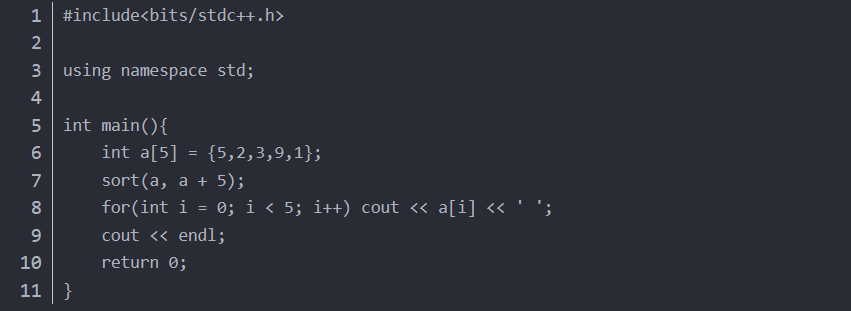
# STL函数库

1. **std::sort**：用于对容器中的元素进行排序。
2. **std::find**：用于在容器中查找特定元素。
3. **std::lower\_bound**：用于在有序容器中查找第一个不小于给定值的元素。
4. **std::upper\_bound**：用于在有序容器中查找第一个大于给定值的元素。
5. **std::transform**：用于对容器中的元素进行某种操作并将结果存储到另一个容器中。
6. **std::accumulate**：对容器中的元素进行累加操作。
7. **std::reverse**：对容器中的元素进行逆序操作。
8. **std::copy**：将一个容器中的元素复制到另一个容器中。
9. **std::count**：计算容器中特定值出现的次数。
10. **std::for\_each**：对容器中的每个元素执行指定的操作

**1.sort( ) 快排**

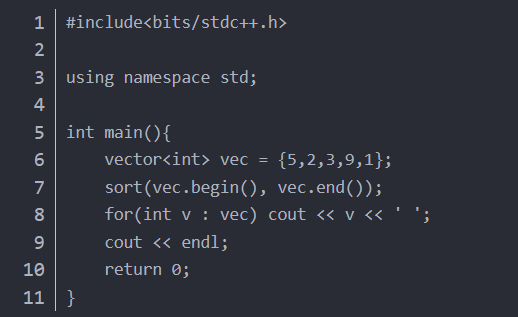
函数原型：sort(起始地址, 末尾地址, cmp),其中cmp是可以自己定义的函数名

* **数组排序**



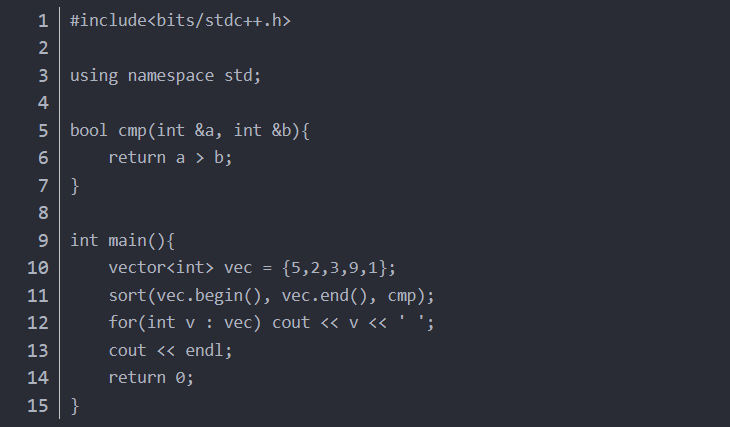
输出结果：1 2 3 4 5 9；

* **Vector排序**



输出：·1 2 3 5 9；

* **自定义cmp；**

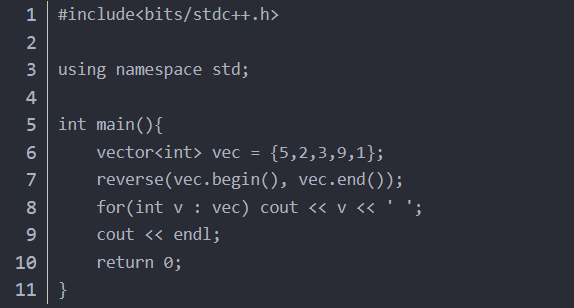
****

输出：1 2 3 5 9；

**2.Reverse()逆置**

函数原型：reverse(起始地址, 末尾地址);

