

计算机语言程序设计

实习指导书（2017）

主编：刘传平

编者：刘尚一 刘燕

中国地质大学（北京）计算机基础教研室

2017 年 2 月

前 言

本书是我校公共基础课“计算机语言程序设计”课程教学的配套实习教材。我校计算机高级语言基础课教学自 2007 年以来进行了一系列改革。2007 年,信息工程学院 8 个班试点 C++ 语言程序设计教学。2011 年选择 C++ 语言的班级达到了 32 个,上机学时也调整为占总学时的一半(32 学时)。2015 年我校引入百科园机考系统,C++ 语言期中考试实现全部机考,2016 年,C++ 语言更注重学生平时学习情况的跟踪,将进行 2 次月考、1 次期中考试、10 次平时测验,更加强调学生的实践动手能力培养,2017 年开始全校学生学习《计算机语言程序设计》课程,更重视实际编程能力的提升。

在实习项目设计和选取过程中,结合计算机语言程序设计课程改革和教学需要,广泛征求任课教师的意见,整合一线教师多年教学经验和实际工作中的体会,摒弃了计算机语言中部分偏、难、过时的语法和技术细节,强调快速上手和实践应用能力的培养,本着易学、实用、有趣、循序渐进的原则精心组织实习内容。本书第 1 版在 2012 年推出并试用,在 5 年的教学实践过程,广泛征求师生意见,调整部分内容,更正部分文字错误,调整部分内容繁杂不易操作的实习项目,更加强调学生应用能力的培养。

全书共安排 14 个实习,每个实习包含 4~5 个部分:实习目的、前导知识、实习任务、综合实例和拓展知识。

实习目的部分给出本次重点需要掌握的知识点;前导知识部分给出实习操作指南和本实验相关的阅读程序题,目的在于使学生实习之前加深对基本知识点的掌握;实习任务部分由浅入深,给出若干实习项目,引导学生循序渐进掌握核心知识点,其中第五个任务带*号,竞赛题目,供学有余力的学生挑战;综合实例部分给出一个贯穿全书的应用实例,旨在使学生能将分散的知识点综合应用,为各个专业学生后期结合各自专业知识应用计算机语言提供样例模板;拓展知识部分给出实习相关的课外补充知识,并安排多个扩展实习任务,针对不同层次学生给出课外学习和探索性实习项目。

书中所有实习内容和代码均符合 C++ 标准(ISO/IEC 14882, 2003 年修订),在 Dev-C++、CodeBlocks、Visual C++ 6.0、Visual C++ 2010 等编译环境下均能

完成。

本书由中国地质大学信息工程学院计算机基础教研室刘传平老师主笔，在修订过程中刘尚一老师和刘燕老师协助完成，并得到了计算机基础教研室一线任课教师的大力支持，在此表示衷心的感谢。另外特别感谢柯凡老师和夏军宝老师对本书提出的许多宝贵建议。

由于编者水平有限，书中疏漏和不足之处，敬请广大师生批评指正。作者的电子邮件地址：liucp@cugb.edu.cn，来信标题请包含 C++ 实习指导书。

编者

2017 年 2 月

目 录

1 开发环境.....	1
1.1 实习目的	1
1.2 前导知识	1
1.3 实习任务	7
1.4 综合实例——学生成绩管理系统设计	10
1.5 拓展知识	10
2 简单程序设计	12
2.1 实习目的	12
2.2 前导知识	12
2.3 实习任务	14
2.4 综合实例——学生成绩管理系统设计	17
2.5 拓展知识	17
3 选择结构程序设计	18
3.1 实习目的	18
3.2 前导知识	18
3.3 实习任务	20
3.4 综合实例——学生成绩管理系统设计	24
3.5 拓展知识	24
4 函数与重载	28
4.1 实习目的	28
4.2 前导知识	28
4.3 实习任务	31
4.4 综合实例——学生成绩管理系统设计	36
4.5 拓展知识	37
5 简单循环结构程序设计	39
5.1 实习目的	39
5.2 前导知识	39
5.3 实习任务	40
5.4 综合实例——学生成绩管理系统设计	45
5.5 拓展知识	45
6 复杂循环结构程序设计	48
6.1 实习目的	48
6.2 前导知识	48
6.3 实习任务	49
6.4 拓展知识	54

