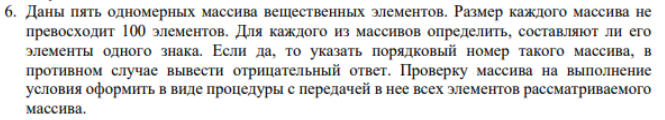
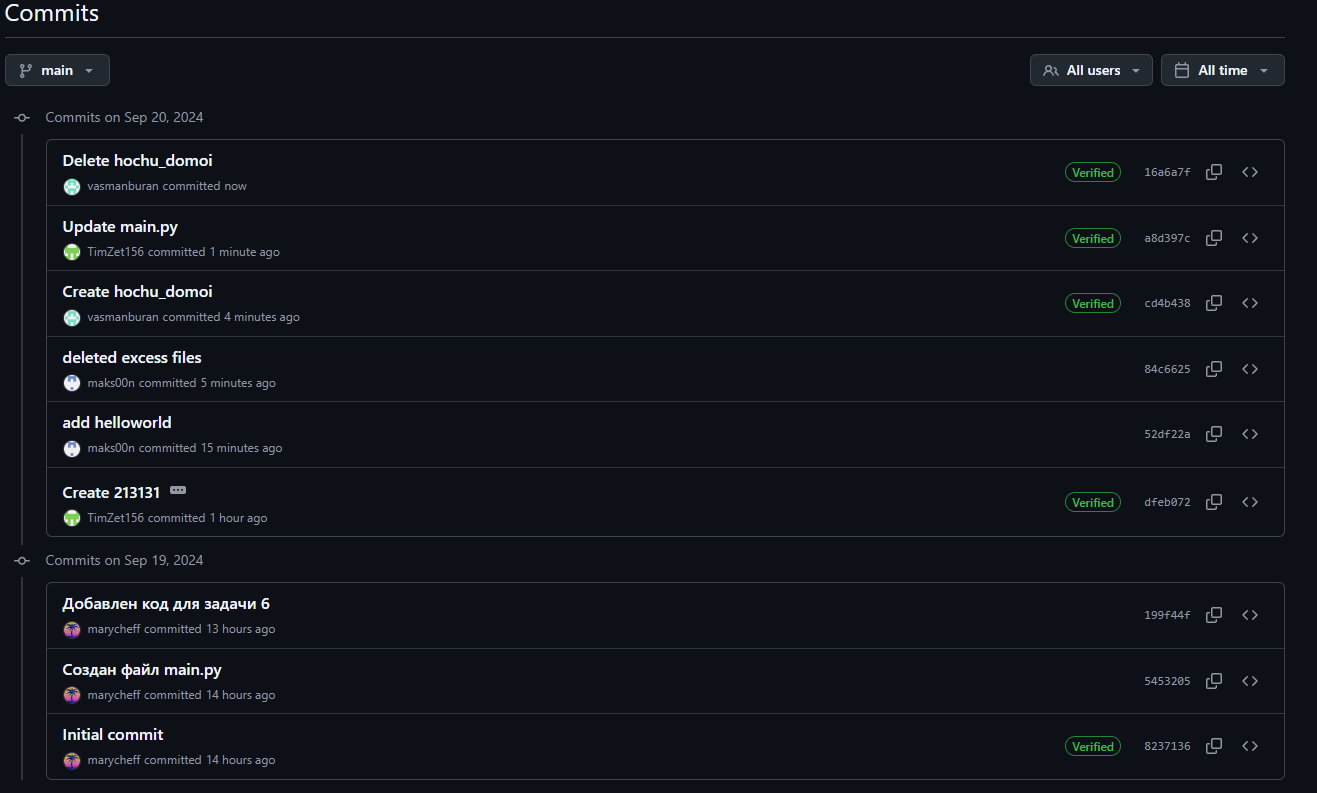
Состав команды: Марычев, Максимов, Тимкин, Воронцов

Д21-191-3

Скриншот задания  


Скриншот работы в репозитории



**Этапы проектирования модулей программы:**

**Этап 1: Определение функциональных требований**

* Определить цель программы: найти массивы с элементами одного знака
* Определить входные данные: список массивов чисел
* Определить ожидаемый результат: вывод массивов с элементами одного знака

**Этап 2: Разбиение на модули (функции)**

* Функция check\_same\_sign: проверяет, имеют ли элементы массива один знак
* Функция process\_arrays: обрабатывает все массивы и выводит результаты

