МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

**«БЕЛГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ
ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ им. В. Г. ШУХОВА»
(БГТУ им. В.Г. Шухова)**

Кафедра программного обеспечения вычислительной техники и автоматизированных систем

Лабораторная работа №5

по дисциплине: Основы программирования
тема: «Использование функций при работе
с двумерными массивами»

Выполнил: ст. группы ПВ-201
Машуров Дмитрий Русланович

Проверил:
Притчин Иван Сергеевич
Брусенцева Валентина
Станиславовна

Белгород 2021 г.

Лабораторная работа №5

«Использование функций при работе с двумерными массивами»

Цель работы: получение навыков работы с функциями и двумерными массивами

Задания для подготовки к работе

1. Изучить способы описания и инициализации многомерных массивов, правила передачи массивов функциям.
2. Разбить задачу соответствующего варианта на подзадачи таким образом, чтобы решение каждой подзадачи описывалось функцией, а основная программа состояла бы из последовательности вызовов функций. Размеры матриц задать константами.
5. Для каждой подзадачи описать спецификацию и блок-схему алгоритма. Спецификация содержит заголовок функции и ее назначение, из которого должен быть понятен смысл каждого параметра.
6. Подобрать наборы тестовых данных.

Задание варианта №17

Дана квадратная матрица. Определить k – количество "особых" элементов матрицы, считая элемент "особым", если он больше суммы остальных элементов своего столбца.

Выполнение работы:

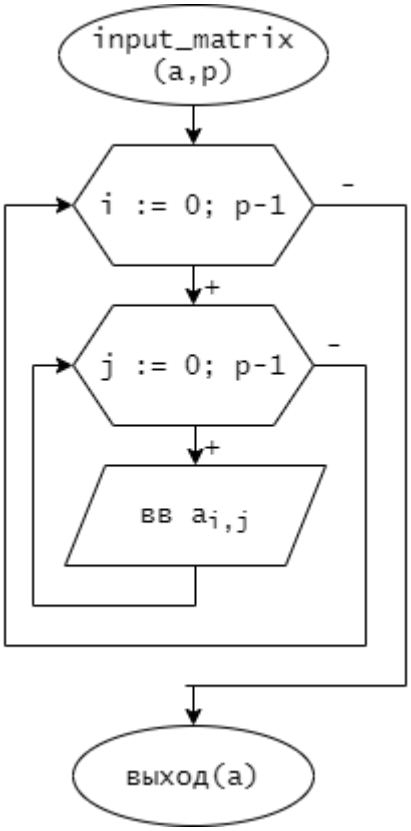
1. **Описание алгоритма и выделение подзадач**
   Исходя из условия задачи, нам нужно находить сумму строки матрицы и сравнивать каждый элемент с суммой строки матрицы, которая не включает в себя данный элемент ($sum - a$, где $a -$ данный элемент, $sum -$ сумма элементов строки)

   Выделим следующие подзадачи:
   1) Нахождение суммы строки
   2) Нахождение количества «особенных» элементов
   3) Ввод матрицы

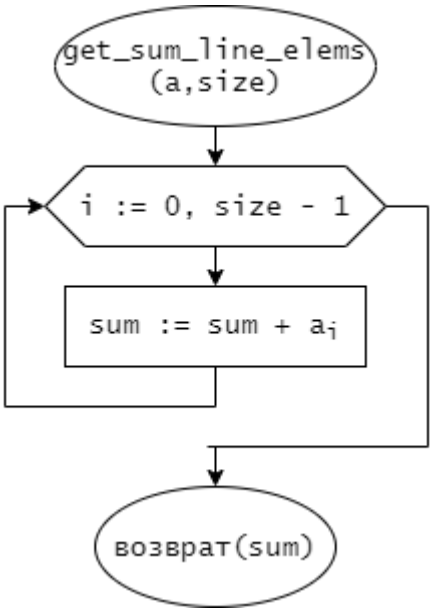
2. **Блок-схема с укрупнёнными блоками**

