简单的程序诠释C++ STL算法系列之十二：find_end

C++STL的非变易算法（Non-mutating algorithms）是一组不破坏操作数据的模板函数，用来对序列数据进行逐个处理、元素查找、子序列搜索、统计和匹配。

find_end算法在一个序列中搜索出最后一个与另一序列匹配的子序列。有如下两个函数原型，在迭代器区间[first1, last1)中搜索出与迭代器区间[first2, last2)元素匹配的子序列，返回首元素的迭代器或last1。

函数原型：

```
template<class ForwardIterator1, class ForwardIterator2>
   ForwardIterator1 find_end(
      ForwardIterator1 _First1,
      ForwardIterator1 _Last1,
      ForwardIterator2 _First2,
      ForwardIterator2 _Last2
   );
template<class ForwardIterator1, class ForwardIterator2, class Pr>
   ForwardIterator1 find_end(
      ForwardIterator1 _First1,
      ForwardIterator1 _Last1,
      ForwardIterator2 _First2,
      ForwardIterator2 _Last2,
      BinaryPredicate _Comp
   );
```

示例程序：

```
/*******************************************************************
 * Copyright (C) Jerry Jiang
 * File Name   : find_end.cpp
 * Author      : Jerry Jiang
 * Create Time : 2011-10-12 20:07:20
 * Mail        : jbiaojerry@gmail.com
 * Blog        : http://blog.csdn.net/jerryjbiao
 * Description :  简单的程序诠释C++ STL算法系列之十二
 *                非变易算法 : 最后一个子序列搜索 find_end
 ******************************************************************/

#include <algorithm>
#include <vector>
#include <iostream>

using namespace std;

int main()
{
 vector<int> v1;
 v1.push_back(5);
 v1.push_back(-2);
 v1.push_back(4);
 v1.push_back(3);
 v1.push_back(-2);
 v1.push_back(4);
 v1.push_back(8);
 v1.push_back(-2);
 v1.push_back(4);
 v1.push_back(9);

 vector<int>::const_iterator iter;
 cout << "v1: " ;
 for (iter = v1.begin(); iter != v1.end(); ++iter)
 {
  cout << *iter << "  ";
 }
 cout << endl;

 vector<int> v2;
 v2.push_back(-2);
 v2.push_back(4);

 cout << "v2: " ;
 for (iter = v2.begin(); iter != v2.end(); ++iter)
 {
  cout << *iter << "  ";
 }
 cout << endl;

 vector<int>::iterator iLoaction;
 iLoaction = find_end(v1.begin(), v1.end(), v2.begin(), v2.end());

 if (iLoaction != v1.end())
 {
  cout << "v1中找到最后一个匹配V2的子序列，起始位置在：" 
    << "v1[" << iLoaction - v1.begin() << "]" << endl;
 }

 return 0;
}
```

****

C++经典书目索引及资源下载：http://blog.csdn.net/jerryjbiao/article/details/7358796

****