目录

[第一章 开始 2](#_Toc6352172)

[练习1.3 练习1.4 练习1.5 2](#_Toc6352173)

[练习1.6 3](#_Toc6352174)

[1.4.1节练习 3](#_Toc6352175)

[1.4.2节练习 4](#_Toc6352176)

[1.4.4 练习 6](#_Toc6352177)

[1.4.4节练习 7](#_Toc6352178)

[1.5节练习 8](#_Toc6352179)

[第一章小结 10](#_Toc6352180)

[第2章 变量和基本类型 10](#_Toc6352181)

[练习2.3 10](#_Toc6352182)

[练习2.40 练习2.41 练习2.42 10](#_Toc6352183)

[第二章小结 13](#_Toc6352184)

[第3章 13](#_Toc6352185)

[练习3.2 13](#_Toc6352186)

[练习3.2.3 14](#_Toc6352187)

[练习3.3.2 15](#_Toc6352188)

[练习3.3.3 16](#_Toc6352189)

[练习3.4.1 19](#_Toc6352190)

[练习3.4.2 21](#_Toc6352191)

[练习3.5.2 23](#_Toc6352192)

[练习3.5.3 24](#_Toc6352193)

[练习3.5.5 26](#_Toc6352194)

[第三章小结 28](#_Toc6352195)

# 第一章 开始

/\*\*\*

## 练习1.3 练习1.4 练习1.5

2019年3月22日22:26:13

by PK

\*\*\*\*/

#include <iostream>

#include <stdio.h>

int main(void)

{

/\*练习1.3

输出hello world

\*/

printf("hello C hello world! \r\n"); //C

std::cout << "hello C++ hello singledog! \r\n"<<std::endl;

//以上为练习1.3

/\*\* 练习1.4

计算两个输入数的加和乘

\*\*/

int num1,num2,res;

std::cout<<"please input two number \r\n"<<std::endl;

std::cin>>num1>>num2;

res = num1+num2;

std::cout<<"num1+num2 = "<<res<<std::endl;

res = num1\*num2;

std::cout<<"num1\*num2 = "<<res<<std::endl;

printf("\r\n\r\n");

/\*\*练习1.5

将所有的输出操作放在一条语句中

\*\*/

std::cout<<"the sum of num1 + num2 is "<<num1+num2<<" and result of num1\*num2 is "<< num1\*num2<<"\r\n"<<std::endl;

getchar();

while(1);

return 0;

}

## 练习1.6

程序段是不合法的，因为<<运算符的用法错误，<<左侧为流对象，右侧为流输出内容，流输出可以有多个对象，但是不能用分号将其隔开，否则编译器会认为语句已经结束

修改：

代码修改为

Std::cout<<”The sum of ”<<v1<<”and”<<v2<<”is”<<v1+v2<<std::endl;

程序输出v1 和v2的和及对应字符串

/\*\*\*\*\*\*\*\*\*\*88

\*\*\* 2019年3月24日14:55:32

## 1.4.1节练习

while语句实现循环控制

练习1.9

练习1.0

练习1.11

\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <iostream>

int main(void)

{

//练习1.9 求50到100的循环加和

int num = 50,sum = 0;

while(num<101)

{

sum += num;

num++;

}

std::cout<<"sum = "<<sum<<"\r\n"<<std::endl;//结果为3825

//练习1.10 递减

num = 10;

while(num>=0)

{

std::cout<<" "<<num<<" "<<std::endl;

num--;

}

//练习1.11 输入2个整数，输出二者之间的所有整数

int innum1,innum2;

std::cout<<" please input two increasing integer"<<std::endl;

std::cin>>innum1>>innum2;

if(innum1 > innum2)

{std::cout<<" input error! \r\n please input two increasing integer!"<<std::endl;}

else

{

num = innum1;

while(num<innum2)

{

std::cout<<num<<std::endl;

num++;

}

}

while(1);

return 0;

}

/\*\*\*\*\*\*\*\*\*\*88

\*\*\*

2019年3月24日15:23:18

## 1.4.2节练习

for循环控制流练习

练习1.12

练习1.13

练习1.14

\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <iostream>

int main(void)

{

/\*\*\*

练习1.12

描述for循环的功能，并求sum值

\*\*\*/

int sum = 0;

for(int i = -100;i<=100;++i)

sum+=i;

std::cout<<"sum = "<<sum<<"\r\n"<<std::endl;

