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. |
| (ученая степень, ученое звание, фамилия, инициалы) |
| Студент | 1 090301-РПРОо-24/1 Асылбек уулу Бакыт |
| курс группа (фамилия, имя, отчество) |

Москва, 2024 г.

**СОДЕРЖАНИЕ**

[СЛОВЕСНАЯ ПОСТАНОВКА ЗАДАЧИ 3](#_Toc181636402)

[МАТЕМАТИЧЕСКАЯ ПОСТАНОВКА ЗАДАЧИ 4](#_Toc181636403)

[КОНТРОЛЬНЫЙ ПРИМЕР 5](#_Toc181636404)

[БЛОК-СХЕМА 6](#_Toc181636405)

[ЛИСТИНГ ПРОГРАММЫ 10](#_Toc181636406)

[РЕЗУЛЬТАТЫ 12](#_Toc181636407)

# 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:

, где n – количество элементов в массивах,

Найти:

Массив , где . Если , то элемент

– среднее арифметическое

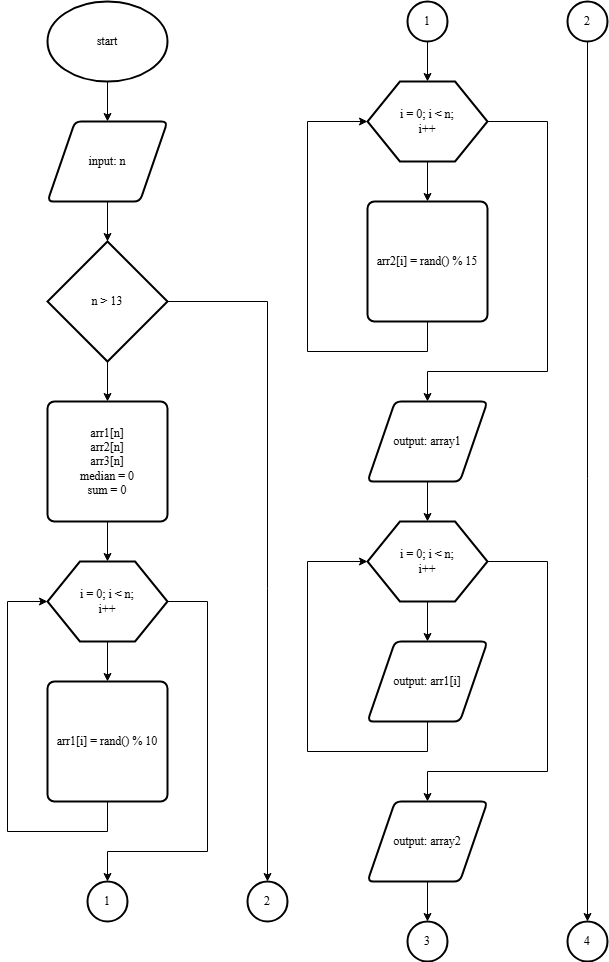
– медиана

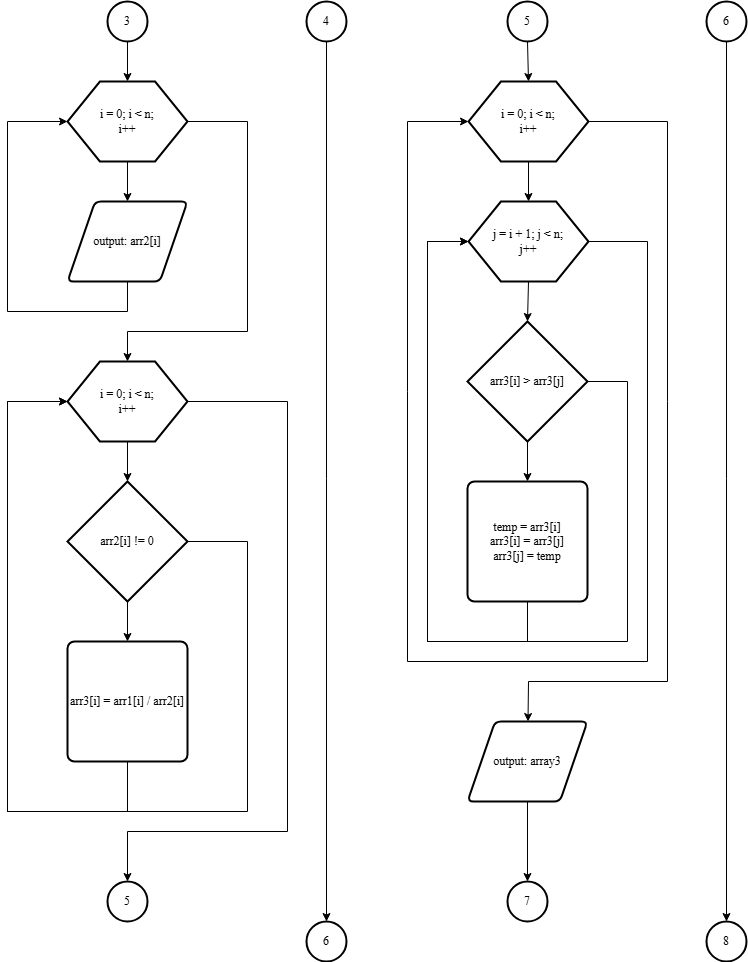
*–* отклонение медианы от среднего арифметического

