**l课下学习1 vector 容器**

vector向量容器这种数据结构和数组非常相似，也称为单端数组， 不同之处在于数组是静态空间，而 vector 可以动态扩展，可以胜任顺序表操作，所以我们可以使用vector来实现顺序表。

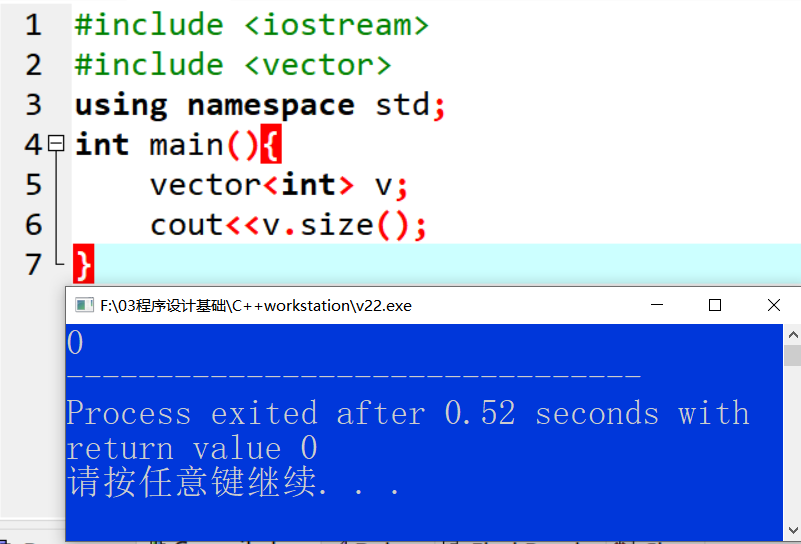
1.创建vector对象

语法：vector<元素类型> 容器名称（容器大小，初始值）；

（1）创建时不指定容器大小

例如：vector<int> v;

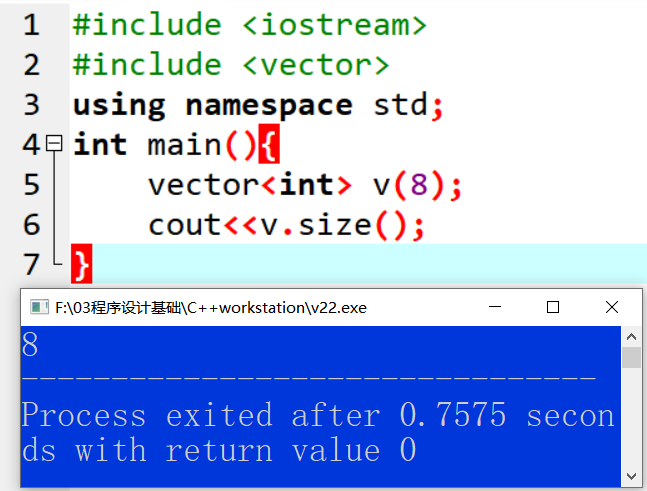
创建了一个名称为v的向量容器（顺序表），存储空间为0；



（2）创建时指定容器大小

例如：vector<int> v（8）;