/\*\*\*\*

\*\* 功能：循环求整数-100到100的和 sum输出值 0

\*\*\*\*/

//练习1.13

//练习1.13.1 求50到100的循环加和 for实现

sum = 0;

for(int i=50;i<101;i++)

{

sum += i;

}

std::cout<<"sum of 50 to 100 inclusive is "<<sum<<"\r\n"<<std::endl;//结果为3825

//练习1.13.2 递减

int num = 10;

for(int i =10;i>=0;i--)

{

std::cout<<i<<std::endl;

}

//练习1.13.2 输入2个整数，输出二者之间的所有整数

int innum1,innum2;

std::cout<<" input two increasing integer"<<std::endl;

std::cin>>innum1>>innum2;

if(innum1 > innum2)

{std::cout<<" input error! u idiot! \r\n please input two increasing integer!"<<std::endl;}

else

{

for(int i = innum1;i<=innum2;i++)

std::cout<<i<<std::endl;

}

//

/\*\*\*\*\*\*\*\*\*\*\*\*\*

//练习1.14

在for循环中，循环控制变量的初始化和修改都放在语句头部分，形式较简洁，且特别适用于循环次数已知的情况

在while循环中，循环控制变量一般在循环前初始化，形式没有for语句简洁，但是适用于循环次数未知的情况

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

while(1);

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\* 2019年3月24日16:03:34

## 1.4.4 练习

\*\* 要点：if语句

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <iostream>

//程序功能：记录连续输入整数中重复出现的数并输出

int main(void)

{

int currVal =0,val = 0;

if(std::cin>>currVal) //输入不合法就不执行

{

int cnt = 1; //计数

while(std::cin>>val)

{

if(val == currVal)++cnt; //记录重复的值

else

{

std::cout<<currVal<<" occurs "<< cnt << " times " <<std::endl;

currVal = val;

cnt = 1;

}

}

std::cout<<currVal<<" occurs "<< cnt << " times " <<std::endl;

}

while(1);

return 0;

}

1.17：输入值都相等，程序会只输出一次 xx occurs x times 就结束；若值全不相等，程序输出xx occurs 1 times ，次数与输入的数的个数相同

/\*\*\*\*\*\*\*\*\*\*88

\*\*\*

2019年3月24日16:18:50

## 1.4.4节练习

练习1.19

\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <iostream>

int main(void)

{

//练习1.19 输入2个整数，输出二者之间的所有整数

//修正逻辑以使输入不必升序

int innum1,innum2;

std::cout<<" input two integer"<<std::endl;

std::cin>>innum1>>innum2;

if(innum1 > innum2)

{

int temp;

temp = innum2;

innum2 = innum1;

innum1 = temp;

}

for(int i = innum1;i<=innum2;i++)

std::cout<<i<<std::endl;

while(1);

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*

2019年3月24日16:33:47

## 1.5节练习

书店程序

练习1.21 1.22 1.23 1.24

关键：类

\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <iostream>

#include "Sales\_item.h"

#define work1\_23 //编译开关

int main(void)

{

//课本例程 & 练习1.21

//功能：计算两个类对应成员的和

#ifdef work1\_21

Sales\_item item1,item2;

std::cin>> item1>>item2;

if(item1.isbn() == item2.isbn())

{

std::cout<<item1+item2<<std::endl;

//return 0;

}

else

{

std::cerr << "Data must refer to same ISBN"<<std::endl;

//return -1;

}

//以上代码对应练习1.21 实际为课本例程

#endif

//练习1.22 读取多个具有相同isbn的销售记录，输出所有记录和

#ifdef work1\_22

Sales\_item item1,item2;

if (std::cin>> item1)

{

int cnt = 1; //计数

while(std::cin>> item2)

{

if(item1.isbn() == item2.isbn())

{

cnt ++;

item1 += item2;

}

else

std::cerr << "Data must refer to same ISBN"<<std::endl;

}

std::cout<<"total record is "<<cnt<<"\r\n"<<item1<<std::endl;

}

#endif

#ifdef work1\_23

//练习2.23与2.24

//此程序其实有bug，必须连续输入相同的条目才可以统计到相同的书本信息，而且输入和输出混在一起非常难看

Sales\_item item1,item2;

if (std::cin>> item1)

