

3 Лабораторная

C++ - контейнеры vector, map

За выполнение всех заданий лабораторной - **8 баллов** (до 30 апреля, за сдачу после 30 апреля ставлю половину).

Выполнить один вариант. Номер варианта = номер из списка подгруппы % 6

- При выполнении первого задания использовать контейнер vector из стандартной библиотеки, для сортировки число элементов пользователь вводит с клавиатуры, значения элементов задаются случайными в диапазоне (-100, 100).
- Второе задание: есть текстовый файл с произвольным текстом. Создать контейнер map, ключом которого будет слово (string), значением (int) – частота встречаемости этого слова из текста. Считать файл и заполнить контейнер словами и частотой их встречаемости. Написать функцию, которая выводит в консоль все слова с их частотой встречаемости и выполнить второе подзадание из своего варианта.

Вариант 0

1. Реализовать **сортировку выбором** - сначала отыскивается минимальный (максимальный) элемент и помещается в начало списка, далее второй минимальный (максимальный) и так далее, пока не отсортируем.
2. Вывести все слова, которые встречаются в тексте менее двух раз.

Вариант 1

1. Реализовать **сортировку вставкой** - последовательность делится на две части: уже отсортированную и ещё не отсортированную. Элемент из второй части поочередно вставляется в первую часть так, чтобы поддерживалось свойство отсортированной части.
2. Вывести слова которые встречаются однократно в тексте.

Вариант 2

1. Реализовать **пузырьковую сортировку** - обходим массив, попутно меняя местами неотсортированные соседние элементы. В результате прохода на последнее место "всплывает" максимальный элемент.

2. Пользователь вводит с клавиатуры слово - найти это слово в контейнере и вывести, сколько оно раз встречается в тексте.

Вариант 3

1. Реализовать **шейкерную сортировку** - аналогична пузырьковой сортировки (вариант 2), но после одного прохода, как в "пузырьке", очередной проход выполняется в обратном направлении "выдавливая" минимальный элемент в начало массива.

2. Вывести все слова, которые встречаются более, чем $\text{число_самого_встречаемого_слова} - 2$.

Вариант 4

1. Реализовать **чётно-нечётную сортировку** - аналогична пузырьковой сортировки (вариант 2), но за один "шаг" делается два прохода - сравниваем нечётные с чётными, потом, наоборот, чётные с нечётными. Останавливаемся, когда после двух проходов нет обмена.

2. Вывести все слова с их частотой встречаемости, которые начинаются на одну из первых трёх букв алфавита.

Вариант 5

1. Реализовать **сортировку расчёской** - аналогична пузырьковой сортировки (вариант 2), но сравниваются элементы не соседние, а на большем расстоянии. Определяем фактор уменьшения, равный 1,247. Сначала расстояние между сравниваемыми элементами массива равно $\text{размер массива} / \text{фактор уменьшения}$. После каждого прохода новое расстояние равно предыдущее расстояние / фактор уменьшения, и так, пока расстояние не станет равным единице, дальше сортировка уже завершается "пузырьком".

2. Пользователь вводит с клавиатуры символ. Вывести на экран все слова с их частотой встречаемости, которые начинаются на этот символ.