## Exercise 6: Containers and Iterators

**1. Purpose and requirements**

**Purpose:** Familiar with Containers and Iterators, including: (1) sequence: vector,list; (2) associate: map, set; (3) STL

**Requirements:** simplicity, clarity, and user friendly

**2. Experiment contents：**

(1)

1)define an array of ints with the ten elements {0,1,2,3,4,5,6,7,8,9}.

2)define a vector<int> with those ten elements

3)define a list<int > with those ten elements

4)define a second array, vector, and list, each initialized as a copy of the first array, vector, and list, respectively.

5)increase the value of each element in the array by 2; increase the value of each element in the vector by 3; increase the value of each element in the list by 4.

6)write a simple copy() operation,

template<class Iter1, class Iter2> Iter2 copy(Iter1 f1, Iter1 e1, Iter2 f2);

That copies [f1,e1 ) to [f2, f2+(e1-f1)) and returns f2+(e1-f1) just like the standard library copy function. Note that if f1==e1 the sequence is empty, so that there is nothing to copy.

7) Use your copy() to copy the array into the vector and to copy the list into the array.

8) use the standard library find() to see if the vector contains 3, and print out the result; use find() to see if list contains 27, and print out the result; Note that if find() return the end of the sequence, the value wasn’t found.

Remember to test after each step!

1)定义一个包含10个元素{0,1,2,3,4,5,6,7,8,9}的int数组。

2)用这10个元素定义一个向量

3)用这10个元素定义一个列表

4)定义第二个数组、向量和列表，分别初始化为第一个数组、向量和列表的副本。

5)将数组中每个元素的值增加2;将向量中每个元素的值增加3;将列表中每个元素的值增加4。

6)写一个简单的copy()操作，

模板<类Iter1，类Iter2>迭代复制(迭代1 f1，迭代1 e1，迭代2 f2);

它将[f1,e1]复制到[f2, f2+(e1-f1))并返回f2+(e1-f1)，就像标准库复制函数一样。注意，如果f1==e1，序列是空的，因此没有什么可以复制的。

7)使用copy()将数组复制到向量中，并将列表复制到数组中。

8)使用标准库find()查看向量是否包含3，并打印出结果;使用find()查看列表是否包含27，并打印出结果;注意，如果find()返回序列的末尾，则没有找到该值。

记住在每一步之后都要进行测试!

(2)

1) Define a struct Iterm{string name; int id; double value; /\* …\*/}; and make a vector<Item> vi; and fill it with 5 items;

2) Sort vi by name.

3) sort vi by id.

4) Sort vi by value; print it in the order of decreasing value

5) insert Item(“horse shoe”, 99, 12.34) and Item(“Camera”, 9988, 499.5).

6) remove (erase) two Items identified by name from vi

7) remove (erase) two Items identified by id from vi

1)定义一个struct Iterm{string name;int id;双重价值;/ \*…\* /};令向量 vi;填上5项;

2)按名称对vi排序。

3)按id排序vi。

4)按值对vi排序;按值递减的顺序打印

5)插入Item(“horse shoe”，99,12.34)和Item(“Camera”，9988,499.5)。

6)从vi中删除(擦除)名称标识的两项

7)从vi中删除(擦除)id标识的两项

(3)

1) define a map<string, int> called msi

2) Insert five (name, value) pairs into it, using msi[“lecture”]=21 or msi.insert(make\_pair( “lecture”,21));

3) output the pairs in some format of your choice

4) erase the pairs from msi

5) write a function to read from cin and place them in msi

6) define a map<int, string> called mis

7) enter the values from msi into mis; that is if msi has an element(“lecture”, 21), mis should have an element (21,”lecture”)