Ассоциативный массив map

На данный момент существует отличный контейнер map, который позволяет вам ассоциировать элементы с чем угодно. Например, у вас есть список продуктов для покупки (список книг в библиотеке). Вам необходимо хранить сколько и чего нужно купить (название и количество книг в библиотеке).

Контейнер <map> позволяет нам хранить эти данные в виде ключ-значение (книга-количество, продукт-вес\объем...). Контейнер <map> является отсортированным массивом. Сортировка произведена по ключу.
Для использования <map> вам необходимо сначала его подключить:
#include <map>

Для создание контейнера достаточно написать:
map <key type, data type> map_name;

Например map<string, int> books; Содержит связку название книги / количество
или
map<string, CObject*> objects; Содержит связку название объекта / указатель на объект

Для доступа (или записи) в массив нужно писать:
map_name[key];

Контейнер map также имеет возможность работы с итераторами. Поэтому вы можете использовать с ним множество различных алгоритмов.
Элементы класса map имеют только уникальные значения ключей.
Элементы автоматически сортируются.

Класс multimap может иметь несколько элементов с одинаковым значением ключа.

begin() - итератор на первый элемент;
end() - итератор на элемент идущий после последнего;
rbegin() - итератор на послед ний элемент (для обратных алгоритмов);
rend() - итератор на позицию перед первым элементом (для обратных

алгоритмов).
размер отображения
empty() - возвращает истину, если размер отображения 0;
size() - возвращает число элементов;
max_size() - максимально возможный размер отображения.
count(key) - число элементов соответствующих указанному ключу.
Для класса map значения 1 или 0;
find(key) - итератор на первый элемент с указанным ключом;
lower_bound(key) - итератор на первый элемент, чей ключ больше или равен указанному ключу;
upper_bound(key) - итератор на первый элемент, чей ключ больше указанного ключа;
equal_range(key) - диапазон элементов, чей ключ равен указанному ключу;
[] - операция индексации по ключу.
swap(map& m) - обмен значениями с другим отображением;
insert(el) - вставка элемента, возвращается его позиция;
insert(beg,end) - вставка элементов из указанного диапазона;
erase(key) - удалить указанный элемент;
erase(it), erase(start,end) - удаляет элемент с заданным итератором или между заданными
clear() - удалить все элементы.
прочие методы

Для классов перегружены операции =,[],==,!=,<,>,<=,>=.

Пример. Программ для подсчета количества слов в тексте и вывод частоты их встречи в процентном соотношении (подсчет частоты встречи слов в тексте в процентах)

```
#include <iostream>
#include <string>
#include <map>
#include <fstream>

using namespace std;

int main()
	{
	map <string,int> words;

	ifstream in;
	in.open("in.txt");

	string word;
```

```
	while (in>>word)
	{
		words[word]++;
	}

	ofstream out;
	out.open("out.txt");

	int count = 0;

	map <string,int>::iterator cur;

	out<<"Words count:"<<endl;

	for (cur=words.begin();cur!=words.end();cur++)
	{
		out<<(*cur).first<<":
"<<(*cur).second<<endl;count+=(*cur).second;
	}

	out<<"Words percenc:"<<endl;

	for (cur=words.begin();cur!=words.end();cur++)
	{
		out<<(*cur).first<<":
"<<(float)((float)(*cur).second/(float)count)*100<<"%"<<endl;
	}

	return 0;
}
```