**vector示例**

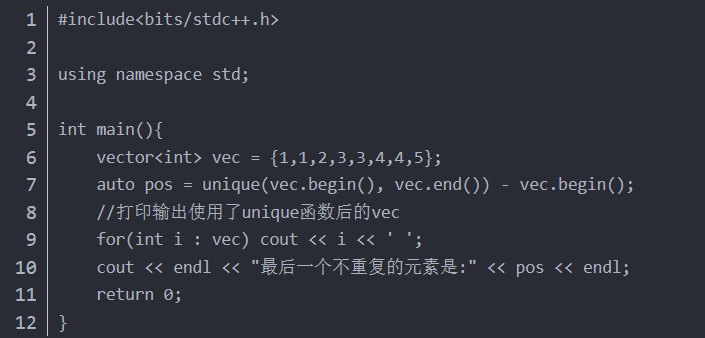
****

**输出：1 9 3 2 5**

**3.unique()去重**

函数原型：unique(起始地址, 末尾地址, fun);其中fun为自定义的函数名。

注意：unique函数去重并不是真正的去重，它是不断的将**后面不重复的元素覆盖前面重复的元素**，最后返回最后一个不重复的元素的地址。

****

**输出：1 2 3 4 5 4 4 5**

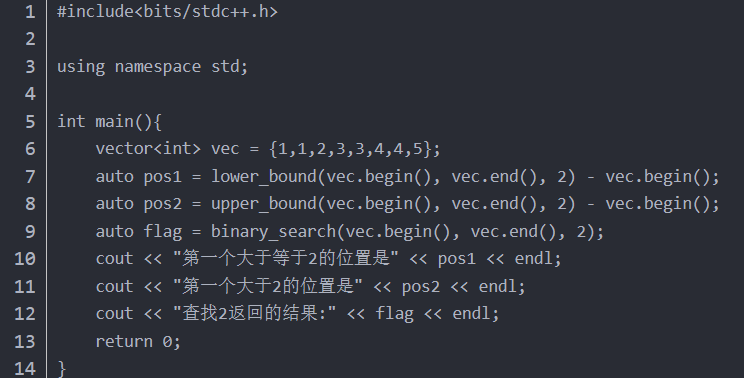
**4.查找函数**

* **二分查找**

lower\_bound(起始地址, 末尾地址, target)：查找第一个大于等于target目标值的位置

upper\_bound(起始地址, 末尾地址, target)：查找第一个大于target目标值的位置

binary\_search(起始地址, 末尾地址, target)：查找target是否存在于数组或vector中，找到返回true，否则返回false



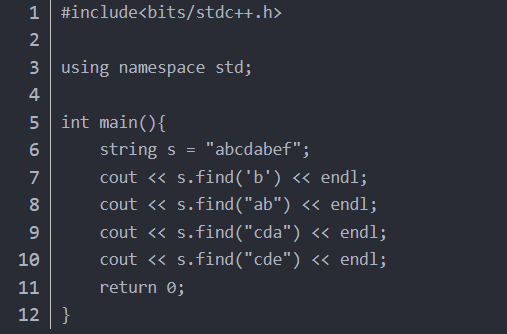
输出：第一个大于等于2的位置是2

第一个大于2的位置是3

查找2返回的结果:1

* **字符串查找**

s1.find(s2)：在s1字符串中查找s2，查找到返回第一个字符的位置，查找失败返回s1.npos，这个其实是一个特别标志，也可以看成一个数字，是4294967295，即s1.npos=4294967295



输出：1

0

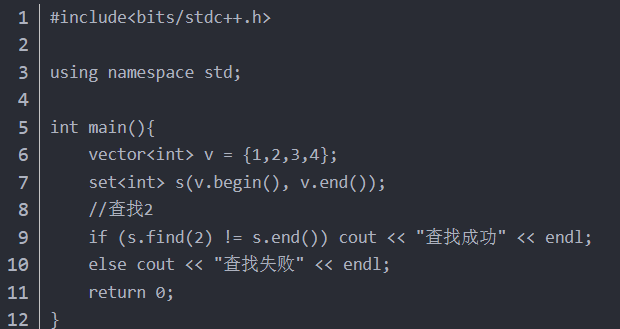
2

4294967295

* **Set集合查找**

set.find(a)：查找a是否在set中，如果找不到，返回set.end()

set.count(a)：本来是计算a出现的次数，但是由于集合中是没有重复元素的，于是count函数也就被作为查找函数了，因为a只能出现1次或者0次，查找成功，返回1；查找失败返回0.



输出：查找成功

* **Map映射查找**

map.find()：主要用于查找key是否存在map中，不存在返回map.end()，用法和set一样

**5.字符串整形转换函数**

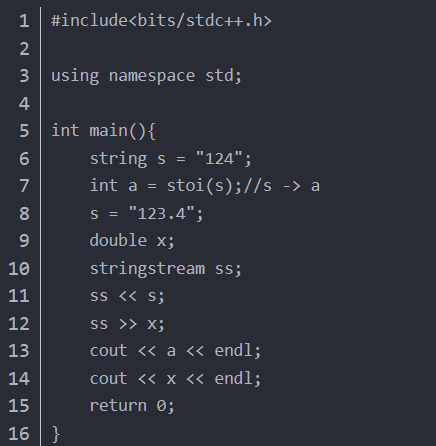
* **字符串转整形**

stoi(s)：将字符串s转化成整形，s为string类型，即string --> int

atoi(s)：将字符串转化为整形，但s为const char\*类型，可以先用s.c\_str()方法把string类型转化为const char\*类型，再转为整形，即const char --> int

stringstream：需要头文件#include<sstream>,可将只含数字的字符串转化为整形，也可将数字和小数点组成的字符串转化为浮点型，即string --> int, string --> double

示例：



**输出：**124

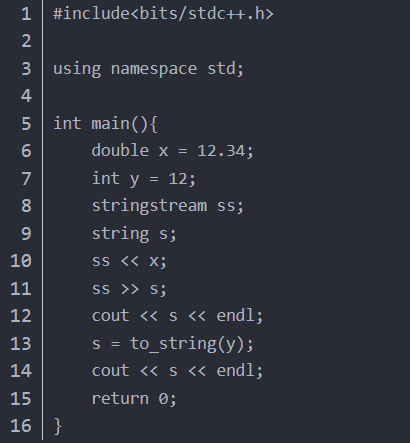
123.4

* **整形转字符串**

stringstream：需要头文件#include<sstream>，可将整形或浮点型数字转化为字符串，即int --> string, double --> string

to\_string()：可将整形转化为字符串，不推荐将浮点型转化为字符串

**示例：**

****

**输出：12.34**

**12**