7 简单数组程序设计	58
7.1 实习目的	58
7.2 前导知识	58
7.3 实习任务	59
7.4 综合实例——学生成绩管理系统设计	61
7.5 拓展知识	64
8 常用算法	65
8.1 实习目的	65
8.2 前导知识	65
8.3 实习任务	66
8.4 综合实例——学生成绩管理系统设计	71
8.5 拓展知识	73
9 二维数组程序设计	74
9.1 实习目的	74
9.2 前导知识	74
9.3 实习任务	75
9.4 拓展知识	80
10 字符数组和字符串	81
10.1 实习目的	81
10.2 前导知识	81
10.3 实习任务	83
10.4 拓展知识	86
11 复杂字符串程序设计	90
11.1 实习目的	90
11.2 前导知识	90
11.3 实习任务	91
11.4 拓展知识	95
12 类的简单程序设计	97
12.1 实习目的	97
12.2 前导知识	97
12.3 实习任务	98
12.4 综合实例——学生成绩管理系统设计	101
12.5 拓展知识	105
13 文件	107
13.1 实习目的	107
13.2 前导知识	107
13.3 实习任务	107

13.4	综合实例——学生成绩管理系统设计	109
13.5	拓展知识	110
14	指针与动态分配内存.....	114
14.1	实习目的	114
14.2	前导知识	114
14.3	实习任务	118
14.4	综合实例——学生成绩管理系统设计	120
14.5	拓展知识	122

1 开发环境

1.1 实习目的

- 1) 熟悉 Bloodshed Dev-C++集成开发环境的使用;
- 2) 熟练掌握创建工程、编辑源程序、编译程序的基本操作;
- 3) 掌握 C/C++基本程序结构;
- 4) 掌握变量和常量的定义和使用;
- 5) 掌握 cout、cin 的使用。

1.2 前导知识

1.2.1 C/C++集成环境

选择一款 C++编译器, 熟悉该编译器基本操作。代码的创建、修改、编译、链接、执行。建议可选的编译器:

1) Dev-C++ (开源, 内核为 G++): 是一个 Windows 环境下 C/C++的集成开发环境(IDE), 它是一款自由软件, 遵守 GPL 许可协议分发源代码。它集合了 MinGW 等众多自由软件, 并且可以取得最新版本的各种工具支持, 而这一切工作都是来自全球的狂热者所做的工作。Dev-C++是 NOI、NOIP 等比赛的指定工具, 缺点是 Debug 功能弱。由于原开发公司在开发完 4.9.9.2 后停止开发, 所以现在正由其它公司更新开发, 但都基于 4.9.9.2.Dev-C++ 最新版本: 5.11。本校考试软件选用环境。

