

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ  
Федеральное государственное бюджетное образовательное учреждение высшего образования  
МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ТЕХНОЛОГИЙ И УПРАВЛЕНИЯ ИМ. К.Г.  
РАЗУМОВСКОГО (ПЕРВЫЙ КАЗАЧИЙ УНИВЕРСИТЕТ)  
(ФГБОУ ВО МГУТУ им. К.Г.РАЗУМОВСКОГО (ПКУ))



---

Факультет цифровых технологий

Кафедра: Информационные системы и цифровые технологии  
Направление подготовки – 09.03.01 «Проектирование и разработка  
программного обеспечения»

ОТЧЕТ ПО ДИСЦИПЛИНЕ

«Основы алгоритмизации и программирования»

Лабораторная работа № 5.

Тема: «Массивы и указатели»

Преподаватель	Таченков О.С. <small>(ученая степень, ученое звание, фамилия, инициалы)</small>
Студент	1 090301-РПРОо-24/1 Асылбек уулу Бакыт <small>курс группа (фамилия, имя, отчество)</small>

Москва, 2024 г.

## СОДЕРЖАНИЕ

СЛОВЕСНАЯ ПОСТАНОВКА ЗАДАЧИ .....	3
МАТЕМАТИЧЕСКАЯ ПОСТАНОВКА ЗАДАЧИ.....	4
КОНТРОЛЬНЫЙ ПРИМЕР .....	5
БЛОК-СХЕМА.....	6
ЛИСТИНГ ПРОГРАММЫ.....	10
РЕЗУЛЬТАТЫ .....	12

## СЛОВЕСНАЯ ПОСТАНОВКА ЗАДАЧИ

По варианту задания на массивы из лабораторной работы № 4 написать программу на одномерные массивы, используя пользовательские функции.

Обязательно реализовать функции для:

- ввода (создания) массива;
- вывода массива;

Все решение задачи по условию также реализовать в виде функций.

Лабораторная работа № 4: Даны два массива. Найти отклонение медианы от среднего арифметического для третьего массива, элементы которого – частное от деления соответствующих элементов 1-го массива на 2-ой (исключайте деление на ноль). Число элементов в массиве не более 13. Формат числа ццц.

## МАТЕМАТИЧЕСКАЯ ПОСТАНОВКА ЗАДАЧИ

Дано:

$A[a_1, a_2, a_3 \dots a_n], B[b_1, b_2, b_3 \dots b_n]$ , где  $n$  – количество элементов в массивах,  $n \leq 13$

Найти:

Массив  $C = \left[ \frac{a_1}{b_1}, \frac{a_i}{b_i} \dots \frac{a_n}{b_n} \right]$ , где  $b_i \neq 0$ . Если  $b_i = 0$ , то элемент  $c_i = 0$