{

int cnt = 1; //计数

while(std::cin>> item2)

{

if(item1.isbn() == item2.isbn())

{

cnt ++;

item1 += item2;

}

else

{

std::cout<<"total record of "<< item1.isbn()<<" is "<<cnt<<"\r\n"<<item1<<std::endl;

item1 = item2;

cnt = 1;

}

}

std::cout<<"total record of "<< item1.isbn()<<" is "<<cnt<<"\r\n"<<item1<<std::endl;

}

#endif

while(1);

}

## 第一章小结

第一章其实没什么特别的内容，主要收获是认识了输入输出两个“流”，掌握了cin和cout的用法

# 第2章 变量和基本类型

/\*\*\*

\*\* 2019年3月26日19:32:03

## 练习2.3

\*

\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <iostream>

int main(void)

{

unsigned u = 10,u2 = 42; //输出

std::cout << u2 - u << std::endl; //30

std::cout << u-u2 << std::endl; //很大的一个数

int i = 10,i2 = 42;

std::cout<<i2-i<<std::endl; //30

std::cout<<i-i2<<std::endl; //-30

std::cout<<i-u<<std::endl; //0

std::cout<<u-i<<std::endl; //0

while(1);

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\* 2019年3月26日20:02:38

## 练习2.40 练习2.41 练习2.42

\*\* 自行编写一个头文件，定义类，重写输出书本销售记录的程序

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <iostream>

#include "Sales\_data.h"

using namespace std;

#define work1\_23

int main(void)

{

#ifdef work1\_21

Sales\_data item1,item2;

double avg;

cin >> item1.bookNo >> item1.units\_sold >>item1.price ;

item1.revenue = item1.units\_sold\*item1.price ;

cin >> item2.bookNo >> item2.units\_sold >> item2.price;

item2.revenue = item2.units\_sold\*item2.price ;

if(item1.bookNo == item2.bookNo)

{

item2.units\_sold += item1.units\_sold;

item2.revenue += item1.revenue;

avg = item2.revenue / item2.units\_sold;

std::cout<<item2.bookNo<< item2.units\_sold <<item2.revenue<<avg<<std::endl;

//return 0;

}

else

{

std::cerr << "Data must refer to same ISBN"<<std::endl;

//return -1;

}

//以上代码对应练习1.21 实际为课本例程

#endif

//练习1.22 读取多个具有相同isbn的销售记录，输出所有记录和

#ifdef work1\_22

Sales\_data item1,item2;

double avg;

if (cin >> item1.bookNo >> item1.units\_sold >>item1.price)

{

int cnt = 1; //计数

item1.revenue = item1.units\_sold\*item1.price ;

while(cin >> item2.bookNo >> item2.units\_sold >>item2.price)

{

if(item1.bookNo == item2.bookNo)

{

cnt ++;

item2.revenue = item2.units\_sold\*item2.price ;

item1.units\_sold += item2.units\_sold;

item1.revenue += item2.revenue;

}

else

std::cerr << "Data must refer to same ISBN"<<std::endl;

}

avg = item1.revenue /item1.units\_sold;

std::cout<<"total record is "<<cnt<<"\r\n"<<item1.bookNo<<item1.units\_sold<<item1.revenue<<avg<<std::endl;

}

#endif

#ifdef work1\_23

//练习2.23与2.24

//此程序其实有bug，必须连续输入相同的条目才可以统计到相同的书本信息,而且输入和输出混在一起非常难看

Sales\_data item1,item2;

if (cin >> item1.bookNo >> item1.units\_sold >>item1.price)

{

int cnt = 1; //计数

item1.revenue = item1.units\_sold\*item1.price ;

while(cin >> item2.bookNo >> item2.units\_sold >>item2.price)

{

if(item1.bookNo == item2.bookNo)

{

cnt ++;

item2.revenue = item2.units\_sold\*item2.price ;

item1.units\_sold += item2.units\_sold;

item1.revenue += item2.revenue;

}

else

{

std::cout<<"total record is "<<cnt<<"\r\n"<<item1.bookNo<<item1.units\_sold<<item1.revenue<<std::endl;

item1 = item2;

cnt = 1;

