

Отчёт

Лабораторная работа №3

студента группы ПИ-211

Зубкова Ильи Леонидович

Выполнение _____

Защита _____

Работа со списками

Цель работы: приобрести навыки использования одномерных списков.

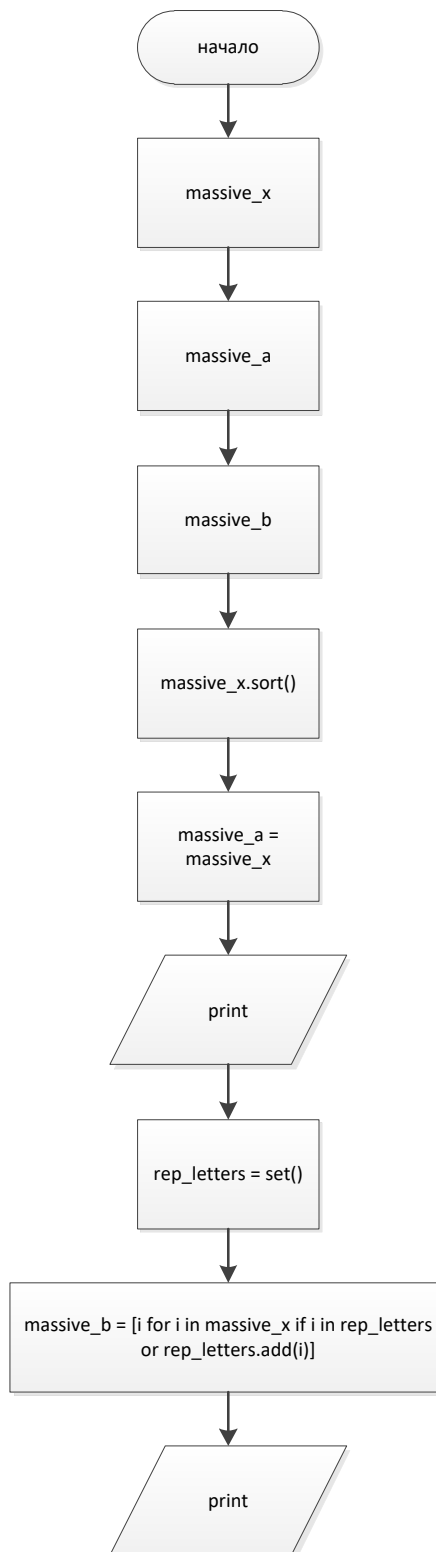
Задания к лабораторной работе

11. Дан массив `x`, состоящий из латинских букв. Вывести его так, чтобы его элементы следовали в алфавитном порядке в массиве `a`, а в массив `b` все повторяющиеся элементы.

1. Разработать алгоритм решения задачи, описать его в виде блок-схемы и составить программу для решения задачи соответствующего варианта.
2. Реализовать два варианта решения задачи: с использованием стандартных методов и операций работы со списками, а также с собственной реализацией этих методов и операций.
3. Набрать текст программы и отладить её в среде разработки PyCharm.
4. Выполнение/Реализация: _____ (блок-схема + алгоритм)
5. Оформить отчет о проделанной работе.

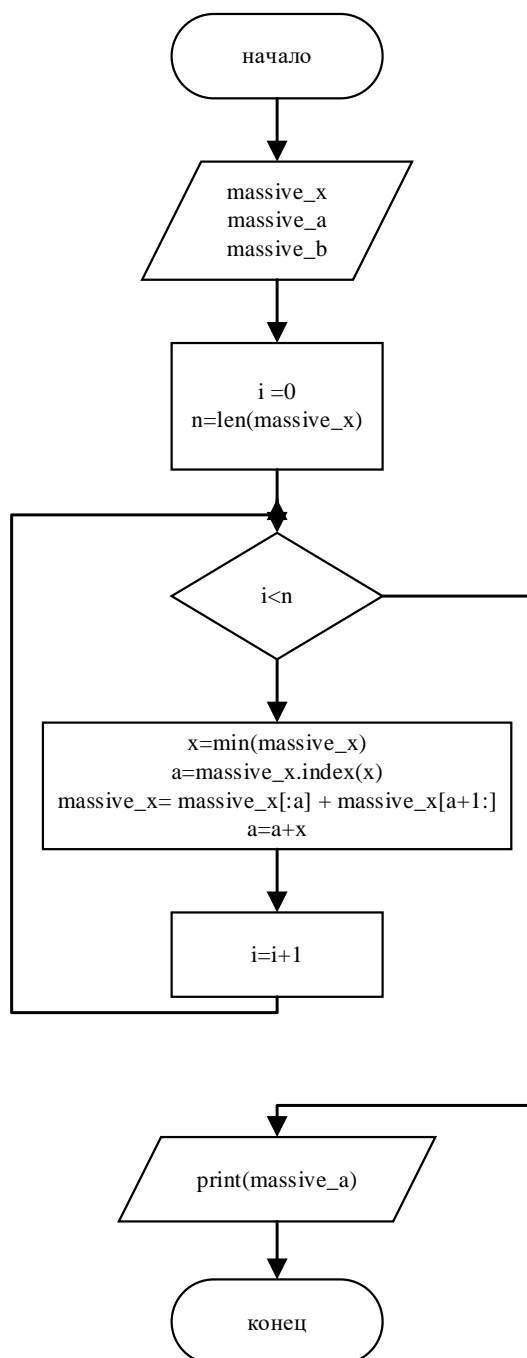
Ход работы:

1. Блок-схема первого способа решения: метод с использованием встроенных функций сортировки.

