

В этой лабораторной работе от вас требуется продемонстрировать умение разрабатывать код с использованием контейнеров библиотеки STL на примере контейнера `std::vector`.

ЗАДАНИЕ ДЛЯ ВЫБРАВШИХ БАЗОВЫЙ УРОВЕНЬ

1. показать умение работать с векторами: объявление, добавление элементов в вектор, вставка и удаление элементов из вектора; применение алгоритмов `for_each` и `random_shuffle` при работе с векторами;
2. привести результат сравнения времени вывода 10 000 элементов типа `char`, `int`, `double` и строк `char[]` с помощью функций `printf` (ANSI C) и `std::cout` (C++), результат представить в виде таблицы;

	<code>char</code>	<code>int</code>	<code>double</code>	"PKIMS RULEZ"
<code>printf</code>				
<code>std::cout</code>				

3. разработать код функции сортировки вставками и для целочисленных векторов размером 3, 5 и 10 элементов привести результаты сравнения быстродействия разработанной функции и реализованного в стандартной библиотеке алгоритма `std::sort`.

	3 элемента	5 элементов	10 элементов
<code>insertion_sort</code>			
<code>std::sort</code>			

ЗАДАНИЕ ДЛЯ ВЫБРАВШИХ ПОВЫШЕННЫЙ УРОВЕНЬ

Необходимо выполнить все задания на минимум. После этого необходимо выполнить дополнительные задания, приведённые ниже.

Эти задания выполняются последовательно.

- **Плюс балл.** Самостоятельно реализовать аналог класса `vector` для динамического массива целых чисел, который имеет следующие функции-члены класса:
`push_back(int val)` – для добавления элемента `val` в конец вектора;
`pop_back()` – для извлечения (удаления) последнего элемента;
`back()` – для доступа к последнему элементу;
`size()` – для получения размера вектора;
`insert(int pos, int val)` – для вставки элемента `val` на позицию `pos`;
`clear()` – для очистки вектора;
`erase(int pos)` – для удаления элемента на позиции `pos`;
`erase(int from, int to)` – для удаления элемента с позиции `from` до позиции `to`;
- **Плюс балл.** Модифицировать предыдущий вариант так, чтобы вместо целых чисел использовался шаблон (`template`).