



Task 47 (revision)

Task 47 (revision)

**C/C++ Basic Syntax.
Pointers. One-dimensional
Dynamic Arrays**



LEARN. GROW. SUCCEED.

© 2025. STEP Computer Academy - a leader in the field of professional computer education
by Viktor Ivanchenko / ivanvikvik@gmail.com / Minsk

Task #47 (revision)

Базовый синтаксис языка C/C++. Основы использования указателей. Динамические одномерные массивы. Адресная арифметика

Цель работы

Проверить и практически закрепить работу с динамической памятью с использованием указателей и адресной арифметики в языке C/C++ на примере работы с динамическими одномерными массивами.

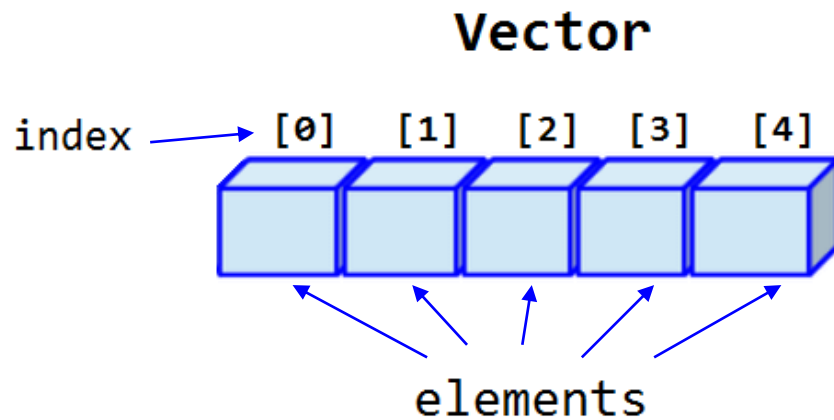
Требования

- 1) Для каждого задания в начале рекомендуется разработать блок-схему алгоритма решения.
- 2) Проект обязательно должен быть сразу реализован и сохранён под управление системой контроля версий (VCS) **git** и в последующем залит в централизованный репозиторий на облачном хостинг-сервисе **GitHub**.
- 3) Все программы должны быть разбиты на отдельные функции. При выполнении задания необходимо по максимуму пытаться разрабатывать универсальный, масштабируемый, легко поддерживаемый и читаемый код.
- 4) В соответствующих компонентах бизнес-логики необходимо предусмотреть **«защиту от дурака»** (*fool-proof*), т.е. прежде чем выполнять действия с данными нужно проверить, являются ли данные адекватными (непротиворечивыми).
- 5) Одномерные и многомерные структуры данных рекомендуется реализовывать на базе **динамических C/C++ массивов**.
- 6) Также рекомендуется придерживаться **Single Responsibility Principle, SRP** (принципа единственной ответственности) – постарайтесь вынести основную бизнес-логику задания в отдельную функцию или функции (т.е. архитектура приложения должна минимум состоять из нескольких функций).

- 7) Программа должна обязательно быть снабжена комментариями на английском языке, в которых необходимо указать краткое предназначение программы, её версию, ФИО разработчика, номер группы и дату разработки.
- 8) Исходный текст основного кода и демонстрационной программы рекомендуется также снабжать поясняющими краткими комментариями.
- 9) Если логически не подразумевается или в задании иного не указано, то входными и выходными данными являются целые числа.
- 10) Программа должна быть снабжена дружелюбным и интуитивно понятным интерфейсом для взаимодействия с пользователем.
- 11) Предусмотреть вывод на консоль удобочитаемого результата для пользователя. Рекомендуется для программного интерфейса использовать английский язык.
- 12) При проверке работоспособности приложения необходимо проверить все тестовые случаи.
- 13) Для предоставляемого решения задания также необходимо подсчитать его алгоритмическую сложность (***Big O Notation***) для всех типов измерений: худший, средний и лучший случаи.
- 14) При разработке программ придерживайтесь соглашений по написанию кода на языке C/C++ (C++ *Code-Convention*).

Основное задание

Вектор – одно из математических понятий, обозначающее последовательность однородных (однотипных) значений. Во многих языках программирования вектор эмулируется (реализуется) с помощью встроенных одномерных массивов или списков.



