Задание 2.1

Постройте сбалансированное дерево (используйте класс map библиотеки STL), которое содержит значения V по ключам K (таблица 2.1). Постройте функции поиска элемента по значению и по ключу. Постройте функцию вывода содержимого дерева с помощью итераторов. Постройте функцию filter(), которая принимает предикат P и возвращает новое дерево с объектами, для которых предикат принимает истинное значение (для всех вариантов условие предиката: значение поля V выше некоторого порога *threshold*, в случае хранения нескольких полей достаточно проверить одно из них). Введите исключения для случаев, когда пользователь пытается добавить новый элемент с ключом, уже присутствующим в дереве.

Напишите функцию, которая возвращает вектор из различных значений, которые встречаются в объекте класса map, заполненном при решении задачи (рекомендуется использовать класс set).

Примечание: *В этом задании не требуется создавать класс дерева, нужно использовать класс map из библиотеки STL и написать отдельно требуемые функции (не методы класса).*

Код 2.1. Пример работы с контейнером map

```
    //красно-черное (сбланси рованное) дерево map, есть интерфейс доступа к значению по ключу

    using namespace std;

    #include <map>
    #include <iostream>

    int main()
    {

     map<string, int> marks;
     marks["Petrov"] = 5;
     marks["Ivanov"] = 4;
     marks["Sidorov"] = 5;
     marks["Nikolaev"] = 3;
     marks["Abramov"] = 4;

     cout << "\nMap:\n";
     //итератор пробегает по map
     map<string, int>::iterator it_m = marks.begin();
```

```
     while (it_m != marks.end())
     {
    //перемещение по списку с помощью итератора, нет операции [i]
          cout << "Key: " << it_m->first << ", value: " << it_m-
>second << "\n";
          it_m++;}}
```

Таблица 2.1. Ключи и хранимая в ассоциативном контейнере map информация

Вариант	Ключ K	Хранимая информация
1.	Адрес	«Объект жилой недвижимости». V: цена квартиры
2.	Название	«Сериал». V: рейтинг
3.	Название	«Смартфон». V: цена
4.	Фамилия и имя	«Спортсмен». V: количество медалей.
5.	Фамилия и имя	«Врач». V: рейтинг (вещественное число от 0 до 100) количество медалей
6.	Международный код	«Авиакомпания». V: количество обслуживаемых линий
7.	Название	«Книга». V: тираж
8.	Номер в небесном каталоге	«Небесное тело». V: расчётная масса в миллиардах тонн
9.	Название	«Населённый пункт». V: численность населения
10.	Имя или псевдоним исполнителя, название альбома	«Музыкальный альбом». V: количество проданных экземпляров
11.	Название фильма	«Фильм». V: доход
12.	Название производителя, имя	«Автомобиль». V: цена

	модели	
13.	Регистрационный номер автомобиля	«Автовладелец». V: фамилия, имя
14.	Название, год постройки	«Стадион». V: вместимость
15.	Название, город	«Спортивная Команда». V: число побед, поражений, ничьих, количество очков
16.	Номер карты	«Пациент». V: группа крови
17.	Фамилия и имя	«Покупатель». V: средняя сумма чека
18.	Фамилия и имя	«Школьник». V: дата рождения
19.	Фамилия и имя	«Человек». V: адрес
20.	Название	«Государство». V: численность населения
21.	Адрес	«Сайт». V: количество посетителей в сутки.
22.	Название	«Программа». V: разработчик
23.	Производитель, модель	«Ноутбук». V: размер экрана, количество ядер, объем оперативной памяти
24.	Марка, диаметр колеса	«Велосипед». V: тип, наличие амортизаторов
25.	Фамилия и имя	«Программист». V: уровень (число от 1 до 10)
26.	Псевдоним	«Профиль в соц.сети». V: количество друзей
27.	Псевдоним	«Супергерой». V: суперсила
28.	Производитель, модель	«Фотоаппарат». V: размер матрицы, количество мегапикселей
29.	Полный адрес	«Файл».

		V: дата последнего изменения
30.	Производитель, название	«Самолет». V: дальность полета, максимальная скорость

Задание 2.2

Постройте сбалансированное дерево (используйте класс multimap библиотеки STL), которое содержит значения V по ключам K (таблица 2.1). Продемонстрируйте работу контейнера при наличии нескольких элементов с одинаковыми ключами. Решите задачу 2.1 для класса multimap. Введите функцию, возвращающую все элементы дерева с одинаковыми ключами (ключ передаётся в функцию как параметр).

Примечание: *В этом задании не требуется создавать класс дерева, нужно использовать класс multimap из библиотеки STL и написать отдельно требуемые функции (не методы класса).*