西安交通大学实验报告

课程 计算机程序设计 实验名称 标准库和输入输出流 第 1 页 共 21页

系 别\_\_\_\_\_\_\_\_\_钱 学 森 班\_\_\_\_\_\_\_\_\_ 实 验 日 期 2018年 5月 30日

专业班级\_\_\_\_钱学森73\_\_组别\_\_\_\_\_\_\_\_\_\_\_ 实 验 报 告 日 期 2018年 6月 1日

姓 名\_\_\_刘贵涛\_\_\_学号\_\_2176110040\_ 报 告 退 发 ( 订正 、 重做 )

同 组 人\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 教 师 审 批 签 字

1. 实验目的

（1）掌握C++语言输入和输出操作的方法；

（2）掌握C++的流和流类库的使用方法。

1. 实验内容
2. 实验题目：编写一个程序，分别用不同的域宽（0~10）打印整数12345和浮点数1.2345。观察当域宽小于数值实际需要的域宽时会发生什么。
   1. 要点分析

第一步：按不同域宽输出整数12345

第二步：按不同域宽输出浮点数1.2345

* 1. 程序源代码

#include<iostream>

#include<iomanip>

using namespace std;

int main()

{

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

{

cout << setw(i) << 12345 << endl;

}

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

{

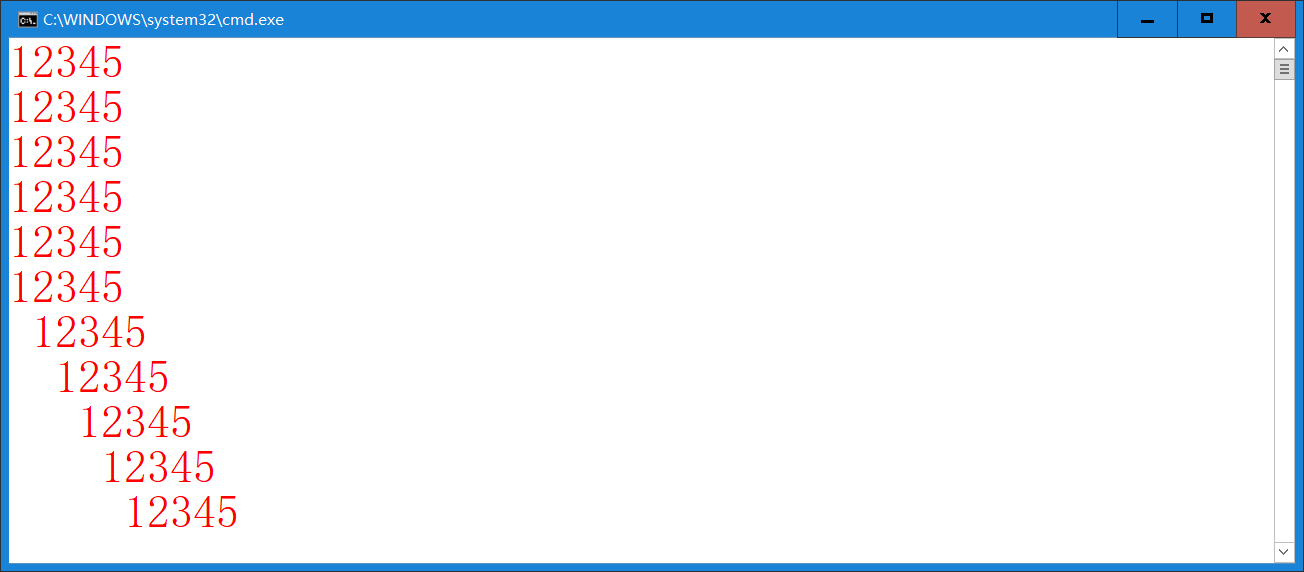
cout << setw(i) << 1.2345 << endl;

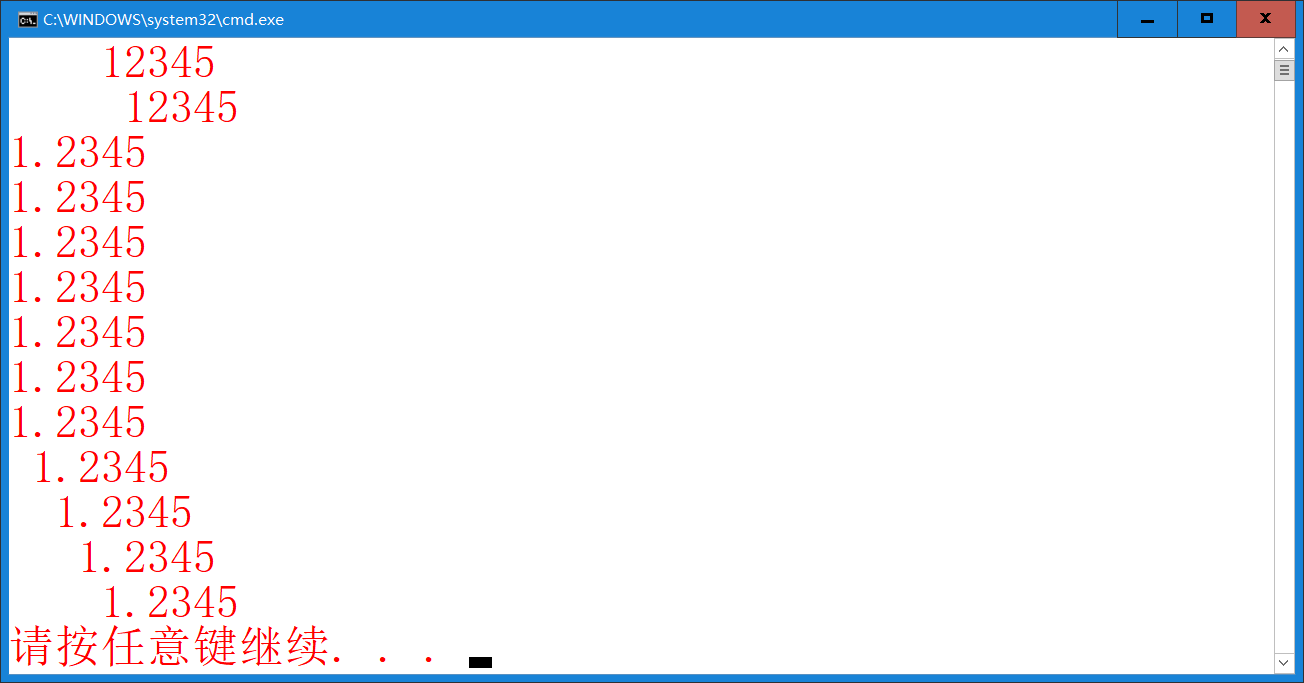
}

return 0;

}

* 1. 实验结果





1. 实验题目：编写一个程序，将华氏温度0~212℉转换为浮点型摄氏温度，浮点数精度为3。转换公式如下：

Celisius = 5.0/9.0\*(Fahrenheit – 32)

输出用两个右对齐列，摄氏温度前面加上正负号。

1. 要点分析

第一步：编写转换函数F(a)，用于华氏度和摄氏度温度的转换

第二步：从0℉循环到212℉，并输出其对应的摄氏度数值，摄氏度输出时加上正负号

1. 程序源代码

#include<iostream>

#include<iomanip>

using namespace std;

double F(int a) { return 5.0 / 9.0\*((double)a - 32); }

int main()

{

cout << setw(8) << "华氏度" << setw(8) << "摄氏度" << endl;

for (int i = 0; i <= 212; i++)

{

cout << setw(8) << i << setw(8);

cout.setf(ios::showpos);

cout << setprecision(3) << F(i) << endl;

