

zjufirefly

个人资料



zjufirefly

恒

访问: 6401次

积分: 463

等级: BLOG 2

排名: 千里之外

原创: 42篇

转载: 4篇

译文: 0篇

评论: 1条

文章搜索

文章分类

C++ (14)

其他 (5)

链接 (13)

linux (13)

shell (7)

操作 (11)

文章存档

2014年12月 (8)

2014年11月 (7)

2014年10月 (8)

2014年09月 (6)

2014年08月 (4)

展开

阅读排行

git操作命令 (327)

blktrace (287)

mysql常用命令 (280)

python (270)

cgroups (262)

linux下如何配置openvpn (260)

2015年4月微软MVP申请

《京东技术解密》有奖试读, 礼品大放送

"我的2014"年度征文活动火爆开启

CSDN 2014博客之星

list中如何查找两个元素间的某个元素

分类: C++

2014-05-04 22:15

73人阅读

评论(0)

收藏

编辑

删除

迭代器

仿函数

[cpp]

C

```
01.  /* The following code example is taken from the book
02.   * "The C++ Standard Library - A Tutorial and Reference"
03.   * by Nicolai M. Josuttis, Addison-Wesley, 1999
04.   *
05.   * (C) Copyright Nicolai M. Josuttis 1999.
06.   * Permission to copy, use, modify, sell and distribute this software
07.   * is granted provided this copyright notice appears in all copies.
08.   * This software is provided "as is" without express or implied
09.   * warranty, and with no claim as to its suitability for any purpose.
10.   */
11.  #include <iostream>
12.  #include <list>
13.  #include <algorithm>
14.  using namespace std;
15.
16.  int main()
17.  {
18.      list<int> coll;
19.      list<int>::iterator pos;
20.
21.      // insert elements from 20 to 40
22.      for (int i=20; i<=40; ++i) {
23.          coll.push_back(i);
24.      }
25.
26.      /* find position of element with value 3
27.       * - there is none, so pos gets coll.end()
28.       */
29.      pos = find (coll.begin(), coll.end(),    // range
30.                 3);                          // value
31.
32.      /* reverse the order of elements between found element and the end
33.       * - because pos is coll.end() it reverses an empty range
34.       */
35.      reverse (pos, coll.end());
36.
37.      // find positions of values 25 and 35
38.      list<int>::iterator pos25, pos35;
39.      pos25 = find (coll.begin(), coll.end(), // range
40.                   25);                      // value
41.      pos35 = find (coll.begin(), coll.end(), // range
42.                   35);                      // value
43.
44.      /* print the maximum of the corresponding range
45.       * - note: including pos25 but excluding pos35
46.       */
47.      cout << "max: " << *max_element (pos25, pos35) << endl;
48.
49.      // process the elements including the last position
50.      cout << "max: " << *max_element (pos25, ++pos35) << endl;
51.  }
```

vim-ctags-taglist-netrw	(249)
性能测试命令字段解释	(213)
正则表达式	(211)
SystemTap	(195)

评论排行	
为什么需要auto_ptr_ref	(1)
valgrind内存检查	(0)
cgroups	(0)
linux下如何配置openvpn	(0)
git操作命令	(0)
thrift	(0)
Apache开源软件	(0)
vim-ctags-taglist-netrw	(0)
vim常用配置	(0)
shell按行读取字符串，并	(0)

推荐文章	
* 挣扎与彷徨——我的2014	
* 校招回忆录——小米篇	
* Android UI-自定义日历控件	
* 30岁程序员回顾人生、展望未来	
* 2014年终总结，我决定要实现的三个目标	
* Android 启动问题——黑屏 死机解决方法	

最新评论	
为什么需要auto_ptr_ref	
zjufirefly: explicit auto_ptr_ref(Tp1* __p): _M_ptr(__p){}...	

以上程序，数字按照顺序插入到链表之中，因此，pos25在pos35前，程序可正常运行。

如果随机插入这些数字，则执行*max_element (pos25, pos35)可能会导致未定义的行为，因为pos25可能在pos35之后。

解决方法1:

如果使用的是vector容器，vector容器的迭代器为随机迭代器，可以使用operator<进行比较。

```
[cpp]
01. if (pos25 < pos35) {
02.     max_element(pos25, pos35);
03. }
04. else if (pos 35 < pos25) {
05.     max_element(pos35, pos25);
06. }
07. else {
08.     // itr equal
09. }
```

解决方法2:

分段查找，先找到一个迭代器，以这个迭代器分界，分别在起点到这个迭代器，这个迭代器到终点进行查找另一个迭代器。代码如下:

```
[cpp]
01. pos25 = find(coll.begin(), coll.end(), 25);
02. pos35 = find(coll.begin(), pos25, 35);
03. if (pos35 != pos25) {
04.     // pos35在pos25前
05.     max_element(pos35, pos25);
06. }
07. else {
08.     pos35 = find(pos25, coll.end(), 35);
09.     if (pos35 != pos25) {
10.         // pos25在pos35前
11.         max_element(pos25, pos35);
12.     }
13. }
14. else {
15.     // itr equal, 没有找pos35
16. }
```

解决方案3:

```
[cpp]
01. pos = find_if (coll.begin(), coll.end(),
02.               compose_f_gx_hx(logical_or<bool>(),
03.                               bind2nd(equal_to<int>>(), 25),
04.                               bind2nd(equal_to<int>>(), 35)));
```

pos 为等于25或35的第一个迭代器。然后在pos和end之间查找另一个迭代器。

主题推荐

解决方案namespacealgorithmiteratoriostream

猜你在找

STL学习小结

STL学习小结

STL学习小结

C++ STL 学习总结

STL学习小结

STL学习小结

STL基础

STL学习小结

STL学习小结

STL学习小结

STL学习小结

1 博客搬家工具

2 发档口招商广告

3 期货贵金属

4 加盟面包房

5 北京婚姻中介

6 深圳面包加盟

7 水浒传街机

8 最新3d大型网

9 一号棋牌

10 冲压不锈钢水箱

11 ewin棋牌官网

12 外贸代加工

13 个税查询

14 小型电动推杆

15 稠州游戏中心

16 欧洲买房移民

17 绿之韵官方网站

18 水浒传老虎机

19 巴奴火锅加盟

20 1号游戏大厅

查看评论

暂无评论

发表评论

用户名:zjufirefly

评论内容:

提交

* 以上用户言论只代表其个人观点，不代表CSDN网站的观点或立场

核心技术类目

全部主题

Hadoop

AWS

移动游戏

Java

Android

iOS

Swift

智能硬件

Docker

OpenStack

VPN

Spark

ERP

IE10

Eclipse

CRM

JavaScript

数据库

Ubuntu

NFC

WAP

jQuery

BI

HTML5

Spring

Apache

.NET

API

HTML

SDK

IIS

Fedora

XML

LBS

Unity

Splashtop

UML

components

Windows Mobile

Rails

QEMU

KDE

Cassandra

CloudStack

FTC

coremail

OPhone

CouchBase

云计算

iOS6

Rackspace

Web App

SpringSide

Maemo

Compuware

大数据

aptech

Perl

Tornado

Ruby

Hibernate

ThinkPHP

HBase

Pure

Solr

Angular

Cloud Foundry

Redis

Scala

Django

Bootstrap