

Лабораторная работа №4. Одномерные массивы. Функции

Особенности работы с одномерными массивами

Массивы используют для хранения групп данных одного типа. Величины, хранящиеся в массиве, называют его элементами. К любому элементу массива можно обратиться, используя имя массива и номер элемента в этом массиве.

Элементы массивов нумеруются с нуля, поэтому максимальный номер элемента всегда на единицу меньше размерности. Автоматический контроль выхода индекса за границы массива не производится, программист должен следить за этим самостоятельно. Размерность нединамического массива может быть только константой или константным выражением. Рекомендуется задавать размерность с помощью именованной константы.

При описании массив можно инициализировать, задав начальные значения его элементов. Если инициализирующих значений меньше, чем элементов в массиве, остаток массива обнуляется, если больше – лишние значения не используются. Можно инициализировать сразу весь массив нулями.

Для работы с массивами удобнее всего использовать цикл **for**. Помните, что массив нельзя вывести на печать или ввести сразу, нужно делать это поэлементно. То же самое верно для всех алгоритмов работы с массивами. Не забывайте, что в C++ существует возможность для цикла **for** задавать шаг счетчика цикла (а не только увеличивать или уменьшать его на единицу). Для многих задач это оказывается весьма полезным.

При нахождении различных промежуточных параметров массивов рекомендуется выводить их на печать во время отладки программы. Когда программа будет готова, лишний вывод нужно убрать.

Указатель – это переменная, в которой хранится адрес области памяти. Имя массива является указателем на его нулевой элемент.

Использование динамических массивов в C++

Если до начала работы неизвестно, сколько в массиве элементов, следует использовать динамические массивы. В этом случае размерность массива задается при помощи переменной. Память для хранения массива выделяется в процессе выполнения программы при помощи операции `new[]`. Например:

```
int n = 10;  
int *a = new int[n];
```

При использовании динамических массивов не следует забывать об освобождении памяти после того, как массив перестает быть нужным. Освобождение памяти выполняется с помощью операции `delete[]`:

```
delete [] a;
```

Обнуления динамической памяти при ее выделении не происходит. Инициализировать при объявлении динамический массив нельзя.

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 размерность массива как отдельный параметр. Прототип функции для работы с одномерными массивами выглядят примерно так:

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

Задания

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

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

```
///*****  
///      Лабораторная работа №4  
///      Одномерные массивы  
///      Выполнил(а) студент(ка) группы 151214  
///      Фамилия Имя Отчество  
///      Вариант №XX  
///*****
```

За каждую задачу 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. **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.

Решите следующие задачи:

1. Написать программу, содержащую функцию заполнения целого массива из n элементов случайными значениями (n – целая переменная) и функцию, осуществляющую корректный вывод массива.
 2. (2 балла) Выполнить предыдущую задачу для вещественного массива.
- Функции из задач 1 и 2 понадобятся для решения всех последующих задач.
3. (2 балла) В одномерном массиве, состоящем из n вещественных элементов, вычислить:
Вариант 1: сумму отрицательных элементов массива.
Вариант 2: произведение положительных элементов массива.
Вариант 3: количество элементов массива, меньших C (вводится с клавиатуры).
Вариант 4: сумму положительных элементов массива.
Вариант 5: количество отрицательных элементов массива.
Вариант 6: количество положительных элементов массива.
Вариант 7: сумму элементов массива, больших C (вводится с клавиатуры).
Вариант 8: произведение отрицательных элементов массива.
Вариант 9: произведение элементов массива, меньших C (вводится с клавиатуры).
Вариант 10: сумму элементов массива, меньших C (вводится с клавиатуры).
Вариант 11: сумму модулей элементов массива с четными номерами.
Вариант 12: произведение модулей элементов массива с четными номерами.
Вариант 13: сумму модулей элементов массива с нечетными номерами.
Вариант 14: произведение элементов массива с четными номерами.
Вариант 15: сумму элементов массива с нечетными номерами.
Вариант 16: произведение элементов массива с нечетными номерами.
Вариант 17: сумму элементов массива с четными номерами.
Вариант 18: произведение модулей элементов массива, меньших C (вводится с клавиатуры).
Вариант 19: количество отличающихся от нуля менее чем на ϵ элементов массива.

- Вариант 20: произведение отличающихся от нуля более чем на ϵ элементов массива.
4. (2 балла) В одномерном массиве, состоящем из n целых элементов, найти (если возможно – вывести соответствующую информацию):

Вариант 1: произведение отрицательных элементов массива, расположенных между минимальным и максимальным элементами.