2) Microsoft Visual C++ 6.0: 安装方便, 占空间少, 但在 64 位操作系统下不能运行。

3) Code::Blocks: 支持现有的 C++标准较好, 有些地方比较灵活。

4) Microsoft Studio 2010: 支持现有的 C++标准较好。

1.2.2 Dev-C++集成环境的使用

1) Dev-C++的启动

执行“开始”→“所有程序”→“Bloodshed Dev-C++”→“Dev-C++”, 启动软件, 运行界面如图 1.1 所示。

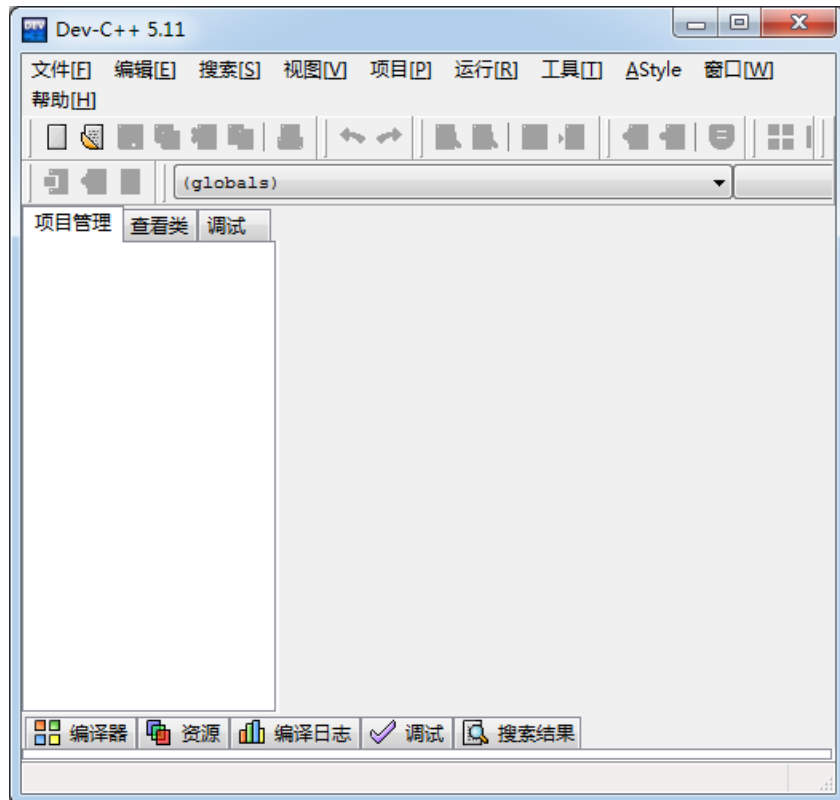


图 1.1 Dev-C++主界面

2) 创建工程

执行“文件”→“新建”→“项目”菜单项，弹出如图 1.2 所示的对话框，在默认的 Basic 选项卡中选择“Console Application”类型，输入项目名，单击“确定”按钮，如图 1.2 所示。

