

Северо-Кавказский федеральный университет  
Институт математики и информационных технологий

**ОТЧЕТ**  
**о выполнении лабораторной работы №4**  
**по дисциплине**  
**«Основы Программной Инженерии»**

Выполнил:

**Маняхин Тимур Александрович**

---

студент 2 курса, ПИЖ-б-о-20-1 группы бакалавриата  
«Программная инженерия»

очной формы обучения

---

Ставрополь, 2021

Северо-Кавказский федеральный университет  
Институт математики и информационных технологий

```

if __name__ == '__main__':
    lst = [0] * 10
    count = 0
    dif = 0
    b = 0
    for i in range(10):
        print("Введите", i+1, "элемент")
        lst[i] = int(input())
        if lst[i] > 0:
            if b == 0 and lst[i] % 11 == 0:
                b = lst[i]
            if lst[i] % 11 == 0:
                dif -= lst[i]
                count += 1
    print("Изначальный список: ", lst, "разность положительных элементов кратных 11:", dif + (b * 2), "количество", count)

```

Рисунок 1.1 – работа программы 1

```

Изначальный список: [11, 22, 33, -11, 1, 2, 3, 4, 5, 6] разность положительных элементов кратных 11: -44 количество 3

```

Рисунок 1.2 – результат работа программы 1

```

if __name__ == '__main__':
    lst = [0] * 10
    count = 0
    d = 1
    c = int(input("Введите c"))
    for i in range(10):
        print("Введите", i + 1, "элемент")
        lst[i] = int(input())
    for i in range(10):
        if lst[i] > c:
            count += 1
    print("количество элементов списка, больших C равно", count)
    b = lst.index(max(lst))+1
    while(b<10):
        d *= lst[b]
        b+=1
    print("произведение элементов списка, расположенных после максимального по модулю равно", d)

```

Рисунок 1.1 – работа программы 1

```

количество элементов списка, больших C равно 4
произведение элементов списка, расположенных после максимального по модулю равно 4032

```

Рисунок 1.2 – результат работа программы 1

## КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Список – это структура данных для хранения объектов различных типов.
2. Для создания списка нужно заключить элементы в квадратные скобки
3. При создании резервируется область, которую можно условно назвать некоторым “контейнером”, в котором хранятся ссылки на другие элементы данных в памяти
4. По списку можно пройти через цикл for
5. оператор сложения ( + ), оператор 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 конец списка
9. Для сортировки списка нужно использовать метод sort
10. Удалить элемент можно, написав его индекс в методе pop
11. Списковое включение это удобный инструмент для создания списков с предварительно заданным количеством элементов. С его помощью можно упростить работу со списками не прибегая к помощи циклов, для работы с такими типами как list, tuple, set, dict
12. С помощью срезов можно получить доступ к от n элемента до m
13. 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.
15. Sorted возвращает новый список, а sort – изменяет список