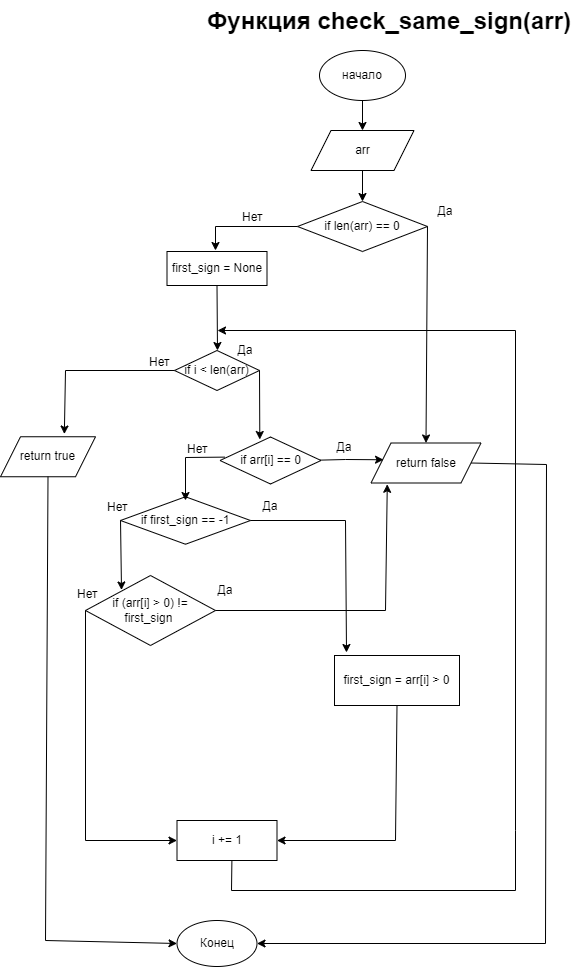
**Этап 3: Реализация модулей**

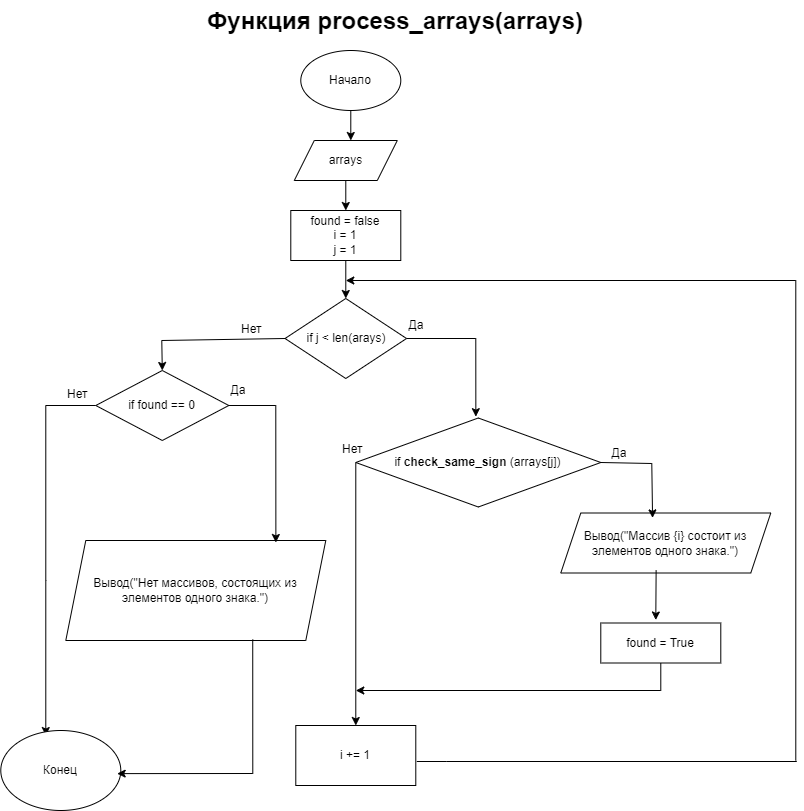
* Реализация модуля check\_same\_sign
* Реализация модуля process\_arrays