}

}

std::cout<<"total record of "<< item1.bookNo<<" is "<<cnt<<"\r\n"<<item1.bookNo<<item1.units\_sold<<item1.revenue<<std::endl;

}

#endif

while(1);

}

程序段使用了条件编译，还是比较习惯叫结构体而不是类；

## 第二章小结

C++的基本变量类型和C差不多，图书馆程序调试起来有点难度，比较花时间。学习了类这种数据类型，但是对类的操作还非常不熟练，感觉类赋值非常繁琐。

# 第3章

/\* 3.2.2节练习

\*\*\*

## 练习3.2

\*\*

\*/

#include <iostream>

#include <string>

using namespace std;

#define work3\_5

int main()

{

#ifdef work3\_2

//练习3.2

string line;

string word;

getline(cin,line);

cout << line << endl;

while(cin>> word)

cout << word << endl;

#endif

#ifdef work3\_4

//练习3.4

string line[2];

getline(cin,line[0]);

getline(cin,line[1]);

//cout<<line[0]<<line[1];

if(line[0] == line[1])

cout <<"ture";

else

cout<< ((line[0]>line[1])?line[0]:line[1]); //比较是否想等并输出大的字符串

cout << endl;

if(line[0].size() == line[1].size())

cout <<"ture";

else cout<<((line[0].size() > line[1].size())?line[0]:line[1]); //比较长度是否想等，若不等输出较长字符串

cout << endl;

#endif

#ifdef work3\_5

//练习3.5 输入多个字符串，将其连起来输出总和

string sum;

string line;

while(cin>>line) //将输入的所有字符串连接起来

sum += line;

cout << sum<<endl; //使用结束符会导致后面的输入也一同结束

sum = "";

while(cin>>line) //将输入的所有字符串用空格隔开连起来

sum = sum + line + " ";

cout << sum<<endl;

#endif

while(1);

return 0;

}

/\*

\*\* 2019年4月3日20:19:57

## 练习3.2.3

知识点：范围for语句

\*\*

\*\*

\*\*\*/

#include <iostream>

#include <string>

using namespace std;

int main(void)

{

//练习3.6

string somestr("whosyourdaddy");

string otherstr;

cout << somestr<<endl;

for(auto &c : somestr)c = 'x';

cout << somestr;

cout << somestr[0] << endl;

//

//练习3.10

//输入一串包含标点符号的字符串，将标点去除后输出

somestr = "how are you ? how old are you ? how old are you two ? go ,leave me alone.";

cout << somestr << endl;

for (auto &c : somestr)

{

if (ispunct(c))continue;

else otherstr += c;

}

cout << otherstr << endl;

while(1);

}

/\*\*\*\*\*\*\*

2019年4月10日21:49:56

## 练习3.3.2

P91

练习3.14 ：编写一段程序，用cin读入一组整数并将其存入一个vector对象；

练习3.15 ： 改写上题程序，使之读入的对象为字符串；

练习3.16 ：将输入vector的元素输出

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*8\*/

#include <vector>

#include <iostream>

#include <string>

using namespace std;

int main(void)

{

//练习3.14

vector<int> num;

int temp;

for (int i = 0; i < 10; i++)

{

cin >> temp;

num.push\_back(temp);

}

for (auto &i : num) //必须要用for语句逐一访问

cout << i << ' ';

//练习3.14

string word; //我记得在C语言中变量只能在执行语句前定义声明

vector<string> someStr;

while (cin >> word)

{

someStr.push\_back(word);

}

for (auto &i : someStr) //必须要用for语句逐一访问

cout << i << ' ';

while (1);

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

2019年4月10日22:33:40

P94

## 练习3.3.3

练习3.16 将练习3.13中vector对象的内容输出出来

练习3.17 从cin读入一组词并把他们存入一个vector对象，然后将所有词改写为大写格式输出

练习3.20 读入一组整数并把他们存入一个vector对象，将每对相邻整数的和输出出来。改写程序，先输出第一个和最后一个元素的和，接着输出第二个和倒数第二个元素和，以此类推

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <iostream>

#include <string>

#include <vector>

using namespace std;

#define work\_3\_16

int main()