$\bar{C} = \frac{\sum_{i=1}^n c_i}{n}$  – среднее арифметическое

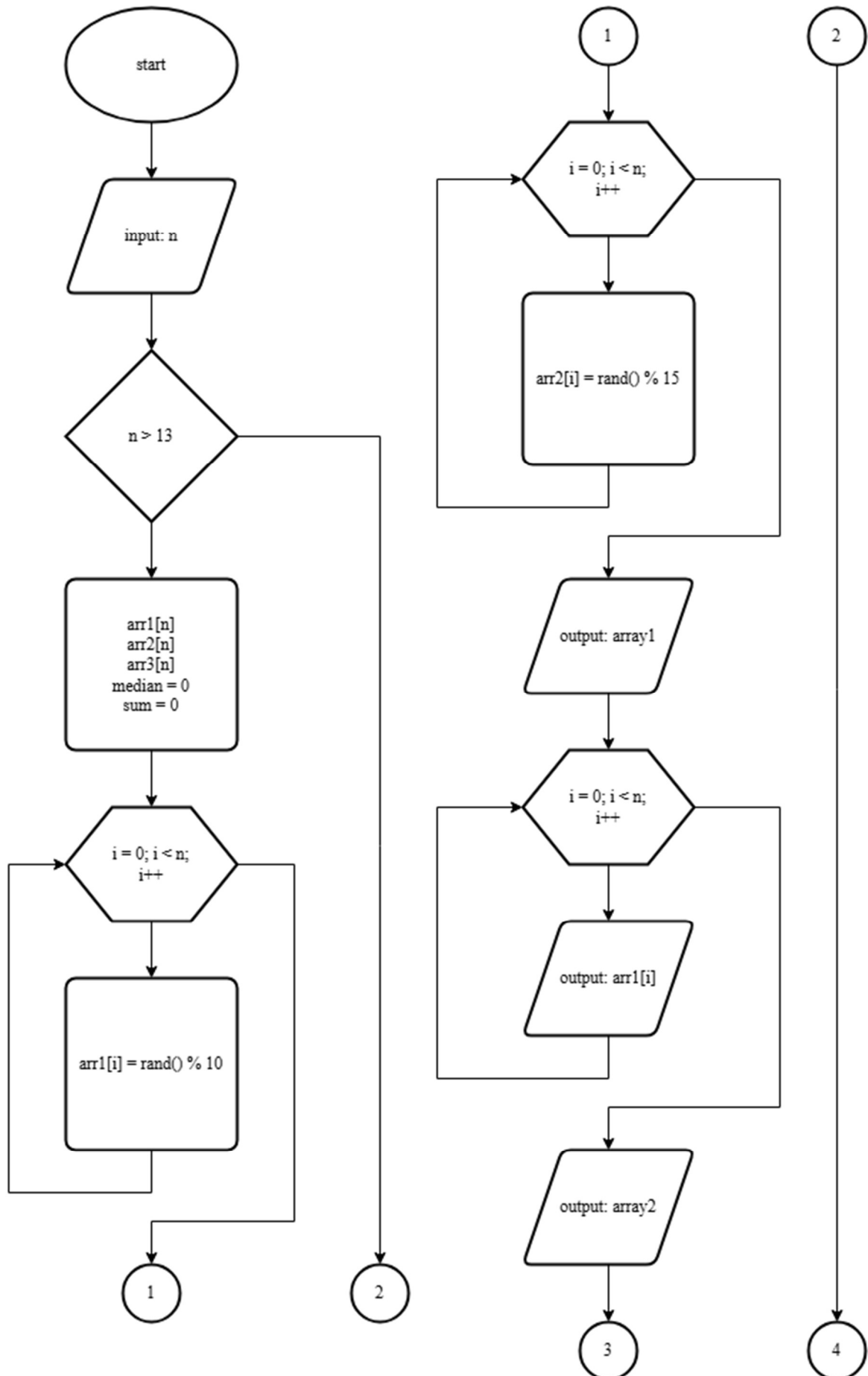
$M = \begin{cases} \frac{\frac{c_{n-1}}{2} + \frac{c_n}{2}}{2}, & \text{если } n \text{ четный} \\ \frac{c_n}{2}, & \text{если } n \text{ нечетный} \end{cases}$  – медиана