- 1) **Сумма элементов** [The sum of elements]. Необходимо разработать программу, которая вычисляет сумму элементов массива, расположенных между первыми максимальным по модулю и минимальным по модулю элементами. Ниже приведен рекомендуемый вид экрана программы (данные, введенные пользователем, выделены полужирным) и тестовые данные. Первоначальные данные вектора могут быть автоматически сгенерированы с помощью встроенного генератора псевдослучайных чисел.

Тест 01

Введите элементы вектора: **-1 2 -3 4 -5 6 -7 8 -9 10**

Сумма элементов вектора: -4

Тест 02

Input vector elements: **7 5 1**

Sum of vector elements: 5

Тест03

...

- 2) **Среднее арифметическое ненулевых элементов** [*The arithmetic mean of non-zero elements*]. Дан вектор целых чисел. Необходимо разработать программу, которая вычисляет среднее арифметическое ненулевых элементов вектора. Ниже приведен рекомендуемый вид экрана программы (данные, введенные пользователем, выделены полужирным) и тестовые данные. Первоначальные данные вектора могут быть автоматически сгенерированы с помощью встроенного генератора псевдослучайных чисел.

Тест 01

*Введите элементы вектора: **23 0 45 -5 12 0 -2 30 0 64***

Сумма элементов вектора: 184

Количество ненулевых элементов вектора: 7

Среднее арифметическое ненулевых элементов вектора равно 23.86.

Test 02

*Input vector elements: **4 0 56 12 -3 0 5 -7 91 12***

Sum of vector elements: 170

Number of non-zero vector elements: 8

The arithmetic mean of the non-zero elements of the vector is 21.25.

Test03

...

- 3) **Среднее арифметическое неэкстремальных элементов** [*The arithmetic mean of non-extreme elements*]. Дан вектор натуральных чисел. Необходимо написать программу, которая вычисляет среднее арифметическое элементов вектора без учета минимального и максимального элементов. Ниже приведен рекомендуемый вид экрана программы (данные, введенные пользователем, выделены полужирным) и тестовые данные. Первоначальные данные вектора могут быть автоматически сгенерированы с помощью встроенного генератора псевдослучайных чисел.

Тест 01

*Введите элементы вектора: **12 4 5 7 15 4 10 17 23 7***

Минимальный элемент вектора: 4

Максимальный элемент вектора: 23

Среднее арифметическое неэкстремальных элементов вектора: 10.43.

Тест 02

*Input vector elements: **4 1 56 12 3 1 5 7 56 12***

Minimum vector element: 1

Maximum vector element: 56

The Arithmetic mean of non-extreme vector elements: 7.166666666666667.

Тест 03

*Input vector elements: **1 2 3***

Minimum vector element: 1

Maximum vector element: 3

The Arithmetic mean of non-extreme vector elements: 2.

Тест 04

*Input vector elements: **1 2 5 1 4 1 5 1 5***

Minimum vector element: 1

Maximum vector element: 5

The Arithmetic mean of non-extreme vector elements: 3.