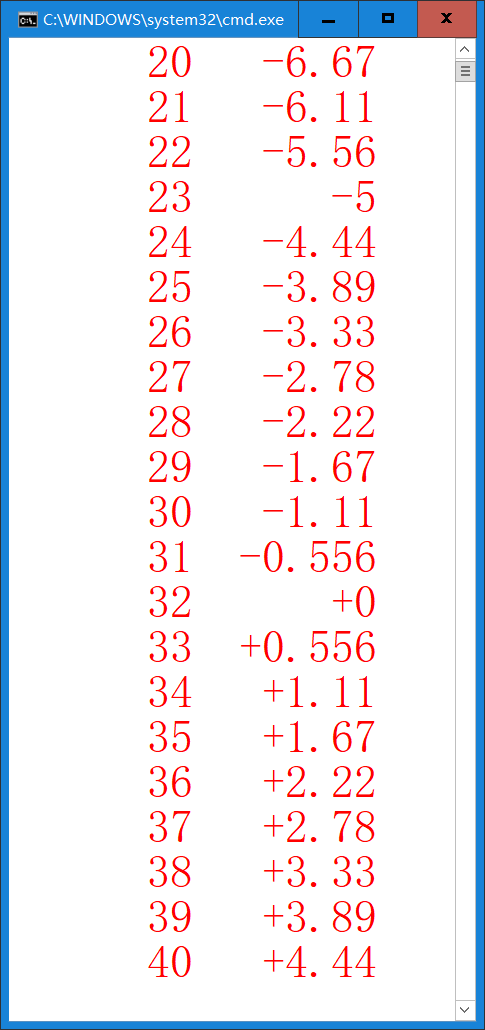
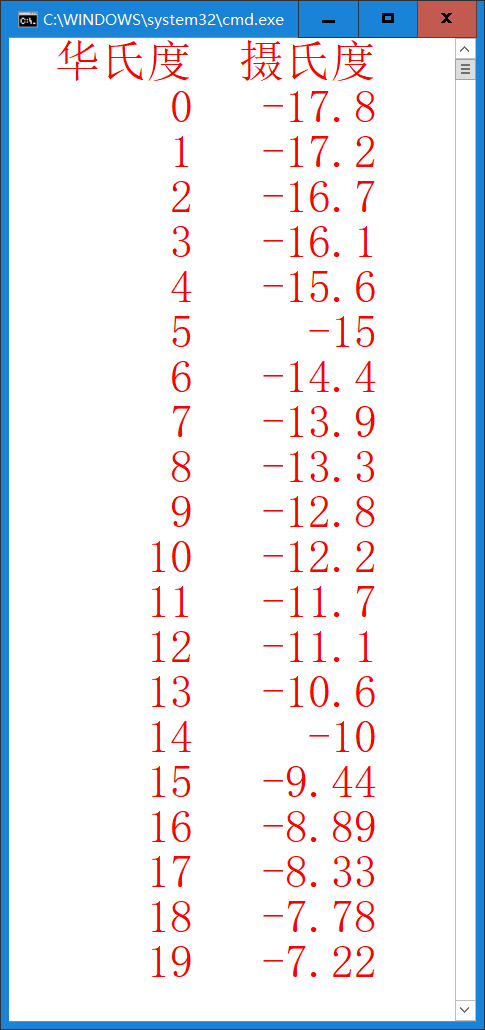
cout.unsetf(ios::showpos);

}

return 0;

}

1. 实验结果





1. 实验题目：编写一个程序，打印出ASCII字符集中码值为33~126的字符的ASCII码表。要求输出十进制值、八进制值、十六进制值以及码值所表示的字符。
2. 要点分析

输出33-126ASCII码所对应的字符，以及33-126在十进制，八进制，十六进制下的数字

1. 程序源代码

#include<iostream>

#include<iomanip>

using namespace std;

int main()

{

cout << "ASCII码表" << endl;

cout << setw(8) << "十进制" << setw(8) << "八进制" << setw(10) << "十六进制" << setw(10) << "表示字符" << endl;

for (int i = 33; i <= 126; i++)

