

Практическое задание №10 (шаблоны функций и макросы)

Изучить материалы Лекции 7 (раздел 7.4) и Лекции 9 (разделы 9.3, 9.4 и 9.5).

Создать проект с необходимыми функциями. Тестирование функций осуществлять с помощью **gtest**. Выполнить задания согласно своему варианту. Использовать стандартные библиотечные функции **запрещается**.

1. Реализовать **шаблонные функции** (массив передаётся в функцию в качестве параметра), выполняющие следующие действия:
 - 1) изменение одномерного массива элементов числовых типов согласно своему варианту;
 - 2) возвращение значения максимального элемента одномерного массива со специализацией шаблона для массива строк типа `char *` (определение максимальной строки при лексикографическом сравнении);
 - 3) возвращение суммы элементов одномерного массива со специализацией для массива строк (типа `char*`); для строк под суммой элементов будем понимать конкатенацию (сцепление) всех строк – элементов массива;
 - 4) возвращение в виде результата вновь созданного одномерного динамического массива (увеличенного размера на 2 элемента по сравнению с передаваемым массивом) и освобождение памяти из-под переданного динамического массива; массив-результат должен содержать все элементы передаваемого в функцию массива, а в конец массива-результата добавляются максимальный элемент (определённый с помощью функции, созданной в соответствии с пунктом 1.2 задания) и сумма элементов передаваемого в функцию исходного массива (определённая с помощью функции пункта 1.3).

При необходимости реализовать вспомогательные функции, например, для определения длины строки, для копирования строк и т.п.

Можно добавить другие шаблоны функций, например, для сравнения массивов, ввода/вывода массивов и т.п.

Протестировать функции для массивов целых чисел, массивов вещественных чисел и массивов строк типа `char *`.

2. Реализовать согласно своему варианту алгоритм сортировки одномерного массива **числовых элементов** (см. раздел 3 в файле «Материалы к Практическому заданию №10») в виде **макроопределения (макроса) с параметром**, задающим тип элементов массива (см. раздел 4 и листинг П10.4 в файле «Материалы к Практическому заданию №10»).

Тестирование произвести для контроля сортировки массивов целых и вещественных чисел.

3. Создать и проанализировать **файл с результатами препроцессорной обработки** текста файла, содержащего вызовы реализованного в соответствии с пунктом 2 макроса (см. раздел 5 в файле «Материалы к Практическому заданию №10»).

Для сдачи проекта использовать структуру в файловой системе:

gxxxxx/10/Task10/*** – файлы с реализацией пунктов задания (имена файлов давать в соответствии с вариантами);

gxxxxx/10/Task10.Tests/*** – файлы с тестами пунктов задания (имена файлов давать в соответствии с вариантами).

Необходимые условия сдачи:

- стиль программирования должен соответствовать установленным правилам;

- имена переменных и функций должны быть осмысленными; никакой транслитерации и нелогичности;
- программа должна быть протестирована с помощью gtest;

Вариант 1

1. Добавить к каждому элементу одномерного массива минимальный по абсолютной величине элемент массива.
2. Реализовать алгоритм сортировки простыми вставками (по возрастанию).

Вариант 2

1. Заменить все нули в одномерном массиве средним арифметическим всех элементов массива.
2. Реализовать алгоритм сортировки простыми обменами (пузырьковая, по возрастанию).

Вариант 3

1. Вычесть из каждого элемента одномерного массива значение минимального элемента массива.
2. Реализовать алгоритм сортировки прямого выбора (по возрастанию).

Вариант 4

1. Добавить к каждому элементу одномерного массива корень квадратный из абсолютного значения минимального элемента массива.
2. Реализовать алгоритм сортировки простыми вставками (по убыванию).

Вариант 5

1. Домножить все нечётные по положению элементы одномерного массива на среднее арифметическое элементов массива.
2. Реализовать алгоритм сортировки простыми обменами (пузырьковая, по убыванию).

Вариант 6

1. Вычесть из каждого элемента одномерного массива сумму всех элементов массива.
2. Реализовать алгоритм сортировки посредством прямого выбора (по убыванию).

Вариант 7

1. Домножить каждый элемент одномерного массива на половину минимального элемента массива.
2. Реализовать алгоритм сортировки простыми вставками (по возрастанию).

Вариант 8

1. Домножить каждый элемент одномерного массива на минимальный элемент массива.
2. Реализовать алгоритм сортировки простыми обменами (пузырьковая, по возрастанию).

Вариант 9

1. Домножить каждый третий элемент одномерного массива на удвоенную сумму первого и последнего элементов.
2. Реализовать алгоритм сортировки прямого выбора (по возрастанию).

Вариант 10

1. Домножить каждый элемент одномерного массива на полусумму максимального и минимального элементов.
2. Реализовать алгоритм сортировки простыми вставками (по убыванию).

Вариант 11

1. Заменить все положительные элементы одномерного массива квадратом минимального элемента.
2. Реализовать алгоритм сортировки простыми обменами (пузырьковая, по убыванию).

Вариант 12

1. Заменить все положительные элементы одномерного массива максимальным элементом.
2. Реализовать алгоритм сортировки посредством прямого выбора (по убыванию).

Вариант 13

1. Изменить все элементы одномерного массива так: умножить на минимальный элемент массива и к результату добавить максимальный элемент.
2. Реализовать алгоритм сортировки простыми вставками (по возрастанию).

Вариант 14

1. Добавить к каждому элементу одномерного массива среднее арифметическое наименьшего и наибольшего по абсолютной величине элементов массива.
2. Реализовать алгоритм сортировки сортировка простыми обменами (пузырьковая, по возрастанию).

Вариант 15

1. Заменить каждый нечётный по положению элемент одномерного массива разностью максимального и минимального элементов массива.
2. Реализовать алгоритм сортировки прямого выбора (по возрастанию).

Вариант 16

1. Заменить каждый второй отрицательный элемент одномерного массива половиной минимального элемента массива.
2. Реализовать алгоритм сортировки простыми вставками (по убыванию).

Вариант 17

1. Заменить каждый третий положительный элемент одномерного массива средним арифметическим всех элементов массива.
2. Реализовать алгоритм сортировки простыми обменами (пузырьковая, по убыванию).

Вариант 18

1. Добавить к каждому элементу одномерного массива среднее арифметическое элементов массива.
2. Реализовать алгоритм сортировки посредством прямого выбора (по убыванию).