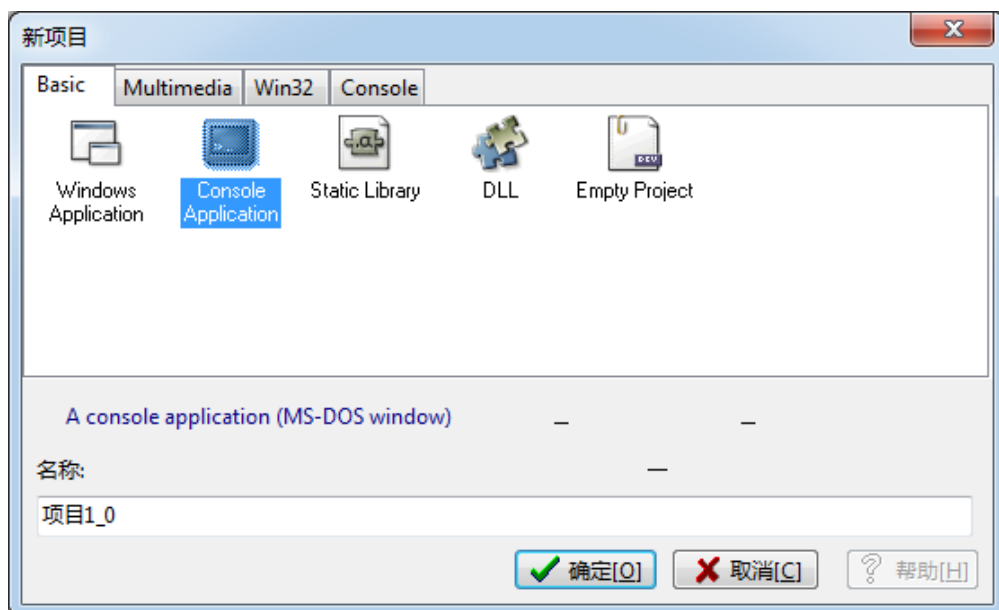


图 1.2 创建工程主界面

弹出“另存为”对话框，在“保存在”的下拉列表中选择保存位置（建议提

前在 D 盘新建一个自己的文件夹，便于管理所有文件)，接受项目文件名和保存类型，单击“保存”按钮，如图 1.3 所示。

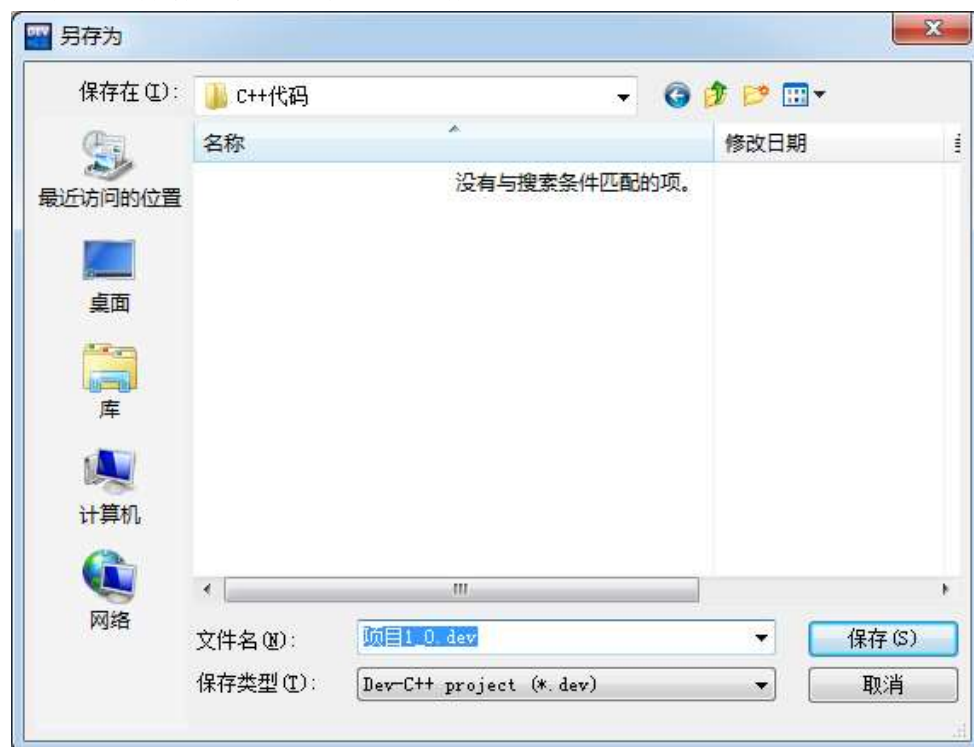


图 1.3 另存为对话框

新建的项目中包含了一个 main.cpp 文件，如图 1.4 所示。

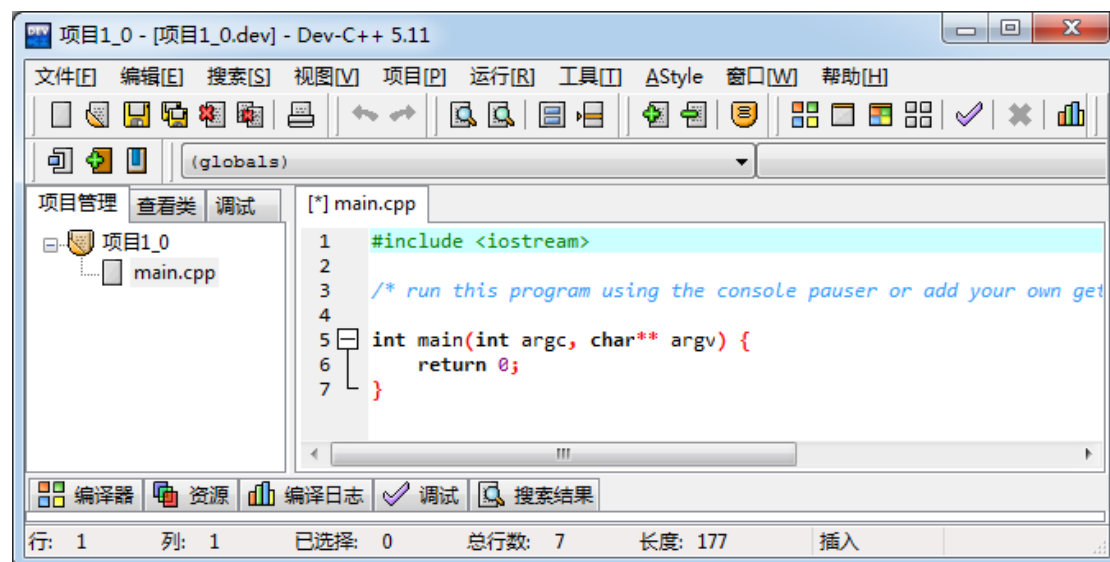


图 1.4 工程文件