{

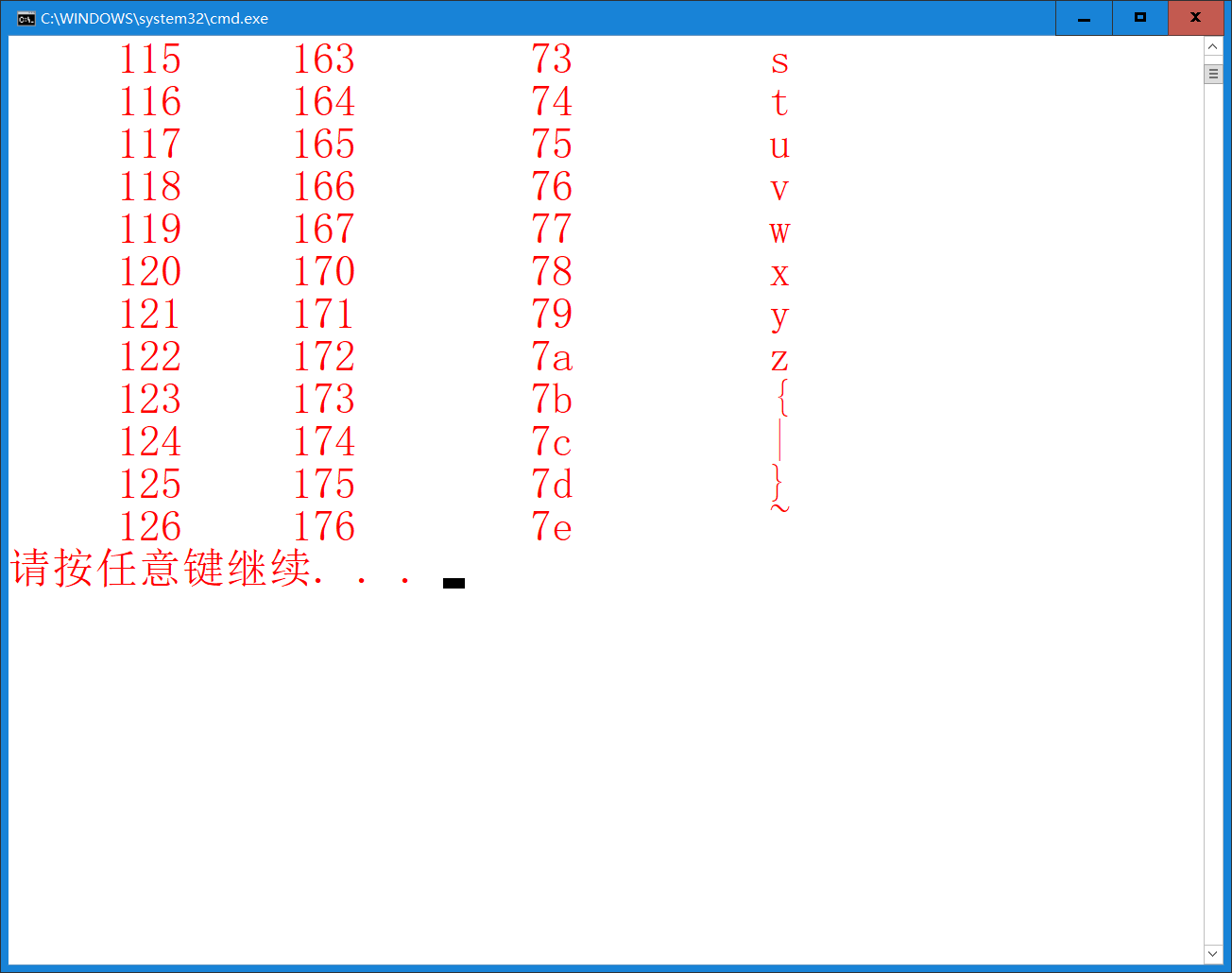
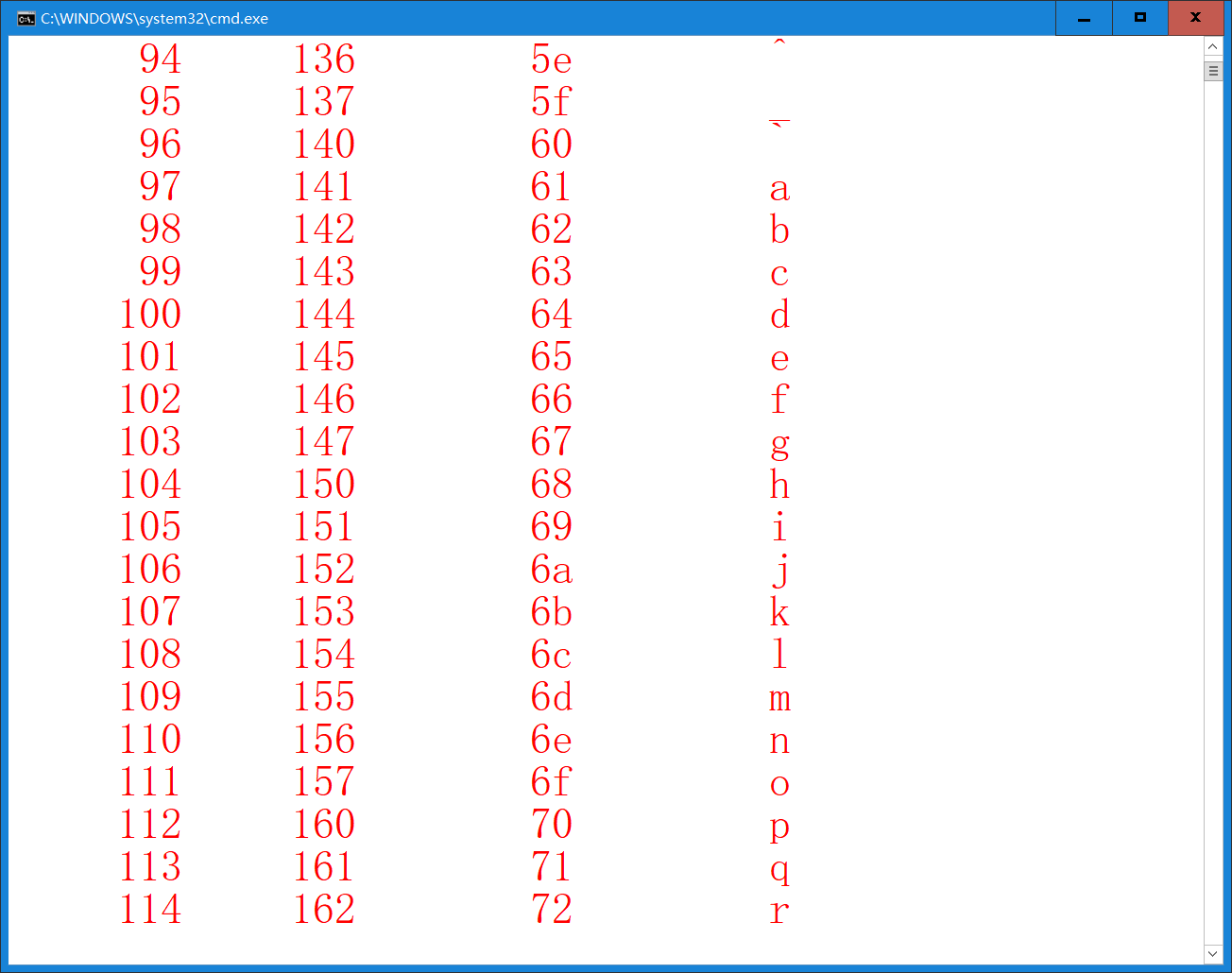
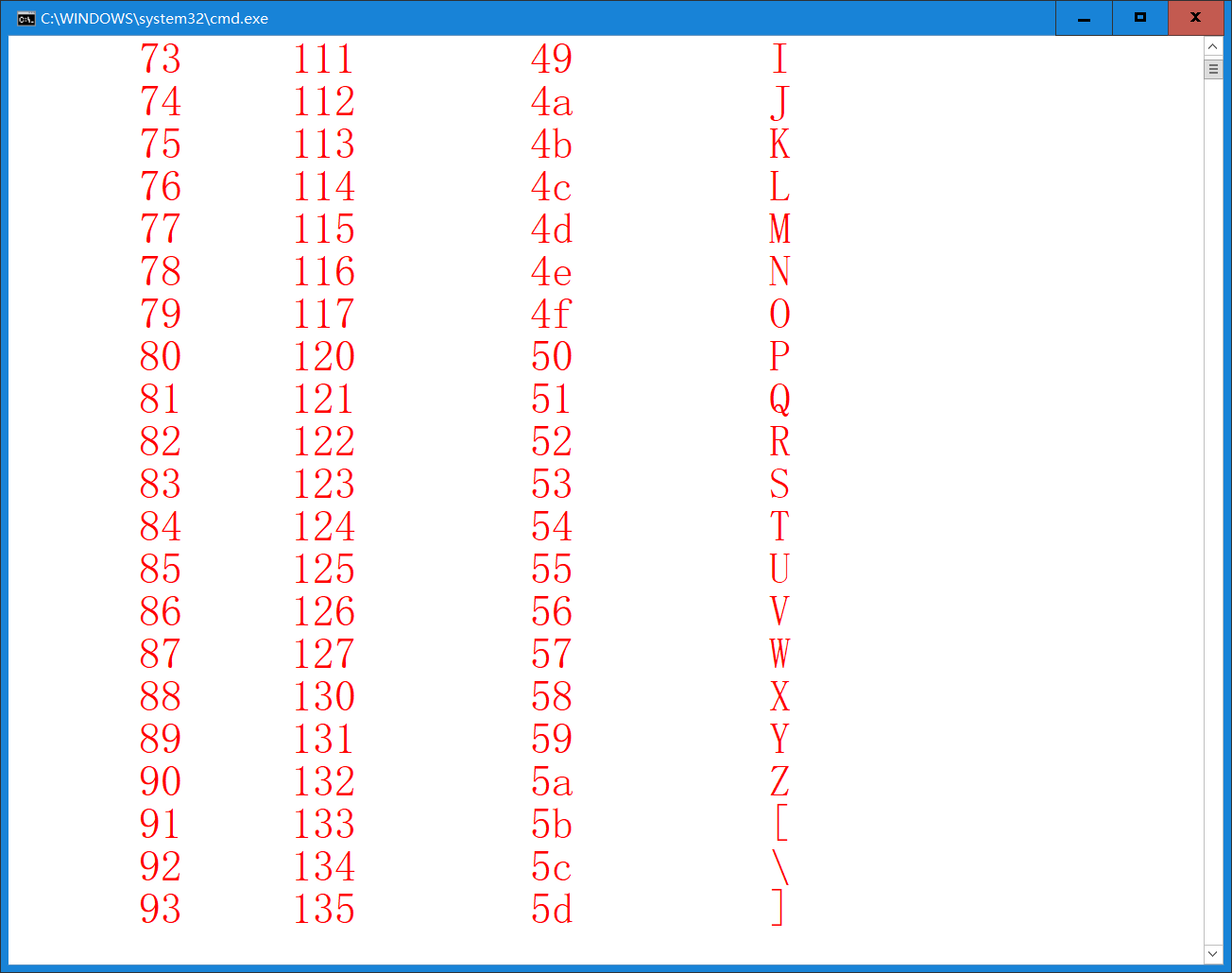
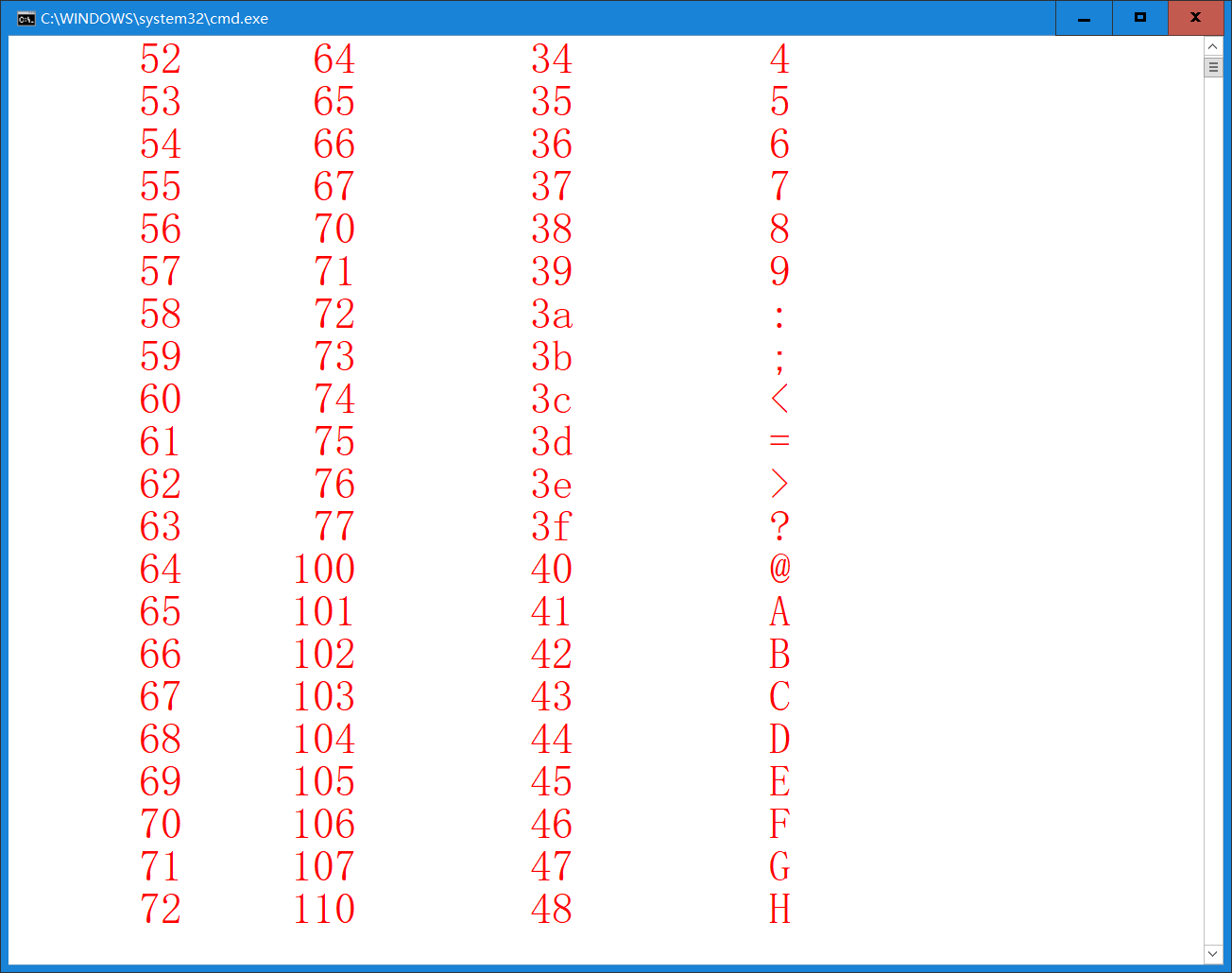
