

Лабораторная работа №5.

Многофайловые проекты. Стандартная библиотека шаблонов.

Цели и задачи работы: программирование и отладка программ формирования и обработки контейнеров, комбинации контейнеров.

Задание к работе: Написать программу решения задачи в соответствии с индивидуальным вариантом.

Методика выполнения работы:

1. Разработать алгоритм решения задачи по индивидуальному заданию.
2. Написать и отладить программу решения задачи.
3. Протестировать работу программы на различных исходных данных.

Замечание: Нумерация для пользователей с единицы. Использовать только контейнеры и циклы Range-based for loop.

Информационное обеспечение:

<https://ru.cppreference.com/w/cpp/container>

<http://www.cplusplus.com/reference/stl/>

ЗАДАНИЕ 5.1 «Учет товаров на складе». 2 ЯП

Реализовать программу для учета товаров на складе. Все ячейки на складе имеют свои адреса (например A1739), адрес состоит из следующих обозначений:

- 1) А, Б, В - зона хранения - теплый, холодный склад или часть склада;
- 2) 17 - порядковый номер стеллажа;
- 3) 3 - порядковый номер вертикальной секции стеллажа;
- 4) 9 - порядковый номер полки.

В каждую ячейку помещается не более 10 единиц товара. Программа должна позволять добавлять товары в ячейки, просматривать состояние склада, убирать товар из ячейки.

Для добавления товара в ячейку использовать команду ADD <наименование товара> <количество> <адрес ячейки>. Пример: ADD Апельсины 8 A1739. Так как размер ячейки ограничен 10 единицами товара, при попытке добавления большего кол-ва товара пользователю должно выводиться соответствующее сообщение.

Для удаления товара из ячейки использовать команду REMOVE <наименование товара> <количество> <адрес ячейки>. Пример: REMOVE Апельсины 8 A1739. Если в ячейке недостаточно товара для списания, пользователю должно выводиться соответствующее сообщение.

Для получения информации о состоянии склада использовать команду INFO. Команда должна выводить насколько процентов загружен склад, на сколько процентов загружена каждая зона склада, а также выводить содержание каждой ячейки в которой есть хотя бы 1 единица товара и выводить список адресов пустых ячеек.

Вариант	Кол-во зон хранения	Кол-во стеллажей в каждой зоне	Кол-во вертикальных секций стеллажа	Кол-во полок в вертикальной секции	Общая вместимость единиц товара
1	3	20	5	2	6000
2	10	3	1	5	1500
3	4	8	2	1	640

4	1	10	7	4	2800
5	2	4	18	5	7200
6	7	15	11	3	34650
7	2	6	4	4	1920
8	3	14	6	20	50400
9	1	5	5	15	3750
10	2	19	4	8	12160

ЗАДАНИЕ 5.2 «Реализация электронной очереди». 2 ЯП

Реализовать программу для электронной очереди в поликлинике. На вход в программу подается кол-во окон способных обрабатывать очередь посетителей. Далее с использованием команды ENQUEUE посетители добавляются в очередь, команда принимает продолжительность посещения и выдает номер талона. После того как введено нужное количество посетителей вводится команда DISTRIBUTE. Программа должна вывести распределение очереди посетителей на все окна, распределение должны быть такое, чтобы обработать очередь максимально быстро.

Пример работы программы:

“>>>” – вывод данных

“<<<” - ввод данных

>>> Введите кол-во окон

<<< 2

<<< ENQUEUE 5

>>> T764

<<< ENQUEUE 10

>>> T047

<<< ENQUEUE 5

>>> T903

<<< DISTRIBUTE

>>> Окно 1 (10 минут): T764, T903

>>> Окно 2 (10 минут): T047

ЗАДАНИЕ 5.3 «Многофайловый проект ENUM» C++

Необходимо реализовать систему хранения и обработки информации по заданию:

Вариант	Задание
1	График движения поездов
2	График движения трамваев г. Новосибирска
3	График движения самолетов
4	График движения троллейбусов

Реализовать многофайловый проект, предусматривающий обработку запросов:

Вариант	Задание	Расшифровка
1	CREATE_TRAIN train town1 town2 town3 ... town _n	Создание поезда с именем train, который проходит через города town1 town2 town3 ... town _n
	TRAINS_FOR_TOWN town	Вывод всех поездов, которые проходят через город town
	TOWNS_FOR_TRAIN train	Вывод всех городов, которые проезжает поезд с именем train. Для каждого города прописать, какие поезда