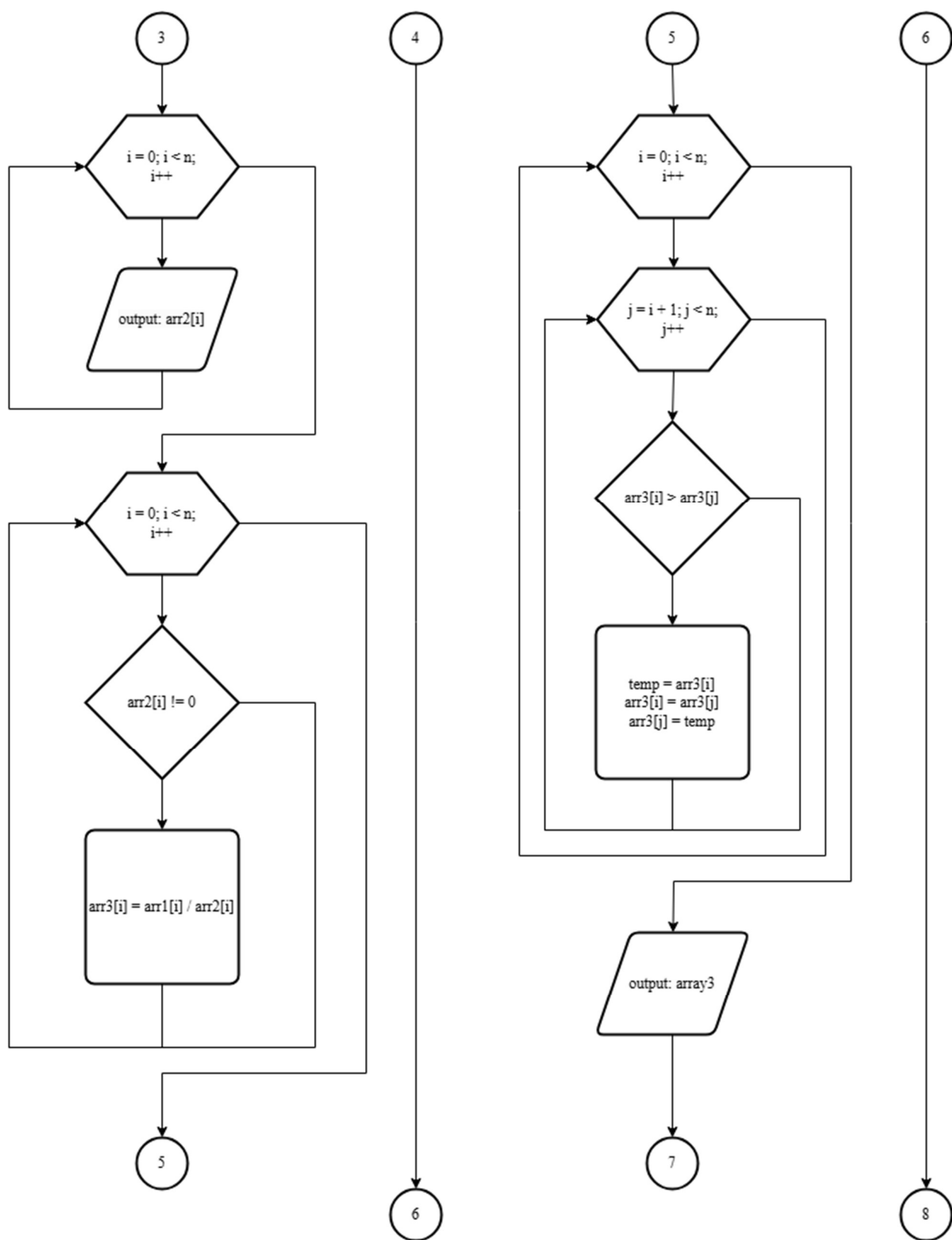
$D = M - \bar{C}$  – отклонение медианы от среднего арифметического