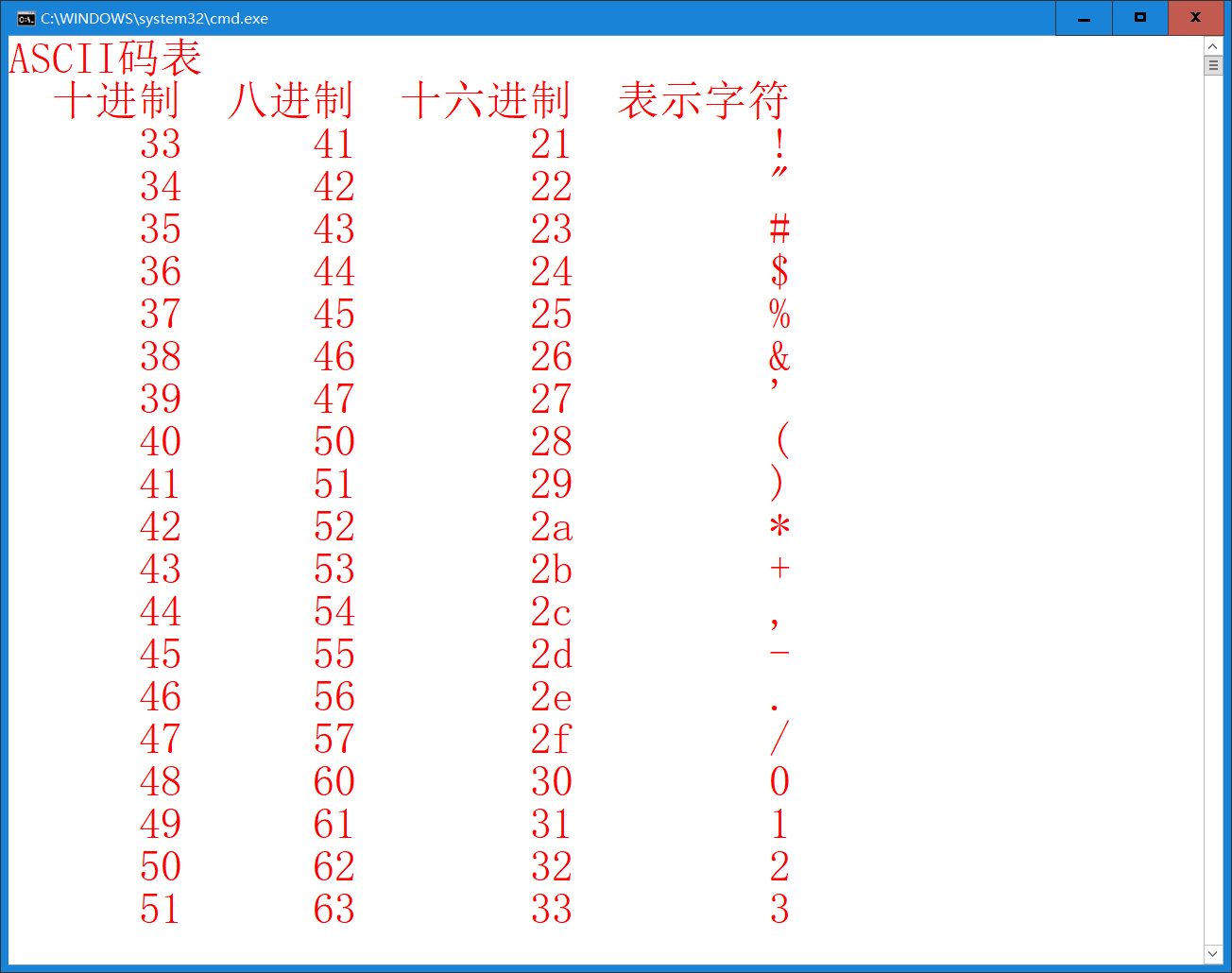
cout << setw(8) << i << setw(8) << oct << i << setw(10) << hex << i << setw(10) << (char)i << setbase(10) << endl;