		проезжают этот город (не включая train)
	TRAINS	Отобразить все поезда с указанием остановок
2	CREATE_TRAM tram stop1 ... stop _n	Создание трамвая с именем tram, который проходит через остановки
	TRAMS_IN_STOP stop	Вывод всех трамваев, которые проходят через конкретную остановку
	STOPS_IN_TRAM tram	Вывод всех остановок, которые проезжает трамвай с именем tram. Для каждой остановки прописать, какие трамваи идут через эту остановку (не включая tram)
	TRAMS	Отобразить все трамваи с указанием остановок
3	CREATE_PLANE plane town1 town2 town3 ... town _n	Создание самолета с именем plane, который пролетает через города town1 town2 town3 ... town _n
	PLANES_FOR_TOWN town	Вывод всех самолетов, которые пролетают через город town
	TOWNS_FOR_PLANE plain	Вывод всех городов, которые пролетает самолет с именем plane. Для каждого города прописать, какие самолеты делают остановку в этом городе (не включая plain)
	PLANES	Отобразить все самолеты с указанием городов-остановок
4	CREATE_TRL trl stop1 ... stop _n	Создание троллейбуса с именем trl, который проходит через остановки
	TRL_IN_STOP stop	Вывод всех троллейбусов, которые проходят через конкретную остановку
	STOPS_IN_TRL trl	Вывод всех остановок, которые проезжает троллейбус с именем trl. Для каждой остановки прописать, какие троллейбусы идут через эту остановку (не включая trl)
	TRLS	Отобразить все троллейбусы с указанием остановок

В программе предусмотреть

Команды хранить в классе enum:

Например,

```
enum class Type {
    CREATE_TRAIN,
    TRAINS_FOR_TOWN,
    TOWNS_FOR_TRAIN,
    TRAINS
};
```

};

Предусмотреть создание многофайлового проекта: основной файл cpp, header+source для реализации действий (4 команды = 4 функции).

В отчет вставить осмысленные примеры остановок-городов.

Пример:

Трамвай:

Input
TRAMS TRAMS_IN_STOP Marksa STOPS_IN_TRAM 18 CREATE_TRAM 18 4 Student Marksa TVset Cosmos CREATE_TRAM 15 4 Student Cosmos TVset Titova TRAMS_IN_STOP Student CREATE_TRAM 666 6 Student Gorskyi NGTU NGTUMAIN IKEA Beach CREATE_TRAM 999 4 NGTU SibGUTI NSU Forest STOPS_IN_TRAM 999 TRAMS
Output
Trams is absent Stops is absent Trams is absent 18 15 Stop NGTU: 666 Stop SibGUTI: 0 Stop NSU: 0 Stop Forest: 0 TRAM 18: Student Marksa TVset Cosmos TRAM 15: Student Cosmos TVset Titova TRAM 666: Student Gorskyi NGTU NGTUMAIN IKEA Beach TRAM 999: NGTU SibGUTI NSU Forest

ЗАДАНИЕ 5.4 «Комбинация контейнеров» C++

Вариант 1

Реализовать автоматизированную систему:

Запрос	Расшифровка	Комментарий
FRIENDS <i>i j</i>	Записывает <i>i j</i> как друзей	-
COUNT <i>i</i>	Подсчет количества друзей <i>i</i>	Результат - Число
QUESTION <i>i j</i>	<i>i</i> дружат с <i>j</i> ?	YES/NO

Input:

1) N – количество запросов

2) Nзапросов

Output:

Обработать COUNT и QUESTION

придумать минимум 5 тестов в разными значениями (в том числе, повторяющимися)

Пример

Input

```
8
FRIENDS Peter Goward
COUNT Sally
FRIENDS Goward Sally
COUNT Goward
COUNT Peter
QUESTION Goward Peter
QUESTION Peter Sally
QUESTION Jenny Jastin
```

Output

```
0
2
1
YES
NO
NO
```

Вариант 2

Необходимо реализовать справочник регионов России.

На вход программе поступают следующие запросы:

Запрос	Расшифровка	Output
CHANGE <i>region</i> <i>new_center</i>	Создание региона <i>region</i> с административным центром <i>new_center</i> .	New region <i>region</i> with administrative center <i>new_center</i>
	Переименование названия административного (old_center) центра региона <i>region</i> в <i>new_center</i> .	Region <i>region</i> has changed its administrative center from old_center to <i>new_center</i>
RENAME <i>old_region</i> <i>new_region</i>	переименование региона со старым названием <i>old_region</i> в регион с новым названием <i>new_region</i>	<i>old_region</i> has been renamed to <i>new_region</i>
ABOUT <i>region</i>	вывод административного центра введенного региона <i>region</i>	<i>region</i> has administrative center
ALL	вывод всех административных центров	Region - Center

В случае ошибки (переименование несуществующего региона, переименование в старое значение и т.п.) вывести в поток ошибок «Incorrect»

Input:

1) N – количество запросов

2) N запросов

Output:

Обработать CHANGE, RENAME, ABOUT, ALL.