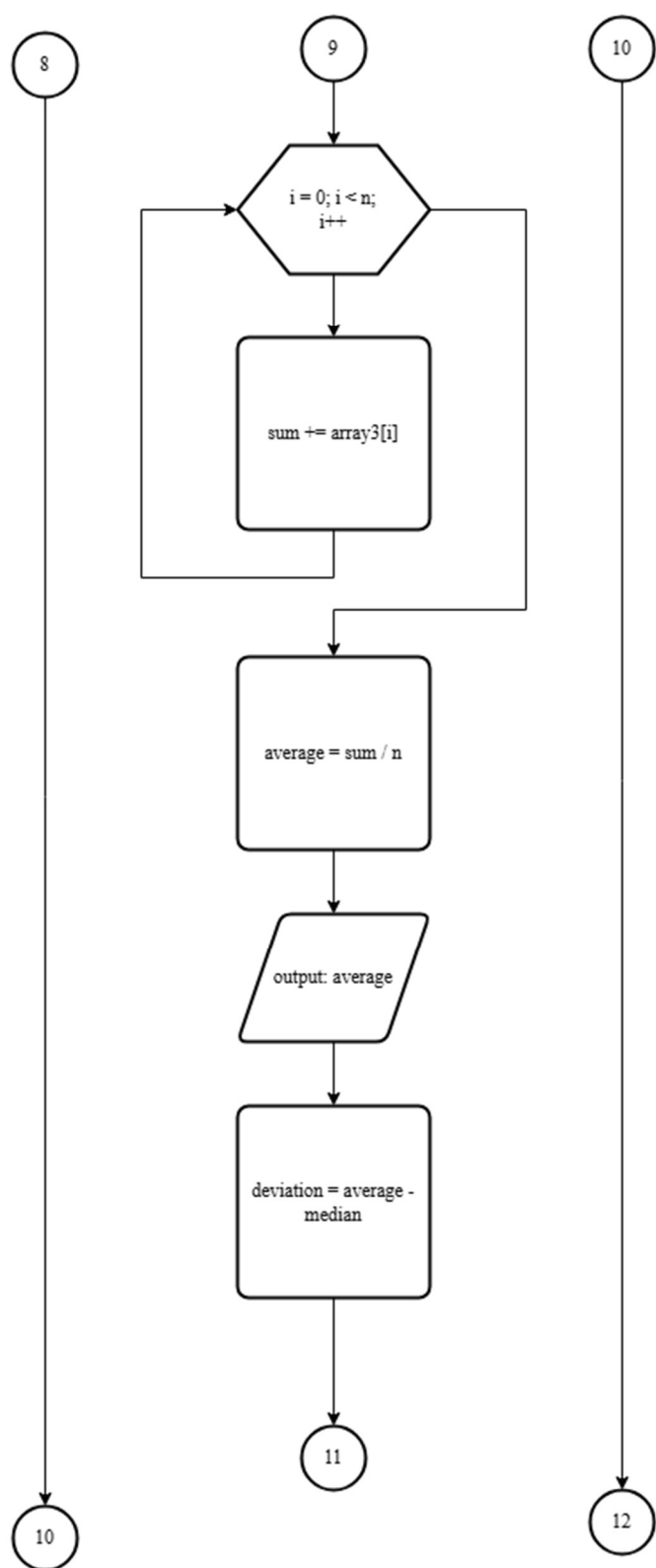
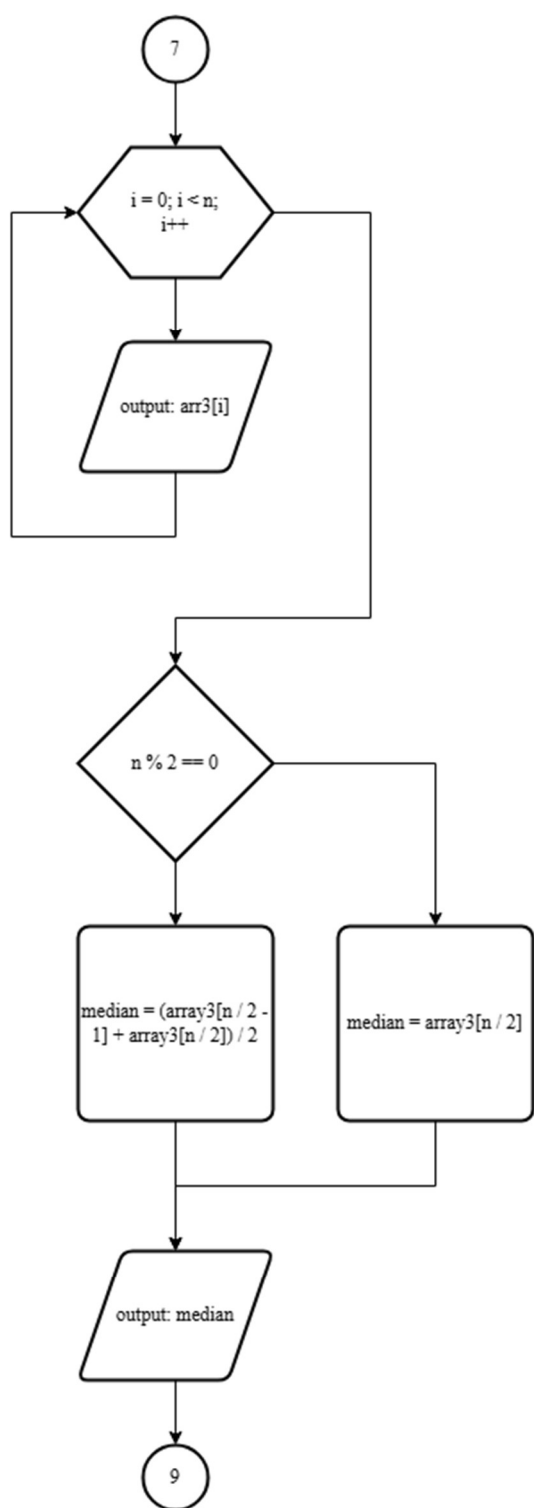
### КОНТРОЛЬНЫЙ ПРИМЕР

Номер по порядку	Заданные массивы		Результирующий массив
	1	2	
1	8, 6, 4, 1, 1, 7, 2, 3, 7	8, 9, 3, 3, 2, 8, 9, 3, 5	1, 0.6, 1.3, 0.3, 0.5, 0.875, 0.2, 1, 1,4
2	8, 6, 4, 1, 1, 7, 2, 3	7, 3, 9, 3, 13, 7, 8, 14	1.1, 2, 0.4, 0.3, 0.07, 1, 0.25, 0.21