为每个作业建立 1 个项目，按照有序的项目名称组织管理所完成的程序。完成 1 个作业后，执行“文件”→“关闭项目”可关闭当前项目，重复创建项目的步骤为下一个作业创建新的项目。

3) 输入代码

在 CPP 文件中输入代码，如图 1.5 所示：

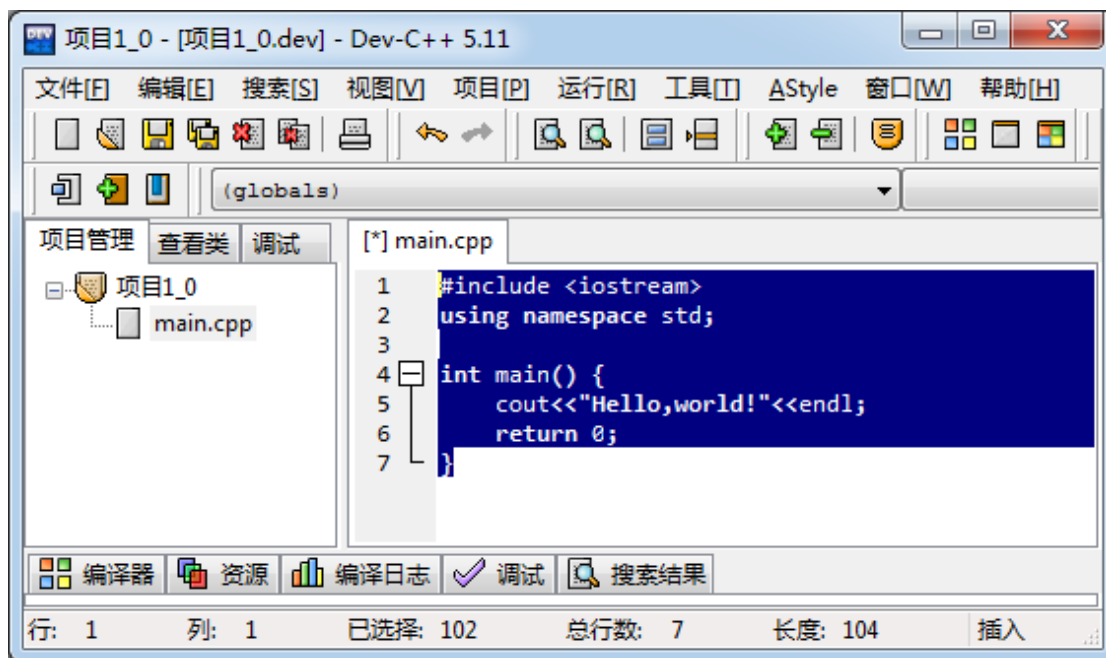



图 1.5 文件编辑界面

程序的左侧显示工程的文件结构，双击某个文件可以打开该文件；在右侧工作区显示文件内容，可以对其进行编辑；下方的状态框主要显示编译结果及错误信息，双击某个错误信息可以定位到错误代码附近。

4) 编译程序

单击“运行”→“编译”子菜单，或者 F9 快捷键，或单击  工具栏按钮，编译程序。在没有保存过文件的情况下，会弹出如图 1.6 所示界面，建议文件名与项目名称相同，便于管理。

