vector 是向量类型，它可以容纳许多类型的数据，如若干个整数，所以称其为容器。vector 是C++ STL的一个重要成员，使用它时需要包含头文件：#include<vector>;

**vector 容器的长度不固定，能够在程序运行时动态地改变。**

一、vector 的初始化：可以有五种方式,举例说明如下：

（1） vector<int> a(10); //定义了10个整型元素的向量（尖括号中为元素类型名，它可以是任何合法的数据类型），但没有给出初值，其值是不确定的。  
（2）vector<int> a(10,1); //定义了10个整型元素的向量,且给出每个元素的初值为1  
（3）vector<int> a(b); //用b向量来创建a向量，整体复制性赋值  
（4）vector<int> a(b.begin(),b.begin+3); //定义了a值为b中第0个到第2个（共3个）元素  
（5）int b[7]={1,2,3,4,5,9,8};vector<int> a(b,b+7); //从数组中获得初值

二、vector对象的几个重要操作，举例说明如下：

（1）a.assign(b.begin(), b.begin()+3); //b为向量，将b的0~2个元素构成的向量赋给a  
（2）a.assign(4,2); //是a只含4个元素，且每个元素为2  
（3）a.back(); //返回a的最后一个元素  
（4）a.front(); //返回a的第一个元素  
（5）a[i]; //返回a的第i个元素，当且仅当a[i]存在  
（6）a.clear(); //清空a中的元素  
（7）a.empty(); //判断a是否为空，空则返回ture,不空则返回false  
（8）a.pop\_back(); //删除a向量的最后一个元素  
（9）a.erase(a.begin()+1,a.begin()+3); //删除a中第1个（从第0个算起）到第2个元素，也就是说删除的元素从a.begin()+1算起（包括它）一直到a.begin()+ 3（不包括它）  
（10）a.push\_back(5); //在a的最后一个向量后插入一个元素，其值为5

     a.push\_front(5);//在a的最前一个向量前插入一个元素，其值为5  
（11）a.insert(a.begin()+1,5); //在a的第1个元素（从第0个算起）的位置插入数值5，如a为1,2,3,4，插入元素后为1,5,2,3,4  
（12）a.insert(a.begin()+1,3,5); //在a的第1个元素（从第0个算起）的位置插入3个数，其值都为5  
（13）a.insert(a.begin()+1,b+3,b+6); //b为数组，在a的第1个元素（从第0个算起）的位置插入b的第3个元素到第5个元素（不包括b+6），如b为1,2,3,4,5,9,8 ，插入元素后为1,4,5,9,2,3,4,5,9,8  
（14）a.size(); //返回a中元素的个数；  
（15）a.capacity(); //返回a在内存中总共可以容纳的元素个数  
（16）a.rezize(10); //将a的现有元素个数调至10个，多则删，少则补，其值随机  
（17）a.rezize(10,2); //将a的现有元素个数调至10个，多则删，少则补，其值为2  
（18）a.reserve(100); //将a的容量（capacity）扩充至100，也就是说现在测试a.capacity();的时候返回值是100.这种操作只有在需要给a添加大量数据的时候才 显得有意义，因为这将避免内存多次容量扩充操作（当a的容量不足时电脑会自动扩容，当然这必然降低性能）   
（19）a.swap(b); //b为向量，将a中的元素和b中的元素进行整体性交换  
（20）a==b; //b为向量，向量的比较操作还有!=,>=,<=,>,<

（21） a.pop\_front();//从容器的首部移走数据；

  (22)   a.pop\_back();//从容器的尾部移走数据；

三、顺序访问vector的几种方式，举例说明如下：

1、向向量a中添加元素

复制代码代码如下:

vector<int> a;  
for(int i=0;i<10;i++)  
a.push\_back(i);

2、也可以从数组中选择元素向向量中添加

复制代码代码如下:

int a[6]={1,2,3,4,5,6};  
vector<int> b；  
for(int i=1;i<=4;i++)  
b.push\_back(a[i]);

3、也可以从现有向量中选择元素向向量中添加

复制代码代码如下:

int a[6]={1,2,3,4,5,6};  
vector<int> b;  
vector<int> c(a,a+4);  
for(vector<int>::iterator it=c.begin();it<c.end();it++)  
b.push\_back(\*it);

4、也可以从文件中读取元素向向量中添加

复制代码代码如下:

ifstream in("data.txt");  
vector<int> a;  
for(int i; in>>i)  
a.push\_back(i);

5、【误区】

复制代码代码如下:

vector<int> a;  
for(int i=0;i<10;i++)  
a[i]=i;  
//这种做法以及类似的做法都是错误的。刚开始我也犯过这种错误，后来发现，下标只能用于获取已存在的元素，而现在的a[i]还是空的对象

（2）从向量中读取元素  
1、通过下标方式读取

复制代码代码如下:

int a[6]={1,2,3,4,5,6};  
vector<int> b(a,a+4);  
for(int i=0;i<=b.size()-1;i++)  
cout<<b[i]<<" ";

2、通过遍历器方式读取

复制代码代码如下:

int a[6]={1,2,3,4,5,6};  
vector<int> b(a,a+4);  
for(vector<int>::iterator it=b.begin();it!=b.end();it++)  
cout<<\*it<<" ";

四、几种重要的算法，使用时需要包含头文件：

复制代码代码如下:

#include<algorithm>

（1）sort(a.begin(),a.end()); //对a中的从a.begin()（包括它）到a.end()（不包括它）的元素进行从小到大排列  
（2）reverse(a.begin(),a.end()); //对a中的从a.begin()（包括它）到a.end()（不包括它）的元素倒置，但不排列，如a中元素为1,3,2,4,倒置后为4,2,3,1  
（3）copy(a.begin(),a.end(),b.begin()+1); //把a中的从a.begin()（包括它）到a.end()（不包括它）的元素复制到b中，从b.begin()+1的位置（包括它）开始复制，覆盖掉原有元素  
（4）find(a.begin(),a.end(),10); //在a中的从a.begin()（包括它）到a.end()（不包括它）的元素中查找10，若存在返回其在向量中的位置