придумать минимум 5 тестов в разных значениях (в том числе, повторяющихся).

Пример

Input

6

CHANGE Sibir Novo-nikolaevsk

RENAME Sibir SibirSFO

CHANGE Sibir Novosibirsk

ABOUT SibirSFO

ABOUT Sibir

ALL

Output

New region Sibir with administrative center **Novo-nikolaevsk**

Sibir has been renamed to SibirSFO

Region SibirSFO has changed its administrative center from **Novo-nikolaevsk** to **Novosibirsk**

SibirSFO has administrative center **Novosibirsk**

Incorrect

SibirSFO - Novosibirsk

Вариант 3

У каждого студента есть расписание занятий. Посещать занятия необходимо в конкретный день определенного месяца. Необходимо автоматизировать работу расписания за счет обработки операций:

Запрос	Расшифровка	Output
CLASS <i>i s</i>	Установить дисциплину <i>s</i> в день <i>i</i> текущего месяца	-
NEXT	Смена месяца на следующий.	-
VIEW <i>i</i>	Организовать вывод всех дня <i>i</i> . При отсутствии пар – выдать соответствующее сообщение	In <i>i</i> day <i>N</i> classes in university: <i>class 1,...,class N.</i>
		In <i>i</i> day We are free!

INPUT: N – число операций ввода и затем эти операции (NEW, NEXT, VIEW).

Замечание, пары сохраняются при смене месяцев строго по дням и могут повторяться. Пропавшие в связи с переходом на новый календарный месяц пары никуда не пропадают, а переходят на предпоследний день месяца. Нумерация для пользователей с единицы.

Пример:

12 CLASS 5 INFORMATICA CLASS 31 YAP CLASS 30 PHISICS NEXT VIEW 5 VIEW 27 NEXT VIEW 31 VIEW 30 VIEW 27 CLASS 27 ENGLISH VIEW 27	In In 5 day 1 classes in university: INFORMATICA In 27 day 2 classes in university: PHISICS, YAP In 31 day We are free! In 30 day We are free! In 27 day 2 classes in university: PHISICS, YAP In 27 day 3 classes in university: PHICICS, YAP, ENGLISH
3 CLASS 4 A CLASS 4 A VIEW 4	In 4 day 1 classes in university: A

Вариант 4

Отчисление студента в процессе обучения – процесс весьма трудоемкий для руководства ВУЗа. Представим поток факультета АВТФ.

Будем считать, что студенты находятся в списке учащихся, и ежегодно данный список пополняется. Причем новые студенты попадают в конец списка всегда. В зависимости от ситуации в долгами, некоторые студенты могут быть кандидатами на отчисления. Но при ликвидации задолженностей могут и уйти из списка на отчисление.

Необходимо реализовать обработку следующих запросов:

Запрос	Расшифровка	Output
NEW_STUDENTS <i>number</i>	Добавить в конец очереди студентов в количестве <i>number</i>	>0: Welcome <i>number</i> clever students! <0: GoodBye <i>number</i> clever students!
SUSPICIOUS <i>number_student</i>	Студент с порядковым номером <i>number_student</i> является крайне подозрительным и входит в топ-лист на отчисление	The suspected student <i>number_student</i>
IMMORTAL <i>number_student</i>	Студент с порядковым номером <i>number_student</i> является неприкасаемым и из топ-листа на отчисление уходит. Такого студента никто и ничто не может отчислить	Student <i>number_student</i> is immortal!
TOP-LIST	Вывод отсортированного списка студентов, входящих в топ-лист на отчисление	List of students for expulsion: Student 1, ..., Student N
SCOUNT	Вывод количества студентов, входящих в топ-лист на отчисление	List of students for expulsion consists of N students

В случае ошибки (удаление числа студентов, превышающих их число в текущий момент, ввод данных, выходящих за диапазон)

ВЫВЕСТИ В ПОТОК ошибок «Incorrect»

Input

6

NEW_STUDENTS 20

SUSPICIOUS 10

NEW_STUDENTS 5

SUSPICIOUS 15

IMMORTIAL 10

TOP-LIST

SCOUNT

Output

Welcome 20 clever students!

The suspected student 10

Welcome 5 clever students!

The suspected student 15

Student 10 is immortal!

List of students for expulsion: Student 15

List of students for expulsion consists of 1 students