在c++中，vector是一个十分有用的容器，下面对这个容器做一下总结。

1 基本操作

(1)头文件#include<vector>.

(2)创建vector对象，vector<int> vec;

(3)尾部插入数字：vec.push\_back(a);

(4)使用下标访问元素，cout<<vec[0]<<endl;记住下标是从0开始的。

(5)使用迭代器访问元素.

vector<int>::iterator it;

for(it=vec.begin();it!=vec.end();it++)

cout<<\*it<<endl;

(6)插入元素：    vec.insert(vec.begin()+i,a);在第i+1个元素前面插入a;

(7)删除元素：    vec.erase(vec.begin()+2);删除第3个元素

vec.erase(vec.begin()+i,vec.end()+j);删除区间[i,j-1];区间从0开始

(8)向量大小:vec.size();

(9)清空:vec.clear();

2

vector的元素不仅仅可以使int,double,string,还可以是结构体，但是要注意：结构体要定义为全局的，否则会出错。下面是一段简短的程序代码：

[复制代码](javascript:void(0);)

#include<stdio.h>

#include<algorithm>

#include<vector>

#include<iostream>

using namespace std;

typedef struct rect

{

int id;

int length;

int width;  
  
　　//对于向量元素是结构体的，可在结构体内部定义比较函数，下面按照id,length,width升序排序。  
　　bool operator< (const rect &a)  const  
    {  
        if(id!=a.id)  
            return id<a.id;  
        else  
        {  
            if(length!=a.length)  
                return length<a.length;  
            else  
                return width<a.width;  
        }  
    }

}Rect;

int main()

{

vector<Rect> vec;

Rect rect;

rect.id=1;

rect.length=2;

rect.width=3;

vec.push\_back(rect);

vector<Rect>::iterator it=vec.begin();

cout<<(\*it).id<<' '<<(\*it).length<<' '<<(\*it).width<<endl;

return 0;

}

[复制代码](javascript:void(0);)

 3  算法

(1) 使用reverse将元素翻转：需要头文件#include<algorithm>

reverse(vec.begin(),vec.end());将元素翻转(在vector中，如果一个函数中需要两个迭代器，

一般后一个都不包含.)

(2)使用sort排序：需要头文件#include<algorithm>，

sort(vec.begin(),vec.end());(默认是按升序排列,即从小到大).

可以通过重写排序比较函数按照降序比较，如下：

定义排序比较函数：

bool Comp(const int &a,const int &b)  
{  
    return a>b;  
}  
调用时:sort(vec.begin(),vec.end(),Comp)，这样就降序排序。