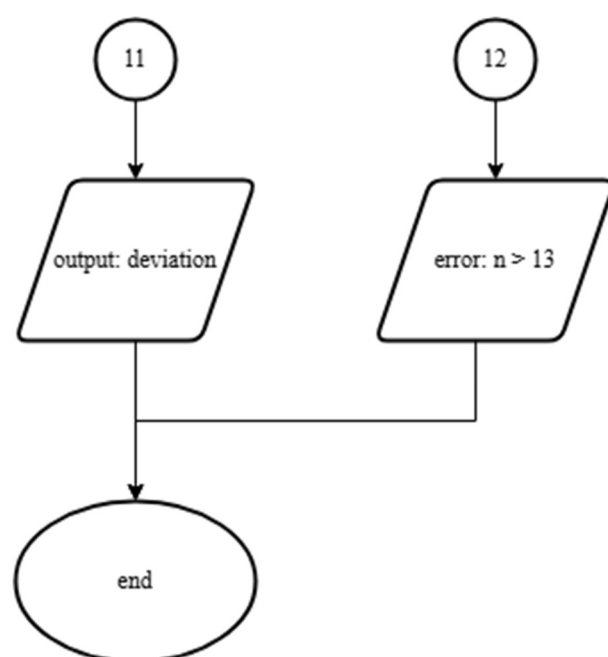
## БЛОК-СХЕМА











## ЛИСТИНГ ПРОГРАММЫ

```
int* createIntArray(const int n) {
    int* array = calloc(n, sizeof(int));
    if (array == NULL) {
        free(array);
        return nullptr;
    }
    for (int i = 0; i < n; i++) array[i] = rand() % 10;
    return array;
}

float* createFloatArray(int n, const int* array1, const int* array2) {
    float* array = calloc(n, sizeof(float));
    if (array == NULL) {
        free(array);
        return nullptr;
    }
    for (int i = 0; i < n; i++) {
        if (array2[i] != 0) array[i] = (float)array1[i] / (float)array2[i];
    }
    return array;
}

void outputIntArray(const int n, int array[]) {
    for (int i = 0; i < n; i++) printf("%d ", array[i]);
}

void outputFloatArray(const int n, float array[]) {
    for (int i = 0; i < n; i++) printf("%.3f ", array[i]);
}

float sumOfArrayElements(const float array[], const int n) {
    float sum = 0;
    for (int i = 0; i < n; i++) sum += array[i];
    return sum;
}

float calculateMedian(float array[], const int n) {
    // Сортировка массива для нахождения медианы
    for (int i = 0; i < n - 1; i++) {
        for (int j = i + 1; j < n; j++) {
            if (array[i] > array[j]) {
                float temp = array[i];
                array[i] = array[j];
                array[j] = temp;
            }
        }
    }
    if (n % 2 == 0) return (array[n / 2 - 1] + array[n / 2]) / 2;
    return array[n / 2];
}

int fifthPW3Task() {
    srand(3u);
    int n = 0;
    printf("input n:");
    scanf_s("%d", &n);
    if (n > 13) return printf("error: n > 13");
    int *array1 = createIntArray(n), *array2 = createIntArray(n);
    if (array1 == NULL || array2 == NULL) {
        free(array1);
        free(array2);
        return printf("error: array1 and array2 are NULL");
    }
    printf("\narray1: ");
    outputIntArray(n, array1);
}
```

```

printf("\narray2: ");
outputIntArray(n, array2);

float* array3 = createFloatArray(n, array1, array2);
if (array3 == NULL) {
    free(array1);
    free(array2);
    free(array3);
    return printf("error: array3 is NULL");
}
printf("\narray3: ");
outputFloatArray(n, array3);

const float median = calculateMedian(array3, n);
printf("\nmedian: %.3f", median);
const float sum = sumOfArrayElements(array3, n);
const float average = sum / (float)n;
printf("\naverage: %.3f", average);

const float deviation = median - average;
printf("\ndeivation: %.3f", deviation);
free(array1);
free(array2);
free(array3);
return 0;
}

```

## РЕЗУЛЬТАТЫ

input n:9

array1: 8 6 4 1 1 7 2 3 7

array2: 8 9 3 3 2 8 9 3 5

array3: 1.000 0.667 1.333 0.333 0.500 0.875 0.222 1.000 1.400

median: 0.875

average: 0.815

deviation: 0.060

Process finished with exit code 0

input n:10

array1: 8 6 4 1 1 7 2 3 7 8

array2: 9 3 3 2 8 9 3 5 9 8

array3: 0.889 2.000 1.333 0.500 0.125 0.778 0.667 0.600 0.778 1.000

median: 0.778

average: 0.867

deviation: -0.089

Process finished with exit code 0

input n:14

error: n > 13

Process finished with exit code 0

|