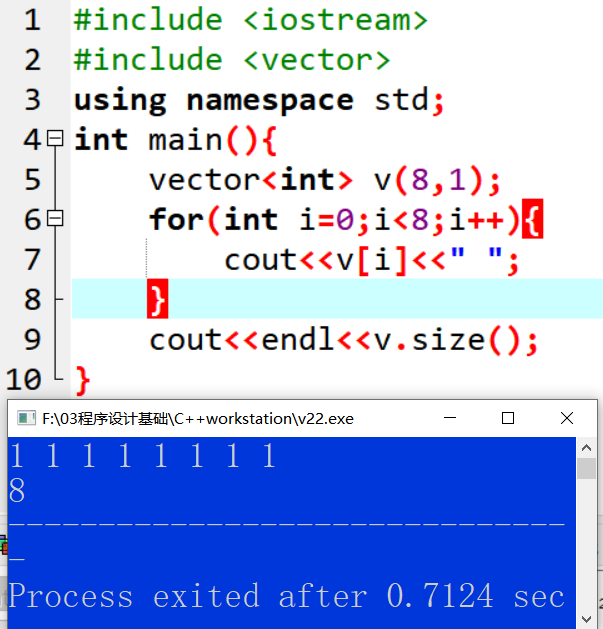
创建了一个名称为v的向量容器（顺序表），存储空间为8；



（3）创建时指定容器大小并设置初始值

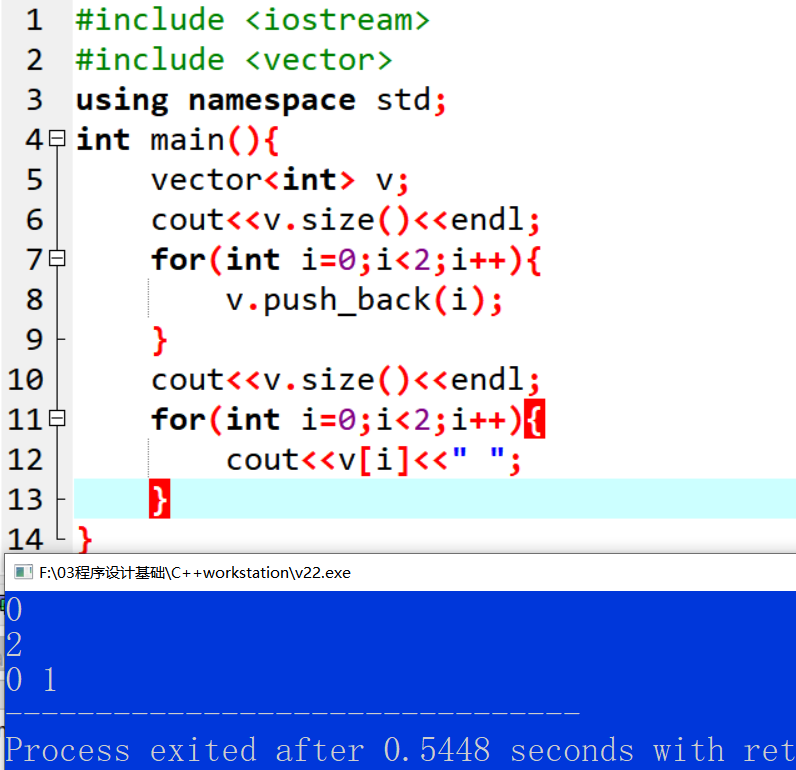
例如：vector<int> v（8，1）;

创建了一个名称为v的向量容器（顺序表），存储空间为8，每个元素的初始值为1；



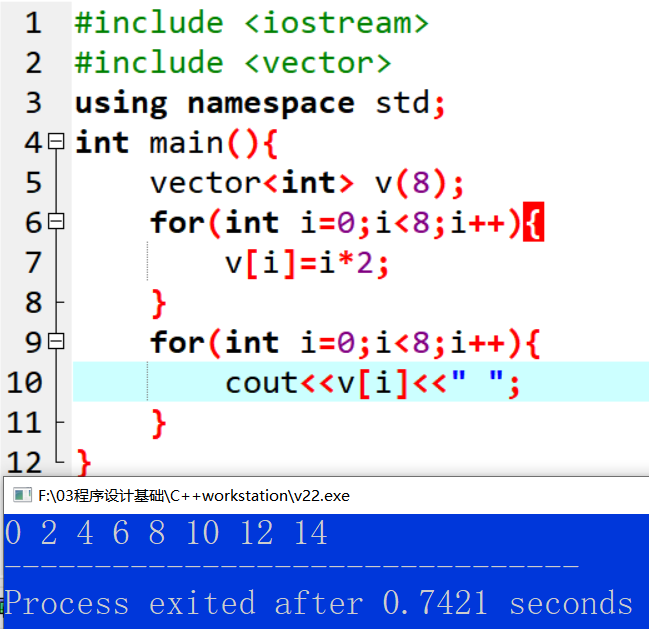
2.在尾部插入元素

vector可以通过使用push\_back()实现在尾部追加元素，vector会自动分配新的存储空间，用以实现尾部扩张。



3.元素访问

（1）下标访问：访问vector向量中的元素可以类似于数组的访问形式，一般格式为：向量名[下标]



