер минимального элемента списка;
2. сумму элементов списка, расположенных между первым и вторым отрицательными элементами.

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

import sys


print("Write list elements with spaces: ")
a = list(map(int, input().split()))
if not a:
    print("List is empty", file=sys.stderr)
    exit(1)
ind_1, ind_2 = 11, 11
min = min(a)
min_ind = a.index(min)
sum = 0;
for ind, val in enumerate(a):
    if val < 0 and ind_1 == 11:
        ind_1 = ind
    elif val < 0 and (ind_1 != 11 and ind_2 == 11):
        ind_2 = ind
for i in a[ind_1 + 1:ind_2]:
    sum += i
print(f"Original list: {a}\nMinimal element index: a[{min_ind}] = {min}\n"
      f"Sum of numbers, between 1st and 2nd elements: {sum}")
```

```
C:\Users\surai5a\Desktop\laba_2_4\pyProj\Scripts\p
Write list elements with spaces:
4 5 6 -4 7 8 5 -6 9 -20
Original list: [4, 5, 6, -4, 7, 8, 5, -6, 9, -20]
Minimal element index: a[9] = -20
Sum of numbers, between 1st and 2nd elements: 20

Process finished with exit code 0
```

Отчет №7
по дисциплине “Основы программной инженерии”
Группа: ПИЖ-б-о-20-1, Примаков В. Д

Карта веток и коммитов



Ссылки на репозитории

GitHub - https://github.com/surai5a/laba_2_4

Ответы на контрольные вопросы

1. Список (list) – это 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. Сложение и умножение.
6. Для проверки принадлежности элемента списку существует оператор in
7. s.count(‘’)
8. s.insert(1,’’) – Вставить после заданного индекса, s.append(‘’) – Вставить в конец списка.
9. s.sort()
10. Для удаления можно использовать:
    a. s.pop(index)
    b. s.remove(‘element’)
    c. del s[0], del s[1:3]
11. Абстракция списков или 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:
    a. len(s) – число элементов в списке
    b. min(s) – минимальный элемент списка
    c. max(s) – максимальный элемент списка
    d. sum(s) – сумма элементов списка
14. es = s.copy()
15. s.sort() сортирует исходный список, sorted возвращает отсортированный список.