

关联容器 set 和 map

作者: [heyang5188](#)

原文链接: <https://ld246.com/article/1503384619371>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

```

map<string, size_t> word_count;
string word;
while (cin >> word&&word != "stop")
    ++word_count[word];
for (const auto &w : word_count)
    cout << w.first << "sssss :: " << w.second << endl;

map<string, vector<string>> word_count;
string word;
string name;
cout << "please input family :" << endl;
while (cin >> word&&word != "stop")
{
    cout << "pleas input family members :";
    while (cin >> name&&name != "ok")
        word_count[word].push_back(name);
    cout << "please input family :" << endl;
}
for (const auto &w : word_count)
{
    cout << "family name is :" <<w.first<< endl;
    cout << "family member is : ";
    for (const auto &v : w.second)
        cout << v << endl;
}

```

一个简单的实例 实现单词的转换

给定一个string，将他转换为另外一个string。程序是输入两个文件，第一个文件保存规则，第二个文件表示实际输入，第一个文件用来转换第二个文件中的文本。

```

const string &transform(const string &s,const map<string,string>&m)
{
    auto map_it = m.find(s); //查找出关键字位置，返回指向这个值的迭代器
    if (map_it != m.end())
        return map_it->second; //如果查询到了，直接替换。
    else
        return s; //can not find ,return s;
}
map<string, string> buildMap(ifstream &map_file)
{
    map<string, string> trans_map;
    string key;
    string value;
    while (map_file >> key&&getline(map_file, value)) {
        if (value.size() > 1)
            trans_map[key] = value.substr(1);
        else
            throw runtime_error("no rule for" + key);
    }
    return trans_map;
}

```

```

}
void word_transform(istream &map_file, istream &input) {
    auto trans_map = buildMap(map_file); //生成trans map
    string text;
    while (getline(input, text)) { //从input文件读取需要转换的string
        istream stream(text); //string 流
        string word;
        bool firstword = true;
        while (stream >> word) {
            if (firstword)
                firstword = false;
            else
                cout << " ";
            cout << transform(word, trans_map);
        }
        cout << endl;
    }
}
int main(){

    ifstream map_file("make_map"); // open transformation file
    if (!map_file) // check that open succeeded
        throw runtime_error("no transformation file");

    ifstream input("input"); // open file of text to transform
    if (!input) // check that open succeeded
        throw runtime_error("no input file");

    word_transform(map_file, input);

    system("pause");
    return 0;
}

```

multimap and multiset中的查找元素

三种方法。

方法一：

- 1.find找到一个关键字的第一个元素的迭代器
- 2.count找到这个元素在关联容器中的出现次数
- 3.while (元素个数) {输出元素; 迭代器后移; 计数值减一}

方法二：

- 1.lower-bound 和 upper-bound 函数查找一个迭代器范围;

方法三：

- 1.直接用equal-range 查询一个关键字, 返回一个pair 就是一个first代表起始迭代器, second代表最元素的迭代器;

特别的, 当这些查询都查询不到的时候, 那么返回的迭代器位置是end();并且相同。