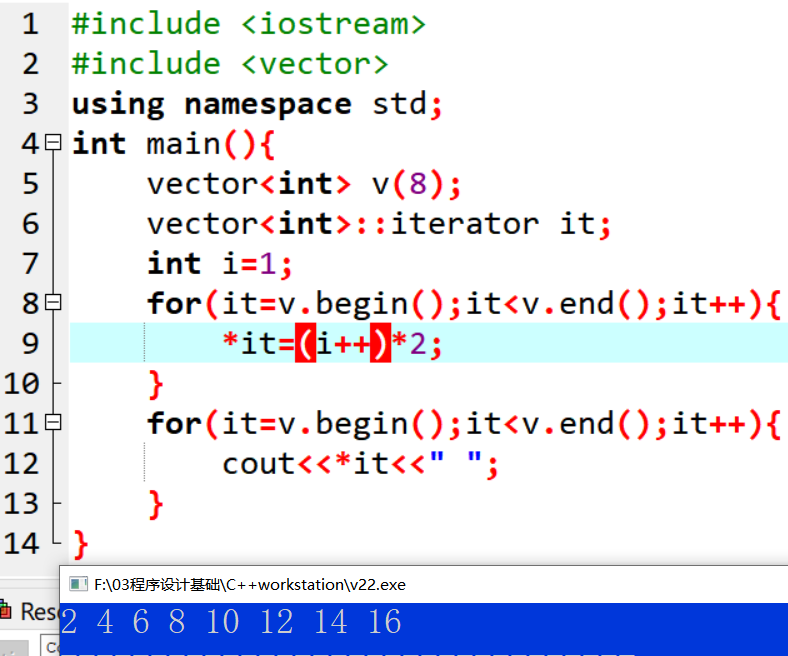
（2）使用迭代器访问

迭代器是一种检查容器内元素并遍历元素的数据类型，从而不使用下标完成容器元素的访问，迭代器定义格式为：

容器类型<容器元素类型>::iterator 迭代器名称；

例如：vector<int>::iterator it;

