

Лабораторная работа №5. Многомерные массивы. Функции

Особенности работы с двумерными массивами

Часто для хранения групп данных одного типа удобнее использовать не одномерные, а двумерные массивы. Их часто называют матрицами. К любому элементу двумерного массива можно обратиться, используя имя массива, номер строки и номер столбца, в которых находится элемент.

Двумерный массив представляется в C++ как массив, состоящий из массивов. Массив хранится по строкам в непрерывной области памяти. Строки массива ничем не отделены одна от другой, т.е. прямоугольной матрицей двумерный массив является только в нашем воображении. Первый индекс всегда представляет собой номер строки, второй – номер столбца. Каждый индекс может изменяться в заданном при объявлении массива диапазоне.

В массивах, определенных с помощью операторов описания, обе размерности должны быть константами или константными выражениями. Рекомендуется задавать размерности через именованные константы.

Для работы с двумерными массивами для большинства задач необходимо использовать два вложенных цикла, иногда нужно больше, в редких случаях хватает одного. Как и в случае одномерных массивов, наиболее употребим цикл **for**.

Во избежание ошибок рекомендуется записывать операторы инициализации накапливаемых в цикле величин непосредственно перед циклом, в котором они вычисляются.

Когда в программе требуется ввести несколько величин, для ее отладки удобно использовать текстовые файлы. Как правило, программу не удастся написать сразу без ошибок, а многократный ввод одних и тех же данных сильно замедляет процесс отладки. Кроме того, при подготовке данных в файле до выполнения программы можно спокойно продумать тестовые примеры для исчерпывающей проверки правильности программы.

Использование динамических двумерных массивов

Способ позволяет создавать массив, в котором все размерности переменные. Для этого используется цикл:

```
int n, m;  
cin>>n>>m;  
int **a = new int *[n];  
for (int i = 0; i < n; i++) a[i] = new int [m];
```

Освобождение памяти из-под массива с любым количеством измерений выполняется с помощью операции `delete[]`.

Особенности передачи массивов в качестве параметров в функции

Массивы всегда передаются в функцию по адресу. В функцию передается указатель на первый элемент массива, при этом информация о количестве элементов в массиве теряется. Поэтому следует передавать обе размерности как отдельные параметры. Прототип функции для работы с массивами выглядят примерно так:

```
тип_результата имя_функции(int **a, const int n, const int m);
```

Задания для всех

Все задания выполняются с использованием функций, количество таких функций для разных задач может различаться. Если на входные данные и результаты накладываются какие-то ограничения, отразить это в программе.

Все задачи лабораторной работы (в том числе индивидуальные) необходимо **выполнять в одном проекте**, который может содержать несколько программных модулей. Каждый файл с кодом должен быть снабжен шапкой вида:

```
///*****  
///      Лабораторная работа №5  
///      Многомерные массивы  
///      Выполнил(а) студент(ка) группы 123456  
///      Фамилия Имя Отчество  
///      Вариант №XX  
///*****
```

За каждую задачу из этого раздела можно получить не более одного балла (если не сказано иначе). Номер варианта можно узнать у преподавателя. Работы, выполненные по чужому варианту, не принимаются. Номер варианта остается постоянным для всех лабораторных работ в семестре. Все задания выполняются с использованием функций.

Не забывайте, что элементы в массиве могут быть как положительными, так и отрицательными. Если в задаче сказано, что массив вещественный, то он и должен быть вещественным. В противном случае будет снижена оценка. Для решения задач желательно, но не обязательно использовать динамические массивы.

Решите следующие задачи:

1. Написать программу, организующую заполнение массива размерности $n \times m$ элементов случайными целыми значениями (n, m – именованные константы) и корректный вывод массива. Необходимо написать отдельные функции для ввода и для вывода.
2. (2 балла) Выполнить предыдущую задачу для вещественного двумерного массива. Функции из задач 1 и 2 удобно использовать для решения всех последующих задач.
3. (2 балла) Написать программу, вычисляющую:
 - Вариант 1: для вещественного массива 3×8 среднее арифметическое элементов.
 - Вариант 2: для целочисленного массива 4×6 сумму его элементов.
 - Вариант 3: для целочисленного массива 9×2 количество отрицательных элементов.
 - Вариант 4: для вещественного массива 7×3 сумму положительных элементов.
 - Вариант 5: для целочисленного массива 6×4 среднее арифметическое положительных элементов.
 - Вариант 6: для целочисленного массива 7×4 среднее арифметическое элементов.
 - Вариант 7: для вещественного массива 4×5 среднее арифметическое отрицательных элементов.
 - Вариант 8: для вещественного массива 6×2 среднее арифметическое положительных элементов.
 - Вариант 9: для целочисленного массива 4×8 сумму положительных элементов.
 - Вариант 10: для целочисленного массива 2×10 количество нулевых элементов.
 - Вариант 11: для вещественного массива 3×7 количество отрицательных элементов.
 - Вариант 12: для вещественного массива 2×9 сумму его элементов.
 - Вариант 13: для целочисленного массива 5×6 среднее арифметическое отрицательных элементов.
 - Вариант 14: для вещественного массива 4×8 сумму элементов, больших максимального отрицательного.

Вариант 15: для целочисленного массива 5×7 количество вхождений минимального элемента массива.

Вариант 16: для вещественного массива 6×5 произведение элементов, меньших минимального положительного.

Вариант 17: для целочисленного массива 7×5 количество элементов, меньших максимального положительного.

Вариант 18: для вещественного массива 8×4 произведение элементов, больших минимального положительного.

Вариант 19: для целочисленного массива 9×4 количество вхождений максимального элемента массива.

Вариант 20: для вещественного массива 5×9 сумму элементов, меньших максимального отрицательного.

4. Написать программу, реализующую умножение матрицы на вектор (оба вещественны).
5. (2 балла) Написать программу, реализующую умножение прямоугольных целочисленных матриц.