}

return 0;

}

1. 实验结果



1. 实验题目：修改例11-9程序，重载>>运算符，使其能够直接使用cin语句输入Date类对象。
2. 要点分析

第一步：重载>>运算符，使其可以直接用cin输入Date类对象

第二步：编写主函数验证>>重载

1. 程序源代码

#include<iostream>

#include<time.h>

using namespace std;

class Date

{

public:

Date(int y, int m, int d);

Date(int m, int d);

Date();

void SystemDate();

void SetDate(int yy, int mm, int dd);

void SetDate(int mm, int dd);

bool IsLeapYear(int yy)const;

bool IsEndofMonth()const;

void print\_ymd()const;

void print\_mdy()const;

const Date &operator +(int days);

const Date &operator +=(int days);

int operator-(const Date& ymd)const;

friend istream &operator>>(istream&, Date &);

int DayCalc()const;

int GetYear() { return year; }

int GetMonth() { return month; }

int GetDay() { return day; }

private:

int day, month, year;

void IncDay();

static const int days[];

};

const int Date::days[] = { 0,31,28,31,30,31,30,31,31,30,31,30,31 };

Date::Date(int y, int m, int d) { SetDate(y, m, d); }

Date::Date(int m, int d) { SetDate(m, d); }

Date::Date() { SystemDate(); }

void Date::SystemDate()

{

tm\*gm;

time\_t t = time(NULL);

gm = gmtime(&t);

year = 1900 + gm->tm\_year;

month = gm->tm\_mon + 1;

day = gm->tm\_mday;

}

void Date::SetDate(int yy, int mm, int dd)

{

month = (mm >= 1 && mm <= 12) ? mm : 1;

year = (yy >= 1900 && yy <= 2100) ? yy : 1900;

if (month == 2 && IsLeapYear(yy))

day = (dd >= 1 && dd <= 29) ? dd : 1;

else day = (dd >= 1 && dd <= days[month]) ? dd : 1;

}

void Date::SetDate(int mm, int dd)

{

tm\*gm;

time\_t t = time(NULL);

gm = gmtime(&t);

month = (mm >= 1 && mm <= 12) ? mm : 1;

year = 1900 + gm->tm\_year;

if (month == 2 && IsLeapYear(year))

day = (dd >= 1 && dd <= 29) ? dd : 1;

else day = (dd >= 1 && dd <= days[month]) ? dd : 1;

}

const Date&Date::operator+(int days)

{

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

IncDay();

return \*this;

}

const Date&Date::operator +=(int days)

{

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

IncDay();

return \*this;

}

int Date::operator-(const Date&ymd)const

{

int days;

days = DayCalc() - ymd.DayCalc();

return days;

}

bool Date::IsLeapYear(int y)const

{

if (y % 400 == 0 || (y % 100 != 0 && y % 4 == 0))return true;

return false;

}

bool Date::IsEndofMonth()const

{

if (month == 2 && IsLeapYear(year))

return day == 29;

else return day == days[month];

}

void Date::IncDay()

{

if (IsEndofMonth())

{

if (month == 12)

{

day = 1; month = 1; year++;

}

else

{

day = 1; month++;

}

}

else day++;

}

int Date::DayCalc()const

{

int dd;

int yy = year - 1900;

dd = yy \* 365;

if (yy)dd += (yy - 1) / 4;

for (int i = 1; i < month; i++)dd += days[i];

if (IsLeapYear(year) && (month > 2))dd++;

dd += day;

return dd;

}

void Date::print\_ymd()const

{

cout << year << '-' << month << '-' << day << endl;

}

void Date::print\_mdy()const

{

const char \*monthName[12] = { "January","February","March","April","May","June","July","August","September","October","November","December" };

cout << monthName[month - 1] << ' ' << day << ", " << year << endl;

}

istream & operator>>(istream &input, Date &date)

{

input >> date.year >> date.month >> date.day;

if (!input)

cout << "error!";

else

return input;

}

int main()

{

Date today;

int m1, d1;

cout << "今天的日期是："; today.print\_ymd();

cout << "本程序可用于判断你输入的日期是不是今天\n请输入一组年月日：";

Date input\_day;

cin >> input\_day;

if (input\_day.GetDay() == today.GetDay() && input\_day.GetMonth() == today.GetMonth() && input\_day.GetYear() == today.GetYear())

{

cout << "输入的日期是今天" << endl;

}

else

{

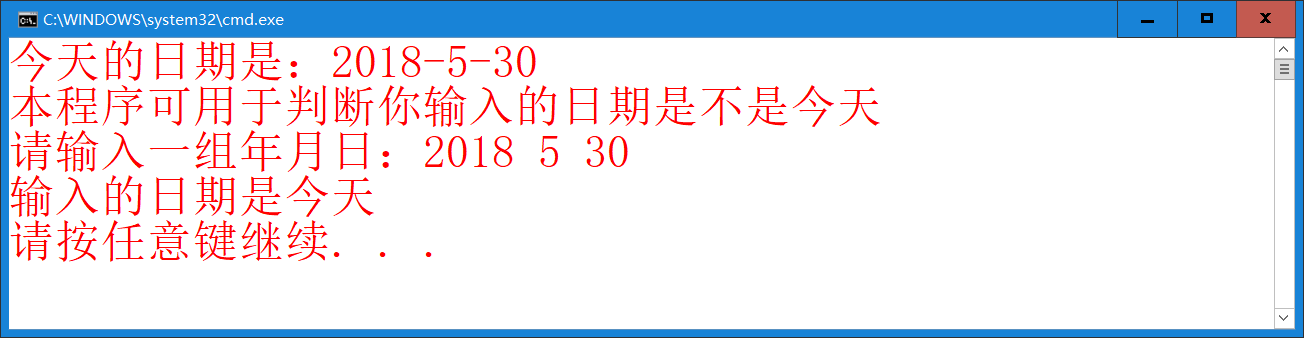
cout << "输入的日期不是今天" << endl;

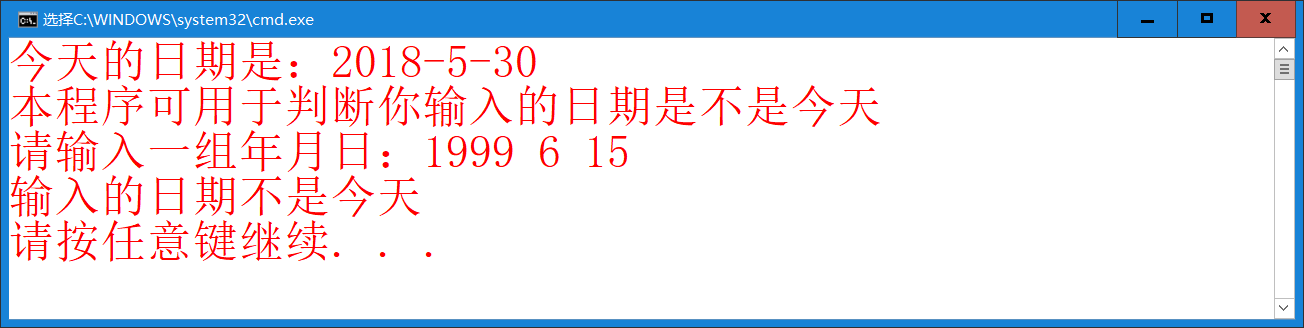
}

return 0;

}

1. 实验结果





1. 实验题目：编写一个程序，可以读入一个C++语言的源文件，每一行上上加上行号保存到另一个后缀为.prn的同名文件中。
2. 要点分析

第一步：使用ifstream定义一个输入流in，并打开一个源文件

第二步：逐行读取源文件中的信息

第三步：使用ofstream定义一个输出流out，并创建一个新文件(.prn)

第四步：填上行号，并将读取得到信息输出到新文件中

1. 程序源代码

#include<iostream>

#include<fstream>

using namespace std;

int main()

{

ifstream in("e:\\demo.cpp");

if (!in)

{

cout << "无法打开demo.cpp文件" << endl;

return 1;

}

char len[999];

ofstream out("e:\\demo1.prn");

if (!out)

{

cout << "无法创建demo1.prn文件" << endl;

}

in.getline(len, 999);

int i = 1;

while(in)

{

out << i++ << ". " << len << endl;

