**暨南大学本科实验报告专用纸**

课程名称 C++程序设计 成绩评定

实验项目名称 vector容器及其迭代器应用 指导教师 王勇杰

实验项目编号 1 实验项目类型 编程型 实验地点 C105

学生姓名 李炜鹏 学号 2017052544

学院 智能科学与工程学院 系 计算机 专业 信息安全

实验时间2019年12月10日 午～12月13日 午 温度 ℃湿度

1. **实验目的**
2. 理解vector容器及其迭代器的概念
3. 上机调试相关的练习题代码，熟练vector容器及其迭代器的使用
4. **实验原理**

vector是C++标准模板库里面重要的容器，通过使用“vector<数据类型> 容器名;”可以声明一个容器变量用于存储数据，而数据的写入由vector模板类的成员函数push\_back(数据变量)完成（容器名.push\_back(数据变量)），而数据的读取则由模板类对应迭代器遍历容器并且读出（定义迭代器：vector<数据类型>::iterator 迭代器名称），通过\*运算符即可读出对应数据。

1. **主要仪器设备**

**仪器：**计算机

**实验环境：**Visual Studio 2017 或 Dev-C++

1. **源程序**

写出程序的源程序，并且注释代码中每条语句

#include <string>

using std::string;//引入std命名空间的string类

#include <vector>

using std::vector;//引入std命名空间的vector容器

#include <iostream>

using std::cin; using std::cout; using std::endl;//引入标准输入输出流和换行标识符

int main()

{

vector<unsigned> grades;//存输入的成绩，数据类型为unsigned(无符号整型)

// 学生成绩有以下11个数量级:

// 0--9, 10--19, . .. 90--99, 100

//本代码的功能是：计算每个成绩数量级的学生人数

vector<unsigned> scores(11, 0); //定义一个存unsigned类型数据的vector容器变量scores，初始化11个值为0的数据

unsigned grade;

while (cin >> grade) //从标准输入流中逐个读取grade，直到遇到输入结束标识符（Ctrl+Z）为止

{

if (grade <= 100) //grades容器只接收有效的成绩数据（小于或等于100）

grades.push\_back(grade);

++scores[grade / 10]; //该成绩数量级的学生数加一（这里有一个小问题，这样写会导致超过100的成绩也被记录到scores容器里面）

}

cout << "grades.size = " << grades.size() << endl;//输出grades容器的容器大小

for (vector<unsigned>::const\_iterator it = grades.begin();it != grades.end(); ++it)//使用迭代器循环整个grades容器

cout << \*it << " ";//iter指向容器中的当前位置的数据，\*iter为数据的值

cout << endl;

cout << "scores.size = " << scores.size() << endl;//输出scores容器的容器大小

for (vector<unsigned>::const\_iterator it = scores.begin();it != scores.end(); ++it)//使用迭代器循环整个scores容器

cout << \*it << " ";//iter指向容器中的当前位置的数据，\*iter为数据的值

cout << endl;

vector<unsigned> alt\_scores(11, 0);//再定义一个存unsigned类型数据的vector容器变量alt\_scores，初始化11个值为0的数据

for (vector<unsigned>::const\_iterator it = grades.begin();it != grades.end(); ++it)

{

unsigned i = \*it; //将it指向的数据赋值给i变量

++(\*(alt\_scores.begin() + i / 10));//该成绩数量级的学生数加一，这里可以这么理解：

alt\_scores.begin()指向alt\_scores容器的第一个数据， i/10为成绩的数量级，alt\_scores.begin()往后移i/10个单位恰好为该数量级学生人数存储的位置，将这个指针用小括号括起来加上\*号则为该指针指向的数据，这时候再将其括起来使用++运算，则可以累计该成绩数量级的人数了

}

cout << "alt\_scores.size = " << alt\_scores.size() << endl;//输出alt\_scores容器的大小

for (vector<unsigned>::const\_iterator it = alt\_scores.begin();it != alt\_scores.end(); ++it)

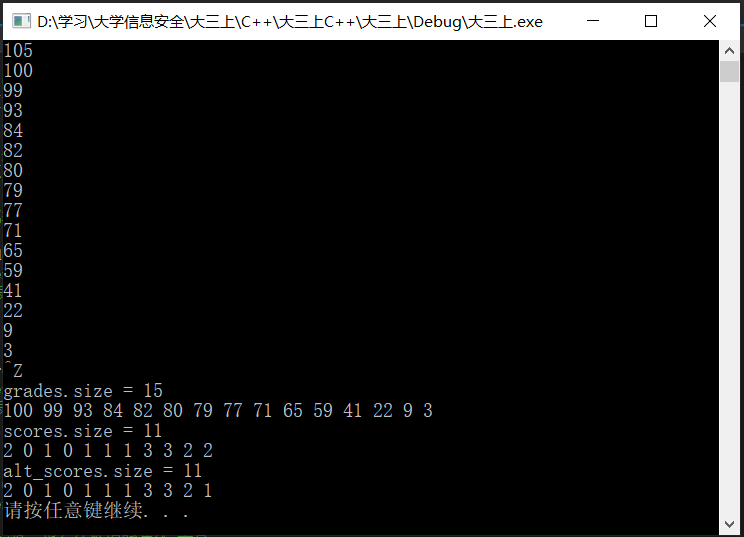
cout << \*it << " ";//输出每个成绩数量级的学生人数

cout << endl;

system("pause");//Visual Studio 2017需要加上这一句才能将程序停下来让编程人员查看运行结果

return 0;

}



这里分别输入了：105,100,99,93,84,82,80,79,77,71,65,59,41,22,9,3，可以看出，scores容器把105也算进去了，而由于alt\_scores容器统计的是grades容器的成绩数据，grades容器把不符合grade<=100的数据筛选掉了，并没有把105算进去，于是grades容器和alt\_scores容器的计算更加准确。

1. **实验评估**

通过本实验，我理解了C++标准模板库（STL）重要模板类容器vector的原理和用法，并且通过上机实现代码，熟悉了vector的读写操作及基本语法，深入理解了迭代器和vector容器的关系。