即，定义一个作用于整型vector上的，名为it的迭代器，这样可以使用it来遍历或访问vector容器中的元素。



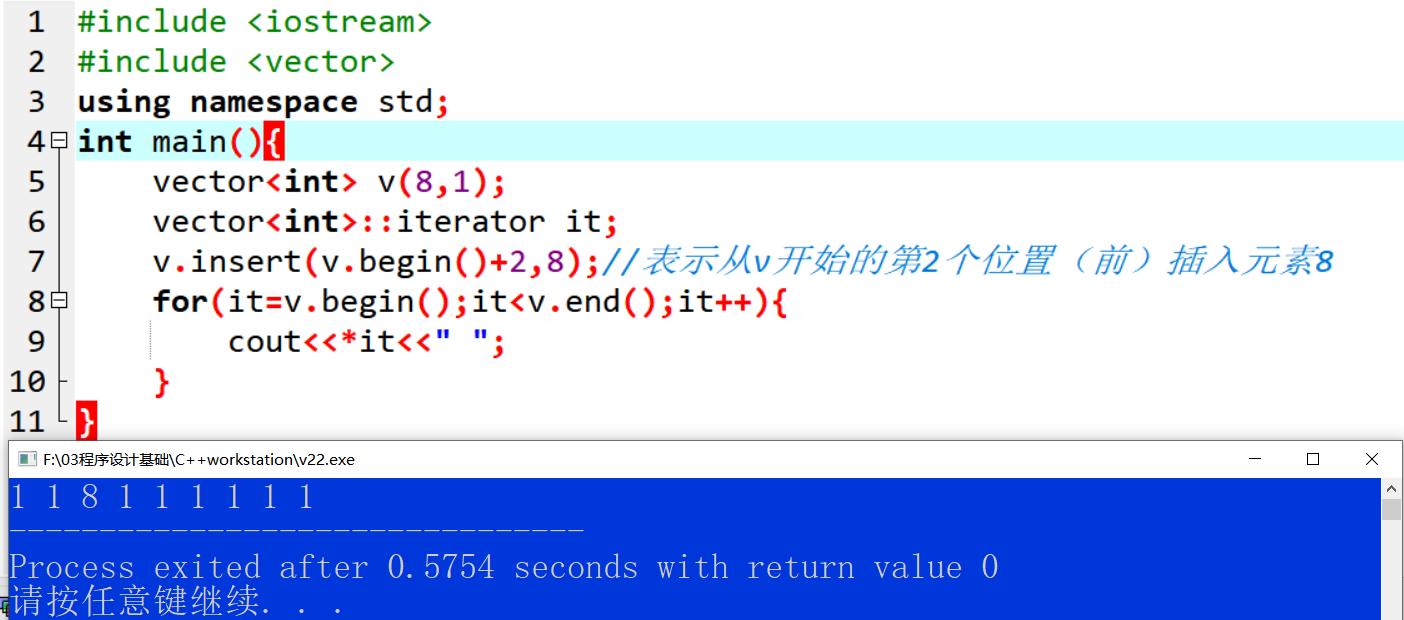
其中v.begin()表示第一个元素的位置（地址），v.end()表示最后一个元素的位置，此时的迭代器更像一个指针。