**Этап 4: Тестирование**

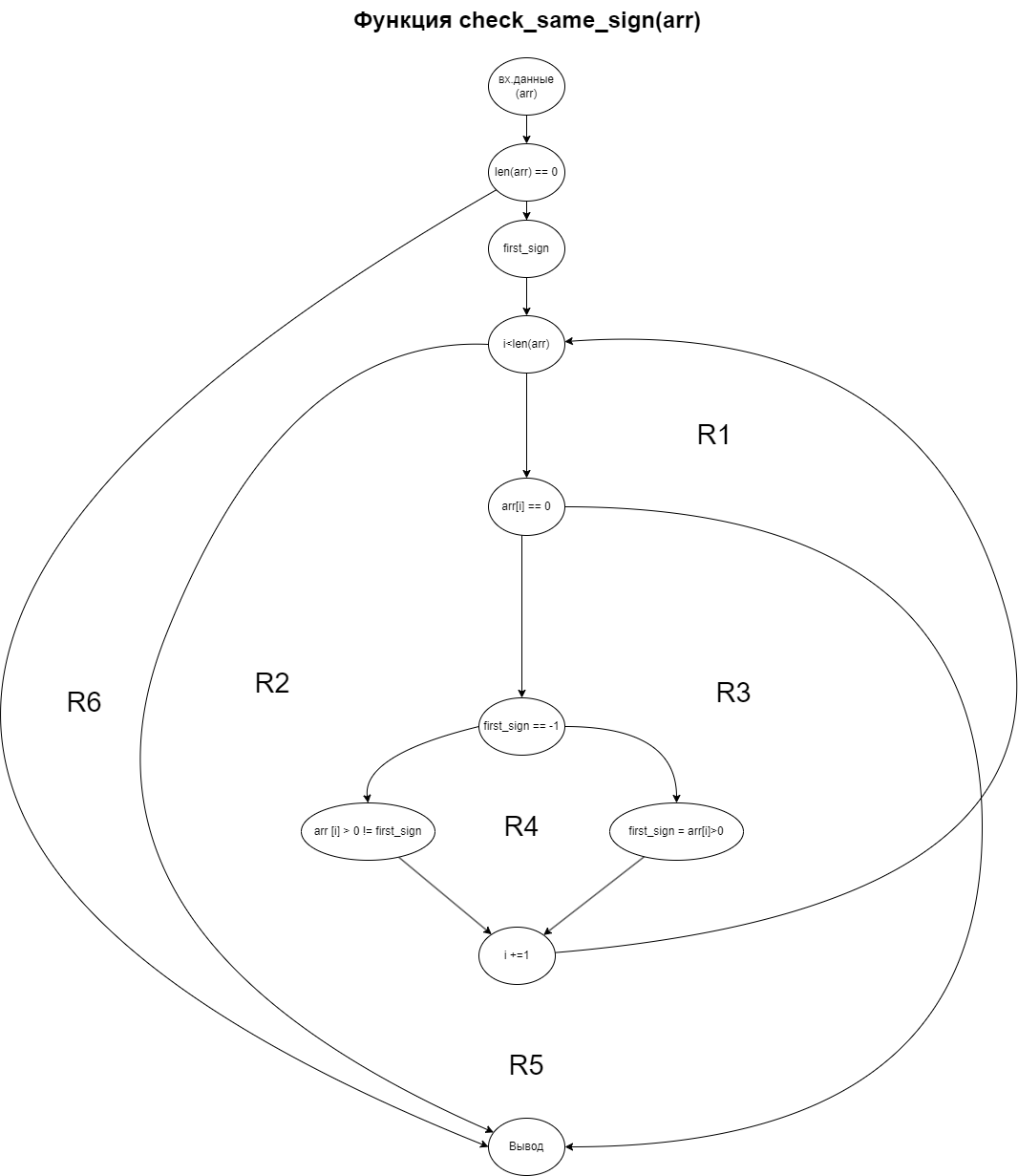
* Тестирование модуля check\_same\_sign на различных входных данных
* Тестирование модуля process\_arrays на различных списках массивов