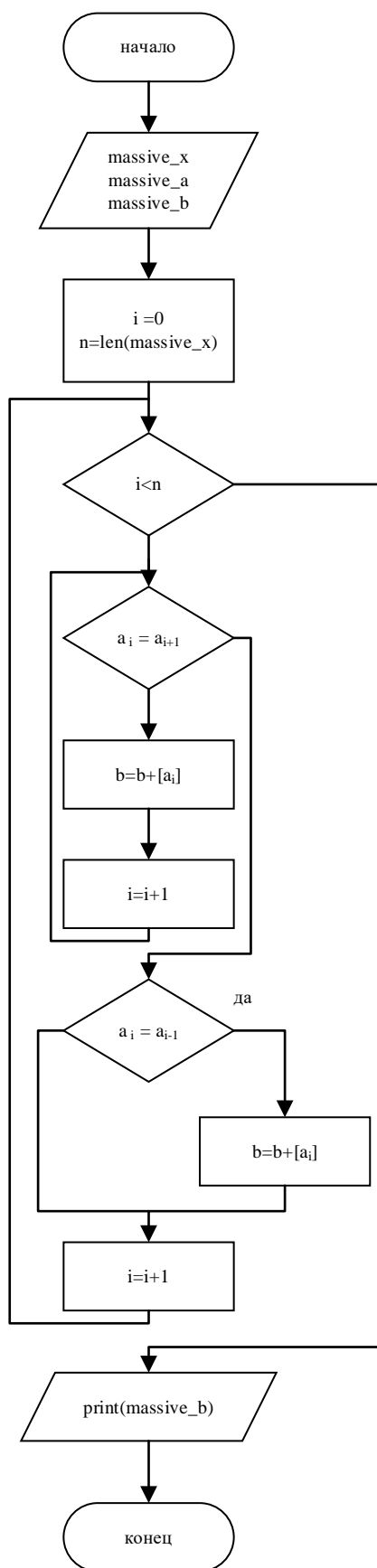


2. Блок-схема первого способа решения: метод собственной реализации.

Сортировка по возрастанию:



Сортировка повторяющихся букв:



3. Код с использованием стандартных методов и операций работы со списками:

```
#первый способ
massive_x = ["a", "k", "h", "b", "q", "k", "b", "z", "m"]

massive_a = [] # в алфавитном порядке
massive_b = [] # повторяющиеся элементы
massive_c = []

massive_x.sort()
massive_a = massive_x
print("Отсортированный первым способом массив в алфавитном порядке : " + str(massive_a))

massive_c = set()
massive_b = [i for i in massive_x if i in massive_c or massive_c.add(i)]
print("Массив с повторяющимися элементами, выведенный первым способом: " + str(massive_b))
```

4. Код с собственной реализацией методов и операций работы со списками:

```
#второй способ
massive_x = ["a", "k", "h", "b", "q", "k", "b", "z", "m"]
massive_a = []
mas_b = []
i=0
n=len(massive_x)
while i < n:
    x=min(massive_x)
    a=massive_x.index(x)
    massive_x = massive_x[:a] + massive_x[a+1:]
    massive_a.append(x)
    i+=1
print("Отсортированный вторым способом массив в алфавитном порядке : " + str(massive_a))

i=0
while i < len(massive_a):
    while i < len(massive_a)-1 and ord(massive_a[i]) == ord(massive_a[i+1]):
        mas_b.append(massive_a[i])
        i = i + 1
    if ord(massive_a[i]) == ord(massive_a[i-1]):
        mas_b.append(massive_a[i])
    i = i + 1
print("Массив с повторяющимися элементами, выведенный вторым способом: " + str(mas_b))
```

5. Набрал и отладил текст программы в среде разработки PyCharm и получил следующие результаты:

```
Отсортированный первым способом массив в алфавитном порядке : ['a', 'b', 'b', 'h', 'k', 'k', 'm', 'q', 'z']
Массив с повторяющимися элементами, выведенный первым способом: ['b', 'k']
-----
Отсортированный вторым способом массив в алфавитном порядке : ['a', 'b', 'b', 'h', 'k', 'k', 'm', 'q', 'z']
Массив с повторяющимися элементами, выведенный вторым способом: ['b', 'b', 'k', 'k']

Process finished with exit code 0
```

Контрольные вопросы:

1. Что такое список?

Список — упорядоченные изменяемые коллекции объектов произвольных типов.

2. Назовите особенности списка

Особенности списка — изменение и сравнение

3. Приведите примеры операций со списком

Операции со списком:

len(list) — длина массива (считывает кол-во элементов)

list.append() — добавление элемента в массив

list.sort() — сортировка элементов

4. Приведите доступные методы списков и примеры их использования

Доступные методы:

list.pop([i]) — удаляет элемент по индексу

list.reverse() — разворачивает список

list.clear() — очищает список