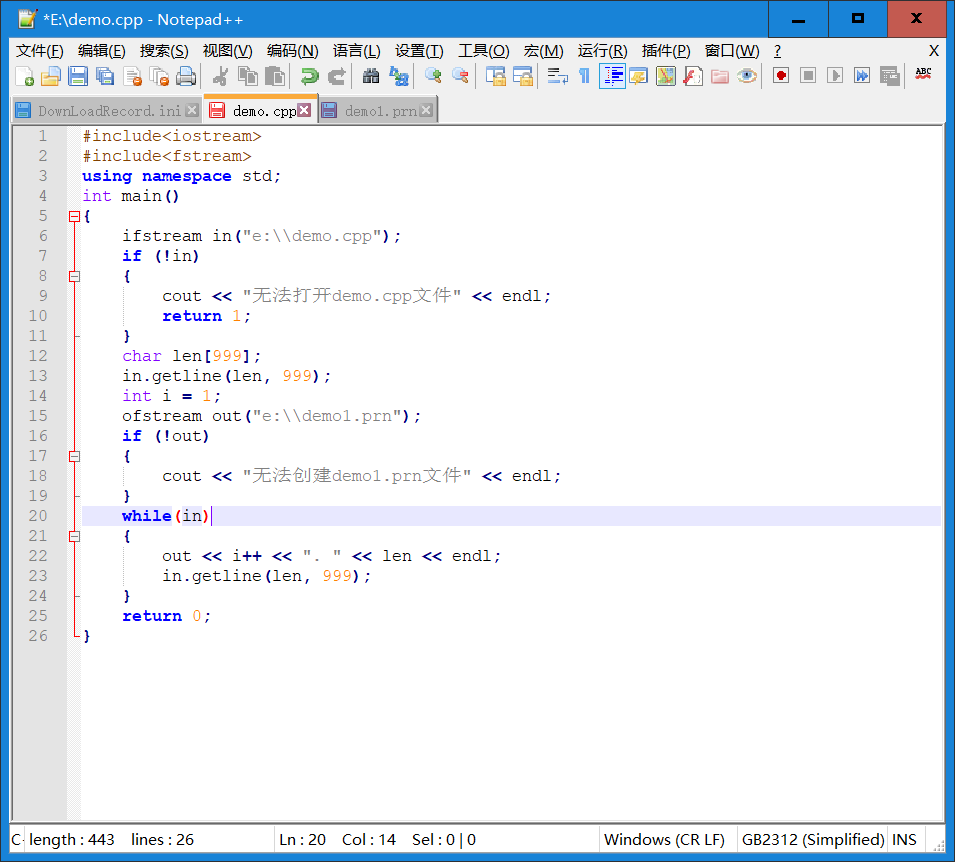
in.getline(len, 999);

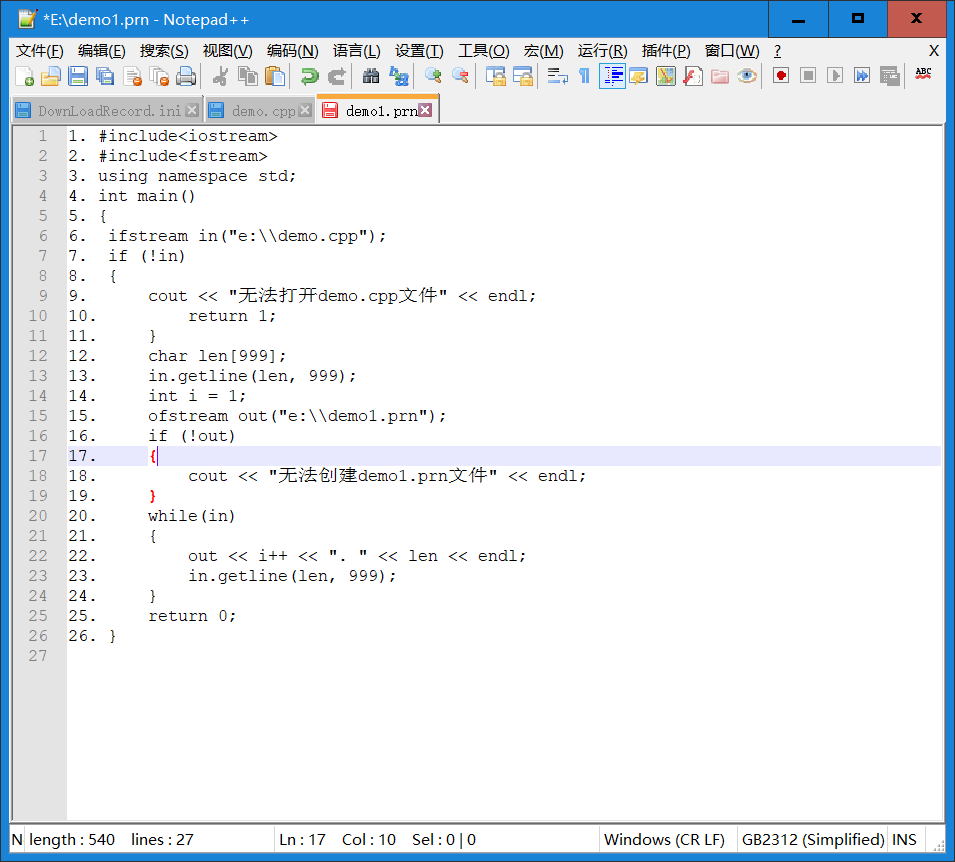
}

return 0;

}

1. 实验结果





1. 实验题目：（必做题）将一个文本文件内容用凯撒尔方法加密，密钥是4。即文本文件中字母用其后第4个字母代替，若是数字则用其后第4个数字代替。例如文本文件原内容为“Red And Black 2008”，加密后文本文件内容为“Vih Erh Fpego 6442”。（提示：先用记事本创建一个文本文件，其内容应该是有意义的英文句子，然后以只读方式打开，加密后的内容写入另一个文本文件中。）
2. 要点分析

第一步：读取文件，并使用凯撒加密

第二步：将加密后内容输出到新创建的文件中，完成加密

1. 程序源代码

#include<iostream>

#include<fstream>

using namespace std;

void jiami(char \*a)

{

for (int i = 0; i < strlen(a); i++)

{

if (a[i] >= 'a'&&a[i] <= 'z') a[i] = (a[i] + 4 - 'a') % 26 + 'a';

else if (a[i] >= 'A'&&a[i] <= 'Z')a[i] = (a[i] + 4 - 'A') % 26 + 'A';

else if (a[i] >= '0'&&a[i] <= '9')a[i] = (a[i] + 4 - '0') % 10 + '0';

}

}

int main()

{

ifstream in("e:\\yuan.txt");

if (!in)

{

cout << "无法打开yuan.txt文件" << endl;

return 1;

}

char aa[999];

ofstream out("e:\\jiami.txt");

if (!out)

{

cout << "无法创建jiami.txt文件" << endl;

}

in.getline(aa, 999);

while(in)

{

jiami(aa);

out << aa;

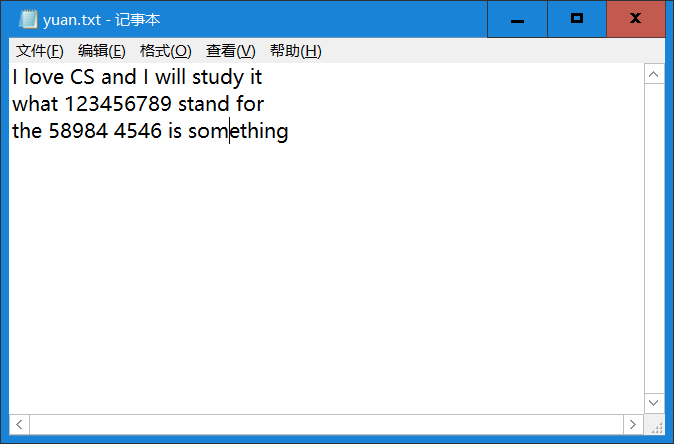
in.getline(aa, 999);

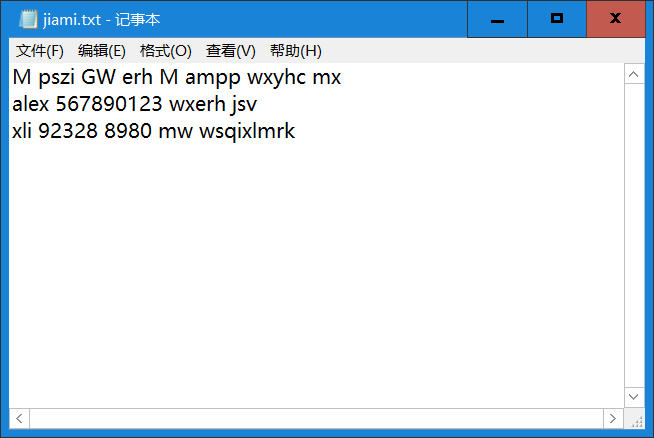
}

return 0;