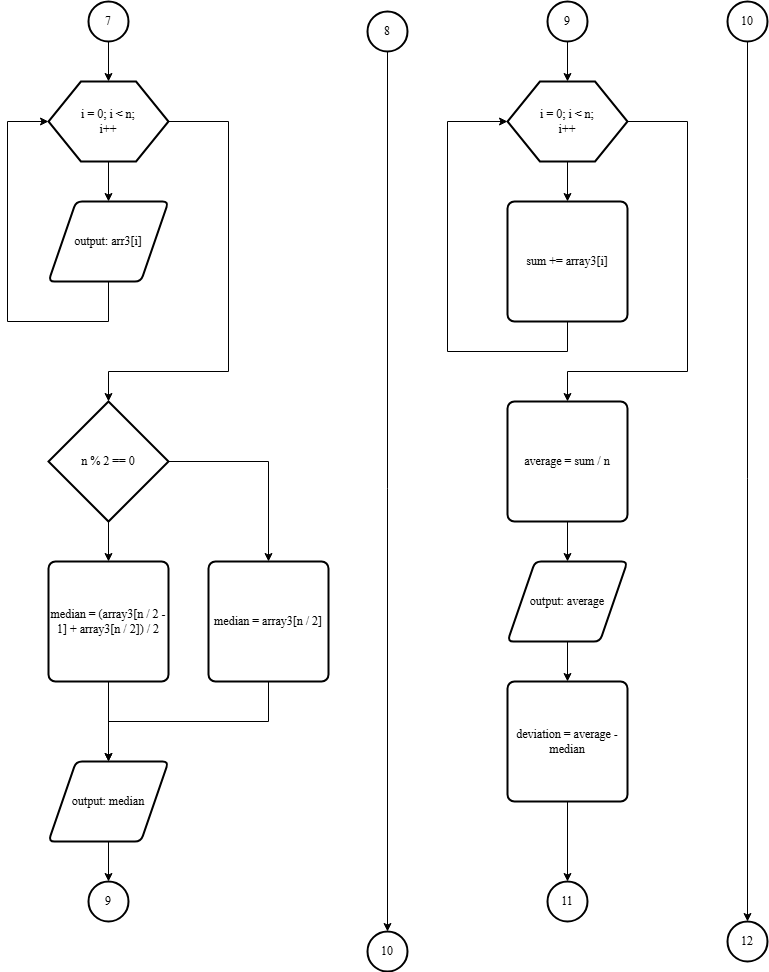
# КОНТРОЛЬНЫЙ ПРИМЕР

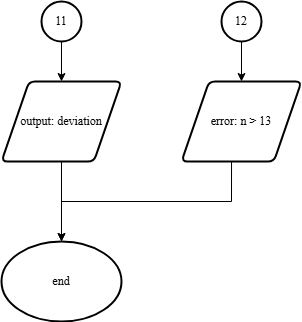
|  |  |  |  |
| --- | --- | --- | --- |
| Номер по порядку | Заданные массивы | | Результирующий массив |
| 1 | 2 |
| 1 | 8, 6, 4, 1, 1, 7, 2, 3, 7 | 8, 9, 3, 3, 2, 8, 9, 3, 5 | 1, 0.6, 1.3, 0.3, 0.5, 0.875, 0.2, 1, 1,4 |
| 2 | 8, 6, 4, 1, 1, 7, 2, 3 | 7, 3, 9, 3, 13, 7, 8, 14 | 1.1, 2, 0.4, 0.3, 0.07, 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("\ndeviation: %.3f", deviation);  
 free(array1);  
 free(array2);  
 free(array3);  
 return 0;  
}

# РЕЗУЛЬТАТЫ