3. Спецификации функций

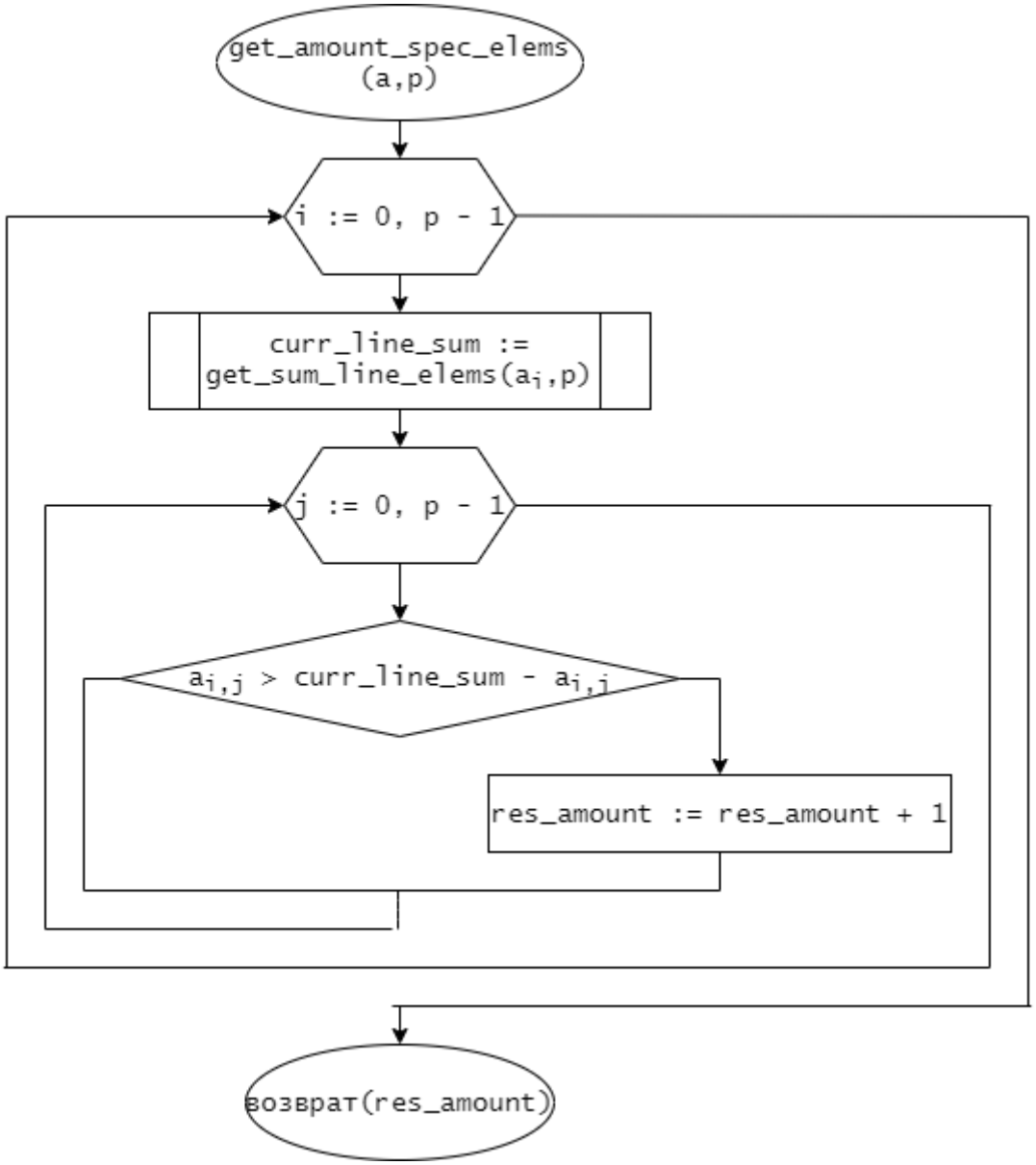
1) Ввод матрицы
   a) Заголовок: `void input_matrix(int a[][MAX], size_t p)`
   b) Назначение: ввод матрицы `a` порядка `p`
   Блок-схема:



2) Нахождение суммы строки
   a) Заголовок: `int get_sum_line_elems(const int a[], size_t size)`
   b) Назначение: возвращает сумму элементов массива `a` размера `size`
   Блок-схема:



3) Нахождение количества «особенных» элементов
    a) Заголовок: `size_t get_amount_spec_elems(const int a[][MAX], size_t p)`
    b) Назначение: возвращает количество элементов, каждый из которых превышает сумму остальных элементов строки, матрицы `a` порядка `p`
    Блок-схема:



4. Тестовые данные

№ п/п	Вход	Выход
1	p = 4 1 2 3 10 3 1 1 0 0 0 0 0 1 1 1 1	2
2	p = 3 1 2 3 3 2 1 2 2 2	0
3	p = 3 -1 2 4 -1 -2 3 -1 -1 -1	5

5. Текст программы

```c
/*
 * Дана квадратная матрица. Определить k – количество "особых"
 * элементов матрицы, считая элемент "особым", если он больше
 * суммы остальных элементов своего столбца
 */

#include <stdio.h>
#include <stdlib.h>
#define MAX 100

/* ввод матрицы a порядка p */
void input_matrix(int a[][MAX], size_t p) {
    for (size_t i = 0; i < p; i++) {
        for (size_t j = 0; j < p; j++) {
            scanf("%d", &a[i][j]);
        }
    }
}

/* возвращает сумму элементов массива a размера size */
int get_sum_line_elems(const int a[], size_t size) {
    int sum = 0;

    for (size_t i = 0; i < size; i++) {
        sum += a[i];
    }

    return sum;
}
/*
 * возвращает количество элементов, каждый из которых
 * превышает сумму остальных элементов строки, матрицы a порядка p
 */
size_t get_amount_spec_elems(const int a[][MAX], size_t p) {
    size_t res_amount = 0;
    int curr_line_sum;
```

```
    for (size_t i = 0; i < p; i++) {
        curr_line_sum = get_sum_line_elems(a[i],p);
        for (size_t j = 0; j < p; j++) {
            if (a[i][j] > curr_line_sum - a[i][j]) {
                res_amount++;
            }
        }
    }

    return res_amount;
}

int main() {
    size_t p;
    printf("Input matrix order\n");
    scanf("%u", &p);

    printf("Input matrix elements\n");
    int a[MAX][MAX];
    input_matrix(a, p);
    size_t res = get_amount_spec_elems(a, p);

    printf("%u", res);
}
```

6. Результаты работы

Пример №1

```
Input matrix order
4
Input matrix elements
1 2 3 10
3 1 1 0
0 0 0 0
1 1 1 1
2
Process finished with exit code 0
```

Пример №2

```
Input matrix order
3
Input matrix elements
1 2 3
3 2 1
2 2 2
0
Process finished with exit code 0
```

Пример №3

```
Input matrix order
3
Input matrix elements
-1 2 4
-1 -2 3
-1 -1 -1
5
Process finished with exit code 0
```

7. **Анализ ошибок**
    - Изначально инициализировал матрицу динамически – невнимательное прочтение условия задания