}

1. 实验结果





1. 实验题目：（必作题）找出100以内的勾股数，输出到文件gouku中。所谓勾股数指找出三个数满足A2+B2=C2，并且A<B<C。要求将三个勾股数的计算公式A^2+B^2=C^2输出到文件中的每一行，例如3^2+4^2=5^2 9+16=25。
2. 要点分析

第一步：设置三个整形变量i, j, k，循环增加其值，并判断是否为勾股数

第二步：将得到的勾股数，按题目要求输出到新的文件中

1. 程序源代码

#include<iostream>

#include<fstream>

using namespace std;

int main()

{

ofstream out("e:\\gouku");

if (!out)

{

cout << "无法创建gouku文件";

return 1;

}

for (int i = 1; i <= 100; i++)

{

for (int j = i + 1; j <= 100; j++)

{

for(int k=j+1;k<=100;k++)

if(k\*k==i\*i+j\*j&&i+j>k)

out << i << "^2 + " << j << "^2 = " << k << "^2 " << i \* i << " + " << j \* j << " = " << k \* k << endl;

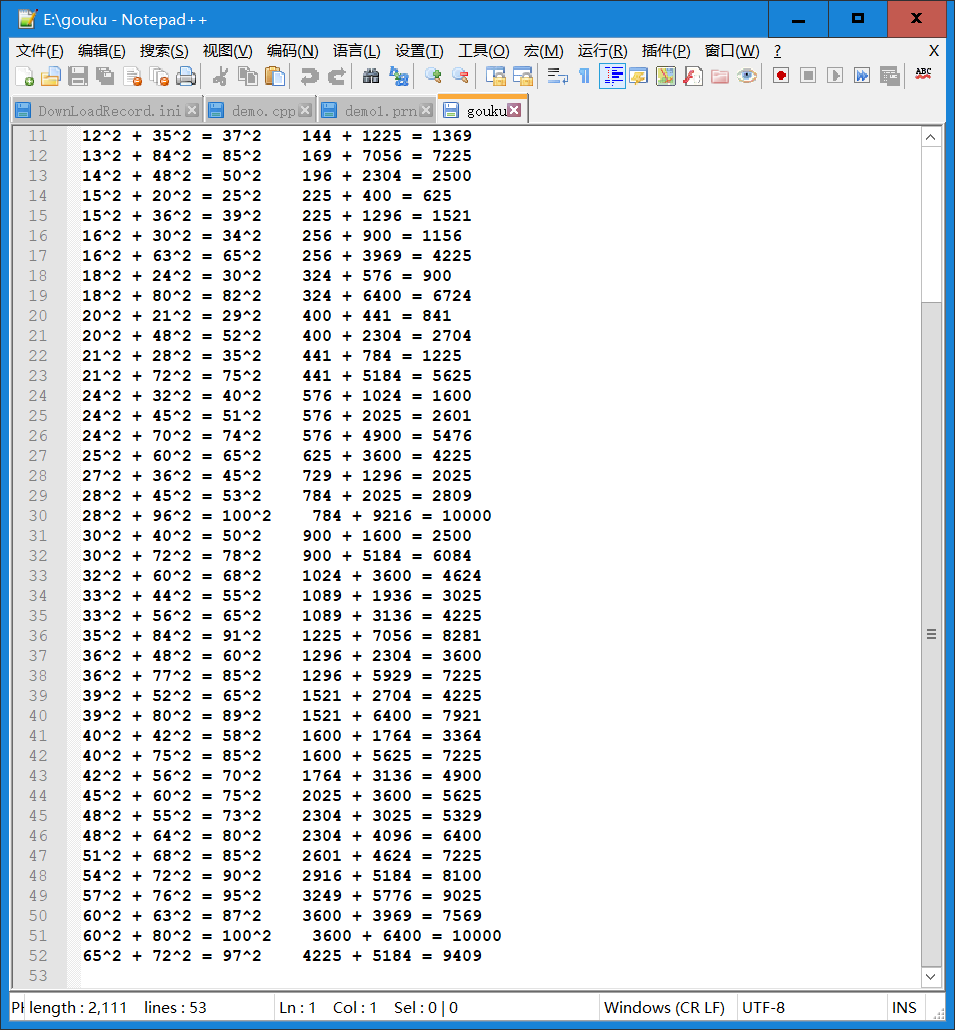
}

}

return 0;

}

1. 实验结果



1. 实验题目：（选做题）仿造实现DOS文件复制命令COPY 源文件名 目标文件名。（提示：利用main（）函数中的参数int argv,char \*argk[ ]识别解析命令。）
2. 要点分析

第一步：编写一个名为mycopy的程序，其内容如下代码

第二步：使用cmd，进入文件所在目录，调用文件完成实验

1. 程序源代码

#include<iostream>

#include<fstream>

using namespace std;

int main(int argc, char \*argv[])

{

char ch;

if (argc != 3)

{

cout << "Usage: ProrameNmae<source filename><target filename>\n";

return 1;

}

ifstream in(argv[1], ios::in | ios::binary);

ofstream out(argv[2], ios::out | ios::binary);

if (!in)

{

cout << "Cannot open the file. ";

return 1;

}

if (!out)

{

cout << "Cannot create the new file. ";

}

while (in)

{

in.get(ch);

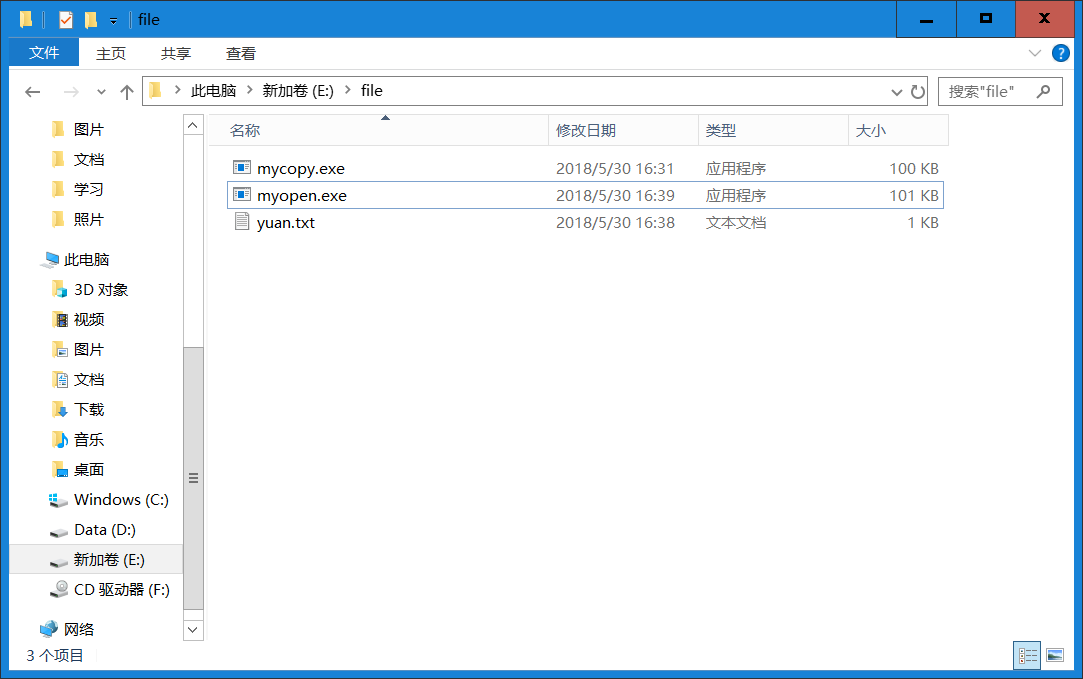
if (in) out << ch;