4.插入元素

vector中的insert（）方法能够实现元素的插入操作，此时vector长度会自动扩张，并自动实现数据的移动。

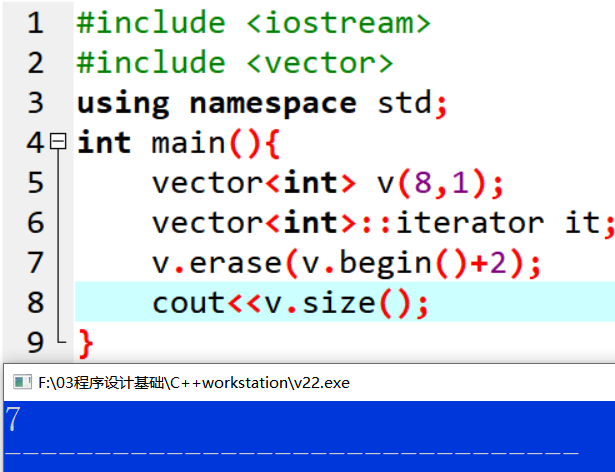
语法：迭代器名称.insert（插入位置，元素）

注意此时插入位置不能是下标，只能是位置或迭代器，所以希望在第i个元素前插入元素必须使用v.begin（）+i求得第i个元素位置。

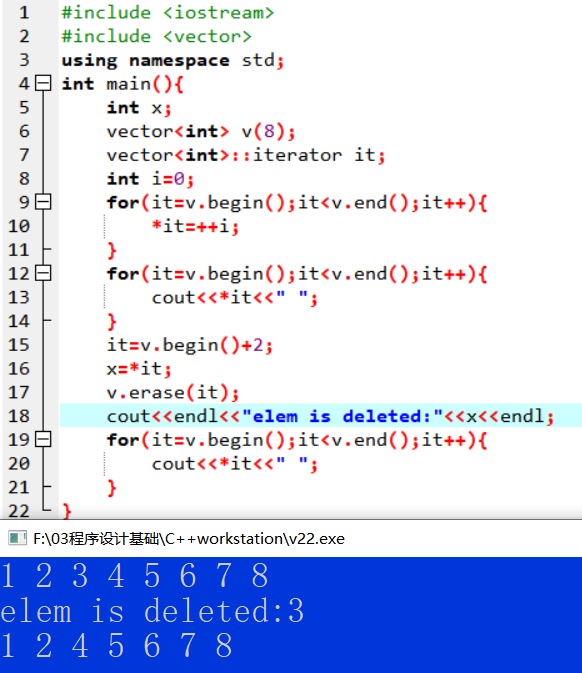


5.删除元素

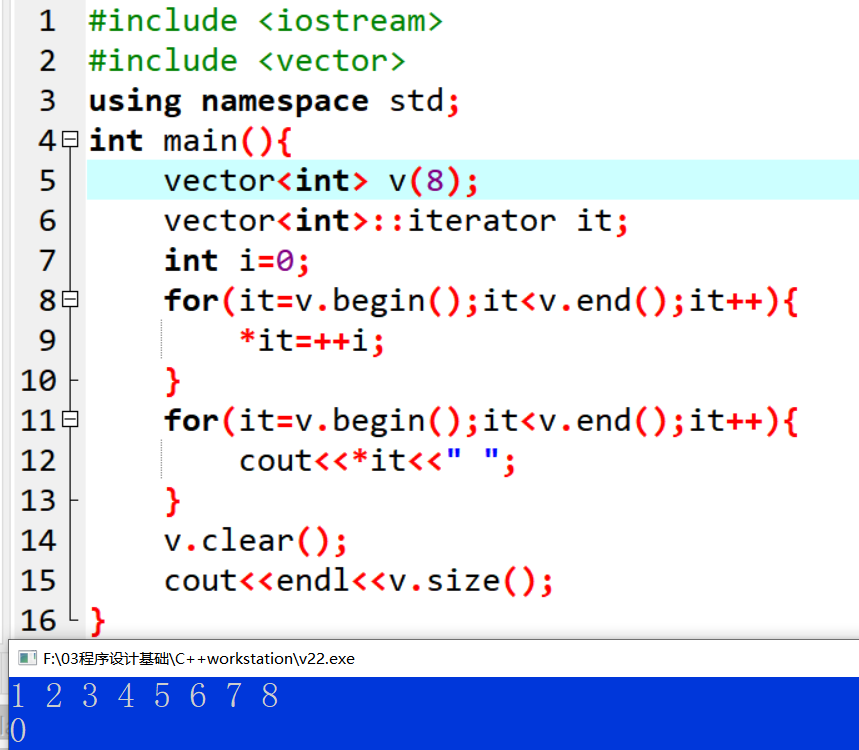
vector中的erase（）方法能够实现元素的删除操作，此时vector长度会自动收缩，并自动实现数据的移动。



注意erase（）函数只能删除元素，不能带回被删除元素，所以需要对被删除元素进行处理时，一定先将被删除元素取出，然后进行删除（erase的返回值是被删除元素的下一个元素的地址）



vector中的clear（）方法能够实现向量的清空，此时将向量的长度设为了0，与顺序表中的clear方法不同。



6.向量反转

reverse（）方法能够将Vector向量中的部分或全部元素反转，但此时需要包含algorithm头文件。语法：reverse（向量的反转起点位置，向量反转结束点位置）

#include <iostream>

#include <vector>

#include <algorithm>

#include <cstdlib>

#include <ctime>

using namespace std;

int main(){

srand(NULL);//以空为随机种子

vector<int> v(10);

vector<int>::iterator it;

int i=0;

for(it=v.begin();it<v.end();it++){

\*it=rand()%100;//rand根据随机种子产生随机数，此时仅取最后两位，因为随机种子为空，每次运行的随机数其实是相同的，如果真正产生不同的随机数需要使用系统时间作为随机种子，任何两个时刻的时间不同，产生的随机数不同，课下自己查阅资料实现

}

for(it=v.begin();it<v.end();it++){

cout<<\*it<<" ";

}

reverse(v.begin(),v.end());

cout<<endl;

for(it=v.begin();it<v.end();it++){

cout<<\*it<<" ";

}

}

7.排序

可以使用algorithm中的sort（）实现对vector向量对象制定范围排序，语法：

sort（排序范围起点位置，排序范围结束点位置），如何实现递减排序上学期课程已经教学过，请自己回顾上学期课程。