{

#ifdef work\_3\_16

vector<int> v1;

vector<int> v2(10);

vector<int> v3(10,42);

vector<int> v4{};

vector<int> v5{ 10, 42 };

vector<int> v6{ 10 };

vector<string> v7{10,"hi"};

cout << "v1: ";

for (auto c : v1)cout << c << " ";

cout << endl;

cout << "v2: ";

for (auto c : v2)cout << c << " ";

cout << endl;

cout << "v3: ";

for (auto c : v3)cout << c << " ";

cout << endl;

cout << "v4: ";

for (auto c : v4)cout << c << " ";

cout << endl;

cout << "v5: ";

for (auto c : v5)cout << c << " ";

cout << endl;

cout << "v6: ";

for (auto c : v6)cout << c << " ";

cout << endl;

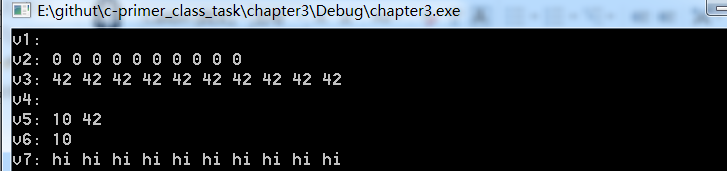
cout << "v7: ";

for (auto c : v7)cout << c << " ";

cout << endl;

#endif

输出



#ifdef word\_3\_17

//练习3.17

vector<string> inputWord;

string temp;

while (cin >> temp)

{

inputWord.push\_back(temp);

}

for (auto i : inputWord)

cout << i ;

cout << endl;

for (int i = 0; i < inputWord.size(); i++)

{

for (int j = 0; j < inputWord[i].length();j++)

{

inputWord[i][j] = toupper(inputWord[i][j]);

}

}

for (auto i : inputWord)

cout << i;

cout << endl;

#endif

#ifdef work\_3\_19

vector<int> vv1 {42,42,42,42,42,42,42,42,42,42};

vector<int> vv2(10, 42);

vector<int> vv3 = { 42, 42, 42, 42, 42, 42, 42, 42, 42, 42 };

//在定义重复的元素时，方法2是最便捷的，在需要定义列表元素时，显然方法1更直观更合适；

#endif

#ifdef work\_3\_20

vector<int> uu;

int temp;

while (cin >> temp)

{

uu.push\_back(temp);

}

for (int i = 0; i < uu.size(); i++)

{

if (i + 1 ==uu.size())

{

cout << uu[i] << " "; //引入防止下标溢出的边界判断

break;

}

else cout << uu[i] + uu[i + 1] << " ";

i++;

}

cout << "\n"<<"输出第i个和最后n-i个元素和" << endl;

for (int i = 0; i < uu.size()/2 ; i++)

{

cout << uu[i] + uu[uu.size() - i - 1] << " ";

}

if (uu.size() % 2 == 1) //对于奇数个数，中间的数输出它本身的两倍

cout << uu[uu.size() / 2]\*2 << " ";

#endif

while (1);

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

2019年4月16日20:43:16

P99

## 练习3.4.1

练习3.21 使用迭代器重做3.16

练习3.22 修改之前输出text的第一段程序，首先把text的第一段全改成大写，再输出它

练习3.23 编写一段程序，创建一个含有10个整数的vector对象，然后使用迭代器将所有元素的值变成原来的两倍。

输出vector对象的内容看是否正确

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <iostream>

#include <string>

#include <vector>

using namespace std;

#define work\_3\_23

int main()

{

#ifdef work\_3\_21

vector<int> v1;

vector<int> v2(10);

vector<int> v3(10, 42);

vector<int> v4{};

vector<int> v5{ 10, 42 };

vector<int> v6{ 10 };

vector<string> v7{ 10, "hi" };

cout << "v1: ";

for (auto it = v1.begin(); it != v1.end();it ++)cout << \*it << " ";

cout << endl;

cout << "v2: ";

for (auto it = v2.begin(); it != v2.end(); it++)cout << \*it << " ";

cout << endl;

cout << "v3: ";

for (auto it = v3.begin(); it != v3.end(); it++)cout << \*it << " ";

cout << endl;

cout << "v4: ";

for (auto it = v4.begin(); it != v4.end(); it++)cout << \*it << " ";

cout << endl;

cout << "v5: ";

for (auto it = v5.begin(); it != v5.end(); it++)cout << \*it << " ";

cout << endl;

cout << "v6: ";

for (auto it = v6.begin(); it != v6.end(); it++)cout << \*it << " ";

cout << endl;

cout << "v7: ";

for (auto it = v7.begin(); it != v7.end(); it++)cout << \*it << " ";

cout << endl;