Вариант 2: сумму отрицательных элементов массива, расположенных между первым и последним положительными элементами.

Вариант 3: произведение ненулевых элементов массива, расположенных между первым и последним отрицательными элементами.

Вариант 4: произведение элементов массива, расположенных между минимальным по модулю и максимальным по модулю элементами.

Вариант 5: сумму положительных элементов массива, расположенных между первым и последним нулевыми элементами.

Вариант 6: произведение ненулевых элементов массива, расположенных между первым и последним нулевыми элементами.

Вариант 7: произведение отрицательных элементов массива, расположенных между первым и последним положительными элементами.

Вариант 8: произведение элементов массива, расположенных после первого отрицательного элемента массива.

Вариант 9: произведение элементов массива, расположенных после минимального по модулю элемента массива.

Вариант 10: произведение элементов массива, расположенных после максимального по модулю элемента массива.

Вариант 11: сумму элементов массива, расположенных после первого отрицательного элемента массива.

Вариант 12: произведение элементов массива, расположенных до первого отрицательного элемента массива.

Вариант 13: произведение элементов массива, расположенных после первого положительного элемента массива.

Вариант 14: сумму элементов массива, расположенных между первым и последним отрицательными элементами.

Вариант 15: произведение элементов массива, расположенных до минимального по модулю элемента массива.

Вариант 16: произведение элементов массива, расположенных до максимального по модулю элемента массива.

Вариант 17: произведение элементов массива, расположенных после максимального элемента массива.

Вариант 18: сумму элементов массива, расположенных после максимального элемента массива.

Вариант 19: произведение элементов массива, расположенных до третьего положительного элемента массива.

Вариант 20: сумму элементов массива, расположенных правее предпоследнего отрицательного элемента.

5. (2 балла) В одномерном массиве, состоящем из n вещественных элементов, заменить:
- Вариант 1: все положительные элементы на противоположные.
- Вариант 2: все отличающиеся от нуля не больше чем на ϵ элементы на случайно сгенерированные отрицательные.
- Вариант 3: все элементы, меньшие чем введенное с клавиатуры число C , на ноль.

Вариант 4: все отрицательные элементы на обратные.

Вариант 5: все ненулевые элементы на обратные.

Вариант 6: все элементы, меньшие чем введенное с клавиатуры число C , на противоположные.

Вариант 7: все ненулевые элементы на случайно сгенерированные положительные.

Вариант 8: все элементы, большие чем введенное с клавиатуры число C , на обратные. Если отличающийся от нуля не больше чем на ϵ элемент подходит под условие, его не заменять.

Вариант 9: все отрицательные элементы на противоположные.

Вариант 10: все элементы, большие чем введенное с клавиатуры число C , на противоположные.

Вариант 11: все элементы, меньшие чем введенное с клавиатуры число C , на обратные. Если отличающийся от нуля не больше чем на ϵ элемент подходит под условие, его не заменять.

Вариант 12: все положительные элементы на обратные.

Вариант 13: все положительные элементы на среднее арифметическое соседних (если первый или последний является положительным, заменить на половину соседнего).

Вариант 14: все ненулевые элементы на случайно сгенерированные отрицательные.

Вариант 15: все отличающиеся от нуля не больше чем на ϵ элементы на случайно сгенерированные положительные.

Вариант 16: все отрицательные элементы на среднее арифметическое соседних (если первый или последний является отрицательным, заменить на половину соседнего).

Вариант 17: все элементы, большие чем введенное с клавиатуры число C , на ноль.

Вариант 18: все ненулевые элементы на среднее арифметическое соседних (если первый или последний является ненулевым, заменить на половину соседнего).

Вариант 19: все отличающиеся от нуля не больше чем на ϵ элементы на среднее арифметическое соседних (если отличающийся от нуля элемент является первым или последним, заменить на половину соседнего).

Вариант 20: все отличающиеся от нуля не больше чем на ϵ элементы на величины, обратные их номерам (если нулевой по номеру элемент окажется равен нулю, его не заменять).

6. Написать программу, решающую задачу линейного поиска элемента в заданном вещественном массиве.
7. Написать программу, которая упорядочивает вещественный массив методом выбора.
8. Написать программу, которая упорядочивает целочисленный массив методом быстрой сортировки.
9. Написать программу, решающую задачу бинарного поиска в отсортированном целочисленном массиве.