

ADT Containers

C++ STL – lesson 5

1 Container sorting

Создайте вектор. Заполните его случайными числами. Отсортируйте элементы вектора в порядке убывания их значений.

2 Uniqueness

Создайте вектор, заполните его таким образом, чтобы некоторые значения повторялись несколько раз. Выведите на экран количество уникальных элементов.

Например, для вектора { 1, 2, 1, 1, 3, 4, 3, 2, 7 } количество уникальных элементов равно пяти: {1, 2, 3, 4, 7}.

Алгоритм выполнения задания похож на алгоритм из предыдущего задания.

3 Cards

Создайте структуру, описывающую игральную карту - масть и достоинство. Подумайте какой контейнер следует использовать для хранения колоды карт. Положите несколько карт в колоду. Извлеките их из колоды и выведите на экран.

4 Excellent students

Создайте контейнер, хранящий фамилии учащихся и их средний балл. Сделайте выборку фамилий отличников в отдельный вектор. Выведите их фамилии из вектора на экран.

Отличниками считаются учащиеся со средним баллом 4.5 и выше.

5 MinusMinMax

Написать функцию, которая удаляет минимальный и максимальный элемент из вектора. Если после этого вектор становится пустым, или минимальный элемент равен максимальному, функция возвращает `false`, иначе `true`.

6 Vector semi-absorbtion

Разработайте функцию, которой передается два вектора. Функция добавляет первую половину из `вектор_2` в середину `вектора_1` и возвращает `true`. Если середина у `вектора_1` или `вектора_2` отсутствует, или `вектор_1` пустой, возвращает `false`.

7 BFS

Создайте `PriceList` для товаров. Написать функцию, которая добавляет скидку 25% всем товарам, цена которых выше некоторого порогового значения. При отсутствии товаров со скидкой, функция возвращает `false`. Выведите на экран вашу версию расшифровки аббревиатуры BFS.

8 Liquidation

Создать контейнер, хранящий `id` товаров и соответствующий им срок годности (`expiration date`). Написать функцию, которая удаляет все просроченные товары и возвращает количество, удаленных товаров. Допустимо в качестве количественной характеристики даты использовать только год. Примечание: при обходе элементов в цикле, например, с использованием Range-based for loop удаление элемента с последующей итерацией по циклу может привести к непредсказуемо катастрофическим результатам.

Подсказка: просроченные удалены в два приема. При первом приеме в отдельный контейнер складываются все id товаров, подлежащие удалению. Далее интерпретация идет по этому отдельному контейнеру, удаляются все id соответствующие для удаления элементов из основного контейнера.

9 Card deck

Сформируйте колоду из тасованных карт. Выложите карты на стол (экран) по порядку из колоды.

Подсказка: создайте структуру, описывающую одну карту. Поместите все карты в текст по порядку. Разверните структуру текста в диапазон размера вектора. Распределите случайный элемент из вектора в стек и удалите его из вектора, пока вектор не окажется пустым. Сдвигните все карты из колоды на стол.

10 Probability

Создайте строку. Поместите в нее какой-нибудь рассказ. Вычислите вероятность встречаемости символов в тексте. Под вероятностью символа $P(a)$ понимать отношение количества символов 'a' к общему количеству символов в тексте. Выведите символы в порядке уменьшения их вероятности.

Подсказка: создайте словарь вероятностей, dict, применяй в качестве ключа символ, а в качестве значения количество данного символа в тексте. Заполните все символы строки в словарь, каждый раз увеличивая значение на 1. Затем в словарь примените и сортируйте по ключу по порядку. От вас требуется вывести символы в порядке уменьшения их вероятности. Чтобы упростить задачу - переименуйте словарь в словарь вероятностей, создайте новый словарь вероятностей, ключ, в качестве значения вычислите вероятности, а в качестве значения символ. Напишите функцию - сортируй, которая два списка выводит сортировку вероятности. От вас можно потребовать, увеличивая значение вероятности на одну миллионную, пока колода не истощит. Для вывода элементов в порядке убывания вероятностей используйте обратной отсортировки.

11 Kueue

Смоделируйте жизненную ситуацию. В семье появился ребенок. Они встали в очередь в детский сад. Пока семья ждала второго ребенка, очередь пополнялась. Первый ребенок поступил в сад раньше, чем родился и был поставлен в очередь второй ребенок. Посчитайте сколько счастливых случаев попало в детский сад между поступлением первого и второго ребенка.

12 Priority Queue

В операционной системе одновременно запущено множество программ (процессов). Некоторые из них важны и должны выполняться микропроцессором в первую очередь. Чем важнее процесс, тем выше его приоритет. Попробуйте найти в `STL` готовый контейнер для реализации такой возможности, если он отсутствует разработайте свой тип. Если присутствует используйте его, но прежде разработайте свой тип.

При добавлении элемента в очередь с приоритетом, кроме самого элемента передается его приоритет. Элементы очереди всегда располагаются в отсортированном порядке согласно их приоритетам. При совпадении приоритетов, приоритетнее является элемент, добавленный раньше.

При вызове метода `push(...)` элемент располагается в вашем контейнере в нужной позиции.

При вызове метода `pop()` из очереди возвращается и удаляется элемент с самым высоким приоритетом.

Подумайте, какую структуру данных удобнее использовать в данном случае. Подумайте еще раз.