#endif

#ifdef work\_3\_22

string s("hello world");

if (s.begin() != s.end())

{

auto it = s.begin();

\*it = toupper(\*it);

}

cout <<"改写首字母为大写： " <<s << endl;

for (auto it = s.begin(); it != s.end()&& !isspace(\*it);it++)

{

\*it = toupper(\*it);

}

cout <<"改写第一个单词为大写： "<< s << endl;

#endif

#ifdef work\_3\_23

vector<int> someNum(10,25);

for (auto c : someNum)

cout << c << " ";

cout << endl;

for (auto it = someNum.begin(); it != someNum.end(); it++)

{

\*it \*= 2;

}

for (auto c : someNum) //访问时必须用这种形式

cout << c << " ";

cout << endl;

for (auto it = someNum.begin(); it != someNum.end(); it++)

{

(\*it)++; //测试发现\*it++语句无效,必须加上括号

}

for (auto c : someNum) //访问时必须用这种形式

cout << c << " ";

cout << endl;

#endif

while (1);

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

2019年4月16日19:38:14

P101

## 练习3.4.2

练习3.24 使用迭代器重做3.20

练习3.25 使用迭代器实现3.3.3节的划分分数段的程序

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <iostream>

#include <string>

#include <vector>

using namespace std;

#define work\_3\_25

int main()

{

#ifdef work\_3\_24

vector<int> uu;

int temp;

while (cin >> temp)

{

uu.push\_back(temp);

}

cout << "输出相邻两个元素和，若一共奇数个，单独输出最后一个数" << endl;

for (auto it = uu.begin(); it != uu.end(); it++) //C++的迭代器其实跟C的指针很像

{

if (it + 1 == uu.end())

{

cout << \*it<< " "; //引入防止下标溢出的边界判断

break;

}

else cout << \*it + \*(it+1) << " ";

it++;

}

cout << "\n"<<"输出第i个和最后n-i个元素和" << endl;

auto itf = uu.begin();

auto ite = uu.end();

for (itf = uu.begin(); itf != uu.begin() + uu.size() / 2; itf++)

{

cout << \*itf + \*(ite-1) << " "; //uu.end()是一个不可访问的地址，访问它会导致溢出

ite--;

}

itf = uu.begin();

if (uu.size() % 2 == 1) //对于奇数个数，中间的数输出它本身的两倍

cout << \*(itf +(uu.size() / 2)) \* 2 << " ";

#endif

#ifdef work\_3\_25

vector<string> score\_part = { "0~9", "10~19", "20~29", "30~39", "40~49", "50~59", "60~69", "70~79", "80~89", "90~99","100" };

vector<unsigned> scores(11, 0);

unsigned grade;// , temp1;

// auto it = scores.begin();

//auto tempIt = scores.begin();

while (cin >> grade)

{

if (grade < 101)

{

auto it = scores.begin() + (grade / 10);

// tempIt = it + temp1; //

(\*it)++;

}

}

for (auto c : score\_part)

cout << c << " ";

cout << endl;

for (auto c : scores)

cout << c << " ";

#endif

while (1);

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

2019年4月16日22:10:57

P104

## 练习3.5.2

练习3.31 编写一段程序定义一个含有10个int的数组，令每个元素的值为其下标值

练习3.32 将上一题创建的数组拷贝给另外一个数组，利用vector重写程序实现类似的功能；

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <iostream>

#include <string>

#include <vector>

#include <stdio.h>

using namespace std;

int main()

{

//练习3.31

int a[10];

for (int i = 0; i < 10; i++)

a[i] = i;

cout << "a :";

for (int i = 0; i < 10; i++)

cout << a[i] << " ";

cout << endl;

//练习3.32

int b[10];

//strcpy(b,a);z只能拷贝字符串 char 类型

for (int i = 0; i < 10; i++)

b[i] = a[i];

cout << "b : ";

for (int i = 0; i < 10; i++)

cout << b[i] << " ";

cout << endl;