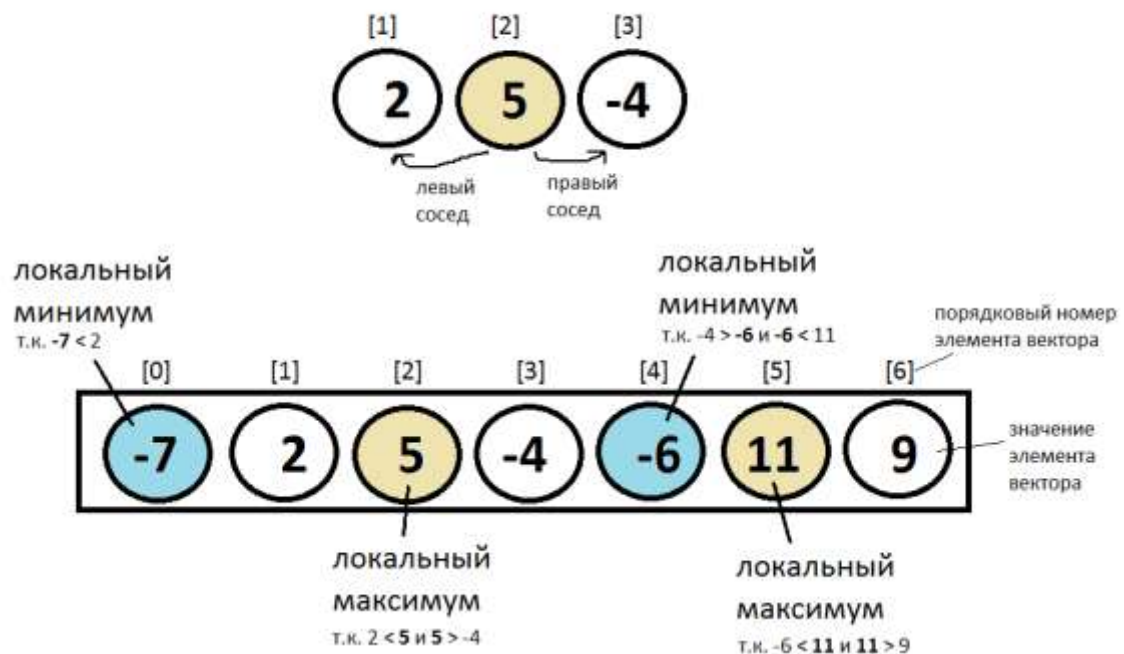
Тест 05

...

- 4) **Количество локальных минимумов и максимумов** [The Number of local minima and maxima]. Дан вектор целочисленных значений. Необходимо написать программу, которая подсчитывает **количество локальных минимумов** и **максимумов**.

Ниже приведен рекомендуемый вид экрана программы (данные, введенные пользователем, выделены полужирным) и тестовые данные. Первоначальные данные вектора могут быть автоматически сгенерированы с помощью встроенного генератора псевдослучайных чисел.

В математике **локальным минимумом** называется элемент вектора, значение которого строго меньше его соседних элементов. Следовательно, **локальным максимумом** называется элемент вектора, значение которого строго больше его соседних элементов.



Тест 01

Введите элементы вектора: **-7 2 5 -4 -6 11 9**

Количество локальных минимумов: 3

Количество локальных максимумов: 2

Тест 02

Input vector elements: **12 4 5 7 15 4 16 12 13 7**

The Number of local minima: 3

The Number of local maxima: 4

Test 03

Input vector elements: **1 1 1 1 1**

The Number of local minima: 0

The Number of local maxima: 0

Test 04

Input vector elements: **1 2**

The Number of local minima: 1

The Number of local maxima: 1

Test 05

...

- 5) **Сортировка из указанного промежутка** [The Sort from the specified range]. Дан вектор натуральных чисел. Необходимо написать программу, которая сортирует часть элементов заданного вектора в указанном диапазоне порядковых номеров либо по возрастанию, либо по убыванию. Ниже приведен рекомендуемый вид экрана программы (данные, введенные пользователем, выделены полужирным) и тестовые данные. Первоначальные данные вектора могут быть автоматически сгенерированы с помощью встроенного генератора псевдослучайных чисел.

Тест 01

Введите элементы вектора: **12 4 5 7 15 4 10 17 23 7**

Введите диапазон сортировки (от a до b): **2 9**

Введите направление сортировки (a или d): **a**

Элементы результирующего вектора: 12 4 4 5 7 10 15 17 23 7

Тест 02

Input vector elements: **12 4 5 7 15 4 10 17 23 7**

Input the sort range (from a to b): **2 9**

Input the sort direction (a or d): **d**

The Result of vector elements: 12 23 17 15 10 5 4 4 7

Тест 03

Input vector elements: **12 4 5 7 15 4 10 17 23 7**

Input the sort range (from a to b): **9 2**

Input the sort direction (a or d): **d**

The Result of vector elements: 12 23 17 15 10 5 4 4 7

Тест 04

...

Best of LUCK with it, and remember to HAVE FUN while you're learning :)

Victor Ivanchenko