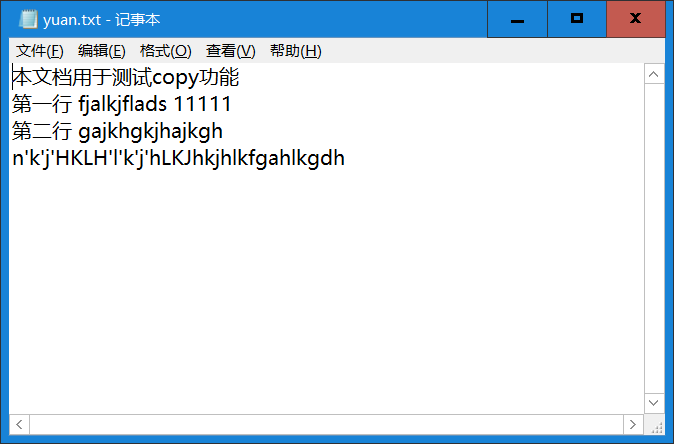
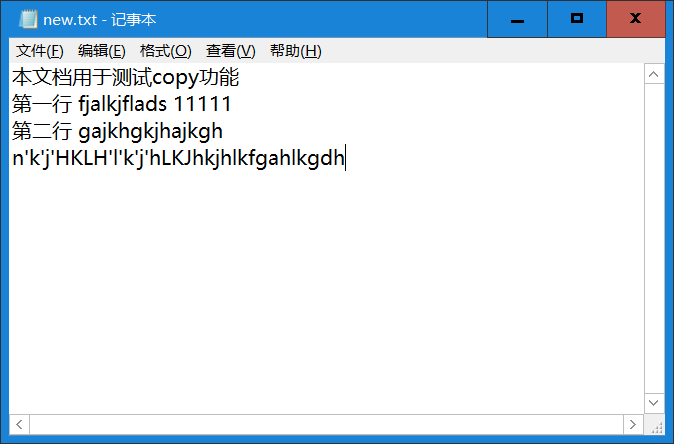
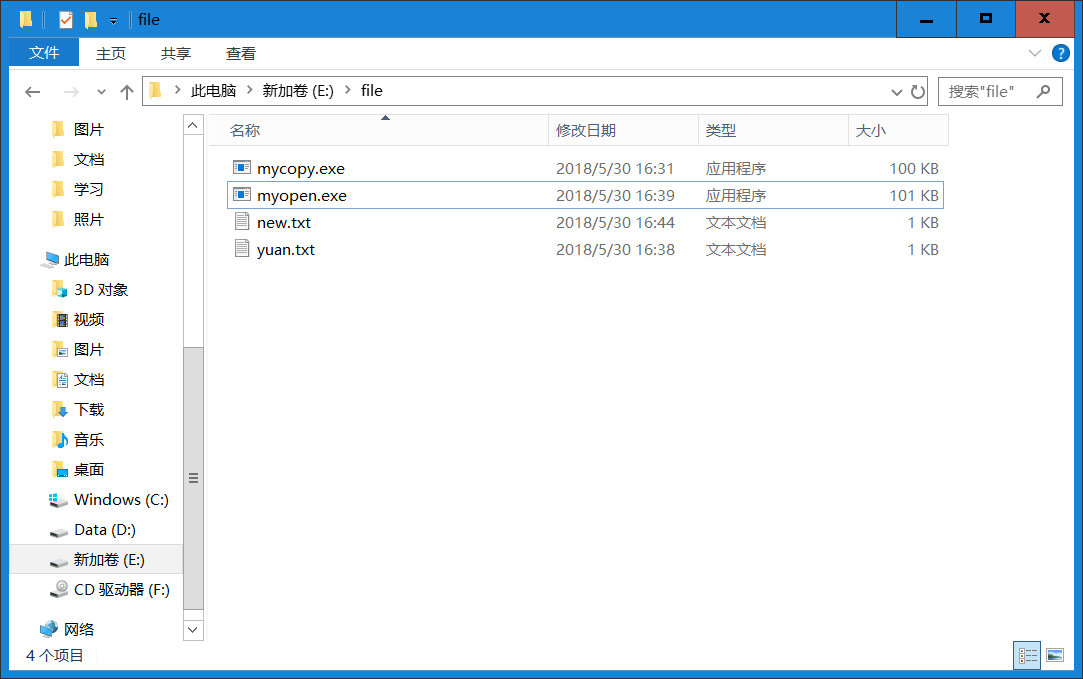
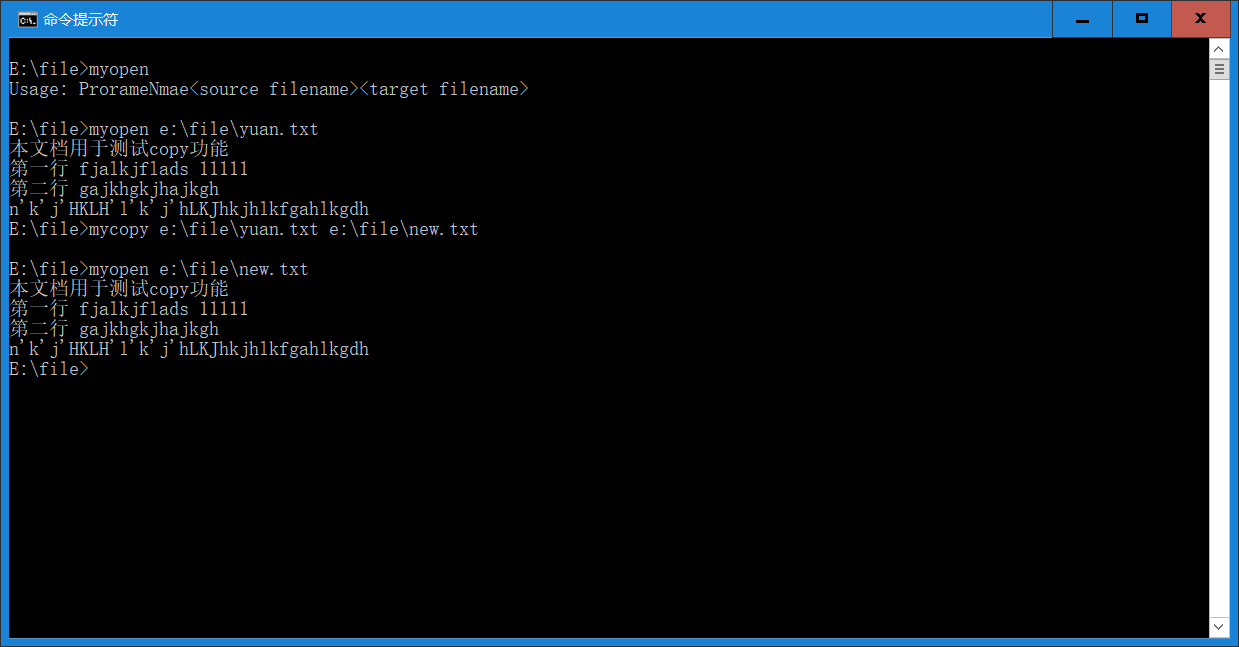
}

return 0;

}

1. 实验结果





1. 个人小结
2. 对比分析

对比学长的报告，总体来说，其使用了过多的注释使程序显得杂乱，不简洁，而且偏好使用绿色，导致报告难以阅读，而且报告整体过于花哨，让人不知道这是一篇正式的实验报告，第六题中出现了分数，代码出错，赵春雨报告中对无需自定义的析构函数可以不写出，其对前置后置的分析十分到位，值得学习。

1. 个人小结

本次实验每个题的难度不是很大，但是提供了一个新的编程思路及方式，就是对文件的操作，这可以解决为后续中更多更复杂的问题提供便利，整体题目不费时间，需要的时对新知识的灵活应用，尤其最后一个题，了解到了cmd对应用的使用，以及带有参数的main函数的使用方法，很大程度上能够起到拓宽学生思路的作用。