//使用vector重写

cout <<"vector reprogram "<< endl;

vector<int> v1 = { 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 };

vector<int> v2(10, 0);

cout << "v1: ";

for (auto it = v1.begin(); it != v1.end(); it++)

cout << \*it << " ";

auto it = v1.begin();

for (auto &c : v2)

{

c = \*it;

if (it != v1.end())

it++;

}

cout <<endl << "v2: ";

for (auto c : v2)

cout << c << " ";

while (1);

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

2019年4月16日22:33:56

P108

## 练习3.5.3

练习3.35 编写一段程序利用指针将数组中的元素置为0

练习3.36 编写一段程序比较两数组是否想等，再写一段程序比较连个vector对象是否想等

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <iostream>

#include <string>

#include <vector>

#include <stdio.h>

using namespace std;

int main()

{

//练习3.31

int a[10];

int \*p;

p = a;

for (int i = 0; i < 10; i++)

{

\*p = 0;

p++;

}

cout << "a :";

for (int i = 0; i < 10; i++)

cout << a[i] << " ";

cout << endl;

//练习3.36

//未指明比较数组比较的是什么，如果是比较数组名，是比较数组首地址；如果比较数组所有元素，则需要逐一遍历数组元素才能确认是否想等，会写，时间关系不写了

//

while (1);

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

2019年4月16日22:10:57

P104

练习3.5.5

练习3.41 编写一段程序，用整形数组初始化一个vector对象

练习3.42 将含有整数元素的vector对象拷贝给整形数组

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <iostream>

#include <string>

#include <vector>

#include <stdio.h>

using namespace std;

int main()

{

//练习3.41

int a[10];

for (int i = 0; i < 10; i++)

a[i] = i;

cout << "a :";

for (int i = 0; i < 10; i++)

cout << a[i] << " ";

cout << endl;

vector<int> v1(begin(a),end(a));

cout << "传过来的v1 ：";

for (auto c : v1)

cout << c << " ";

cout << endl;

//练习3.42

int b[20];

cout << "array b :";

for (int i = 0; i < v1.size(); i++)

{

b[i] = v1[i];

}

for (int i = 0; i < v1.size(); i++)

cout << b[i] << " ";

cout << endl;

while (1);

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

2019年4月16日22:10:57

P104

## 练习3.5.5

练习3.43 编写3个版本程序，均输出ia的元素，版本1使用范围for语句管理迭代过程，版本2、3使用普通for语句，版本2使用下标运算符，版本3使用指针，

不允许使用类型别名，auto关键字或者decltype关键字

练习3.44 将3.43的循环变量用类型别名

练习3.45 使用auto关键字实现

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <iostream>

#include <string>

#include <vector>

#include <stdio.h>

using namespace std;

int main(void)

{

int ia[3][4] = { 1, 2, 3, 4, 5 ,6,7,8,9,10,11,12};

//练习3.43

//版本1

cout << "vertion 1 :" << endl;

for (int(&row)[4] : ia)

{

for (int col : row)

cout << col << " ";

cout << endl;

}

cout << endl;

////版本2

cout << "vertion 2 :" << endl;

for (int i = 0; i < 3; i++)

{

for (int j = 0; j < 4; j++)

cout << ia[i][j] << " ";

cout << endl;

}

cout << endl;

////版本3

int (\*p)[4] = ia; //指向含有4个整形的数组的指针

int \*q;

cout << "vertion 3 :" << endl;

for (; p < ia + 3; p++)

{

for (q = \*p; q < 4 + \*p; q++)

cout << \*q << " ";

cout << endl;

}

cout << endl;

//练习3.44

cout << "接下来使用类型声明控制循环变量" << endl;

//版本1

typedef int int\_array[4];

//这个实在不知道怎么写

for (int\_array \*p1 = ia; p1 != ia + 3;++p1)

{

for (int \*q1 = \*p1; q1 < 4 + \*p1; q1++)

cout << \*q1 << " ";

cout << endl;

}

cout << endl;

//练习3.45 使用范围for语句

cout << "使用范围for语句控制循环变量" << endl;

for (auto &row : ia)

{

for (auto col : row)

cout << col << " ";

cout << endl;

}

while (1);

}

## 第三章小结

学习了string类型，vector类型，迭代器的用法，数组我在C语言中已经比较熟练掌握了。C++11引入了范围for语句和auto变量赋值，开始不太熟悉，用起来之后感觉真的好用。