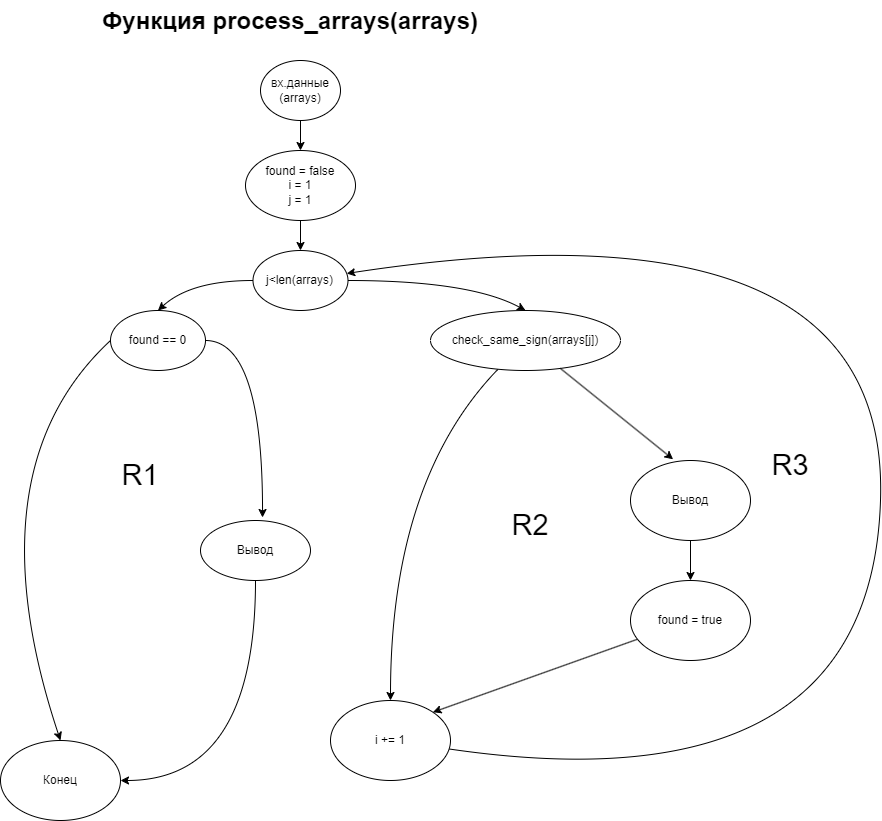
**Скриншоты диаграмм:**





**Скриншоты графов**





Цикломатическая сложность по формуле V(G) = E-N+2

где Е - количество дуг, N - количество узлов потокового графа;

Цикломатическая сложность для графа check\_same\_sign: V(G) = 12 дуг - 9 узлов + 2 = 5

Цикломатическая сложность для графа process\_arrays: V(G) = 12 дуг - 10 узлов + 2 = 4

**Тестирование:**

1. Функция check\_same\_sign

|  |  |  |
| --- | --- | --- |
| Входные данные | Ожидаемый результат | Полученный результат |
| [1.25, 2.12, 3.0, 4.1] | True | True |
| [1,-2,3,-5] | False | False |
| [0, 4.01, 84.56, 12.84] | False | False |
| [-13.5, -12.878, -4.99] | True | True |
| [] | False | False |

1. Функция process\_arrays

|  |  |  |
| --- | --- | --- |
| Входные данные | Ожидаемый результат | Полученный результат |
| [1.2, 3.4, 5.6, 7.8],  [-1.1, -2.2, -3.3],  [0.0, 1.1, -1.1],  [4.4, 5.5, 6.6],  [-7.7, -8.8, -9.9] | Массив 1 состоит из элементов одного знака.  Массив 2 состоит из элементов одного знака.  Массив 4 состоит из элементов одного знака.  Массив 5 состоит из элементов одного знака. | Массив 1 состоит из элементов одного знака.  Массив 2 состоит из элементов одного знака.  Массив 4 состоит из элементов одного знака.  Массив 5 состоит из элементов одного знака. |
| [1.2, 3.4, -5.6, 7.8],  [-1.1, -2.2, -3.3],  [0.0, 1.1, -1.1],  [4.4, -5.5, 6.6],  [-7.7, -8.8, -9.9] | Массив 2 состоит из элементов одного знака.  Массив 5 состоит из элементов одного знака. | Массив 2 состоит из элементов одного знака.  Массив 5 состоит из элементов одного знака. |
| [1.2, 3.4, 5.6, 7.8],  [1.1, 2.2, 3.3],  [-0.5, -1.1, -1.1],  [-4.4, -5.5, -6.6],  [-7.7, -8.8, -9.9] | Массив 1 состоит из элементов одного знака.  Массив 2 состоит из элементов одного знака.  Массив 3 состоит из элементов одного знака.  Массив 4 состоит из элементов одного знака.  Массив 5 состоит из элементов одного знака. | Массив 1 состоит из элементов одного знака.  Массив 2 состоит из элементов одного знака.  Массив 3 состоит из элементов одного знака.  Массив 4 состоит из элементов одного знака.  Массив 5 состоит из элементов одного знака. |
| [1.2, 3.4, -5.6, 7.8],  [-1.1, -2.2, 3.3],  [0.0, 1.1, -1.1],  [-4.4, -5.5, 6.6],  [-7.7, -8.8, 9.9] | Нет массивов, состоящих из элементов одного знака. | Нет массивов, состоящих из элементов одного знака. |

**Код программы на языке Python:**

def check\_same\_sign(arr):

*if* len(arr) == *0*:

*return* *False*

    first\_sign = *None*

*for* x *in* arr:

*if* x == *0*:

*return* *False*

*if* first\_sign == *None*:

            first\_sign = x > *0*

*elif* (x > *0*) != first\_sign:

*return* *False*

*return* *True*

def process\_arrays(arrays):

    found = *False*

    i = *1*

*for* arr *in* arrays:

*if* check\_same\_sign(arr):

            print(f"Массив *{*i*}* состоит из элементов одного знака.")

            found = *True*

        i += *1*

*if* not found:

        print("Нет массивов, состоящих из элементов одного знака.")

def main():

    arrays = [

        [*1.2*, *3.4*, *5.6*, *7.8*],

        [-*1.1*, -*2.2*, -*3.3*],

        [*0.0*, *1.1*, -*1.1*],

        [*4.4*, *5.5*, *6.6*],

        [-*7.7*, -*8.8*, -*9.9*]

    ]

    process\_arrays(arrays)

main()