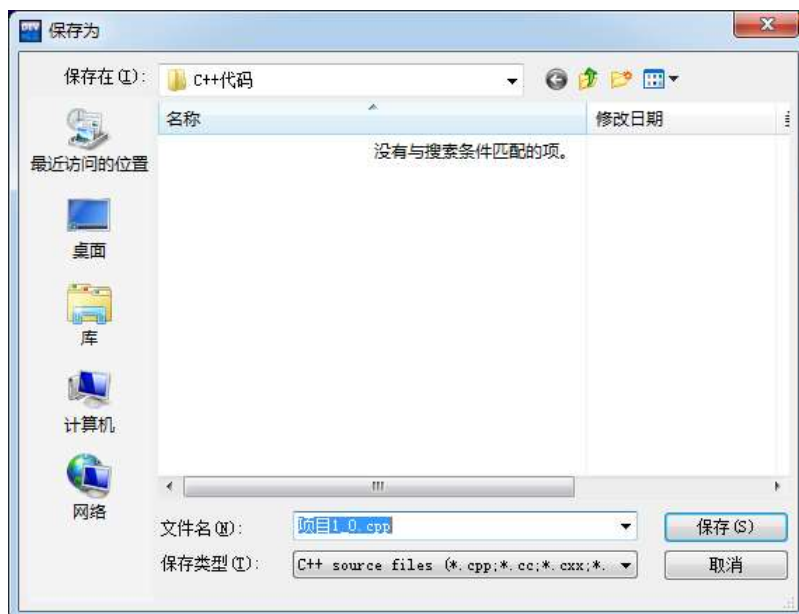


图 1.6 源代码保存窗口

没出错的情况下，生成可执行程序（扩展名为.exe），编译日志窗口如图 1.7

所示。

```
编译项目变更...
-----
- 项目文件名: G:\C++代码\项目1_0.dev
- 编译器名: TDM-GCC 4.9.2 64-bit Release

生成 makefile...
-----
- 文件名: G:\C++代码\Makefile.win


正在处理makefile...
-----
- makefile处理器: C:\Program Files (x86)\Dev-Cpp\MinGW64\bin\mi
- 命令: mingw32-make.exe -f "G:\C++代码\Makefile.win" all

g++.exe -c 项目1_0.cpp -o 项目1_0.o -I"C:/Program Files (x86)/I
g++.exe 项目1_0.o -o 项目1_0.exe -L"C:/Program Files (x86)/Dev-

编译结果...
-----
- 错误: 0
- 警告: 0
- 输出文件名: G:\C++代码\项目1_0.exe
- 输出大小: 1.83212089538574 MiB
- 编译时间: 0.80s
```

图 1.7 程序运行结果窗口

5) 编译程序

执行“运行”→“运行”，或者 F10 快捷键，或者点击 工具栏按钮，将运行程序，如图 1.8 所示。

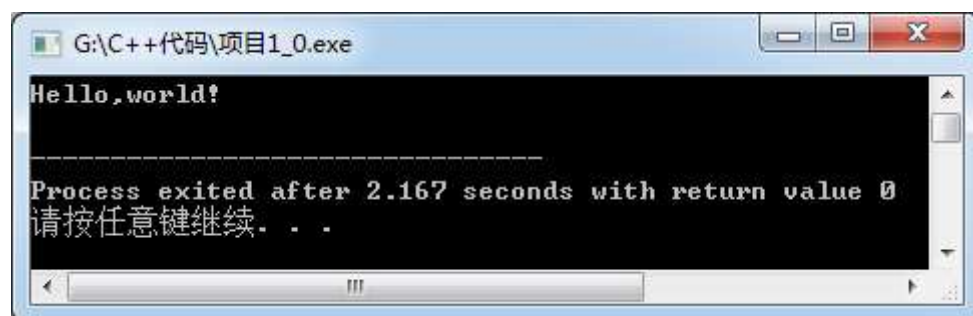


图 1.8 程序运行结果窗口

编译出错时，下方的编译器显示错误信息，双击某个错误信息可以定位到错误代码附近，如图 1.9 所示。修正错误后，重新编译运行程序。排除错误时，应按照从上到下的顺序逐个排除，排除 1 个错误后，重新编译。有时候 1 处代码错误，会导致编译器报告多条错误信息。编译通过后，程序可能没有按照期望的结果输出，这些属于逻辑错误。出现逻辑错误时，需要对源代码进行调试，以发现算法中的缺陷和错误，逻辑错误比语法错误更难捕获。

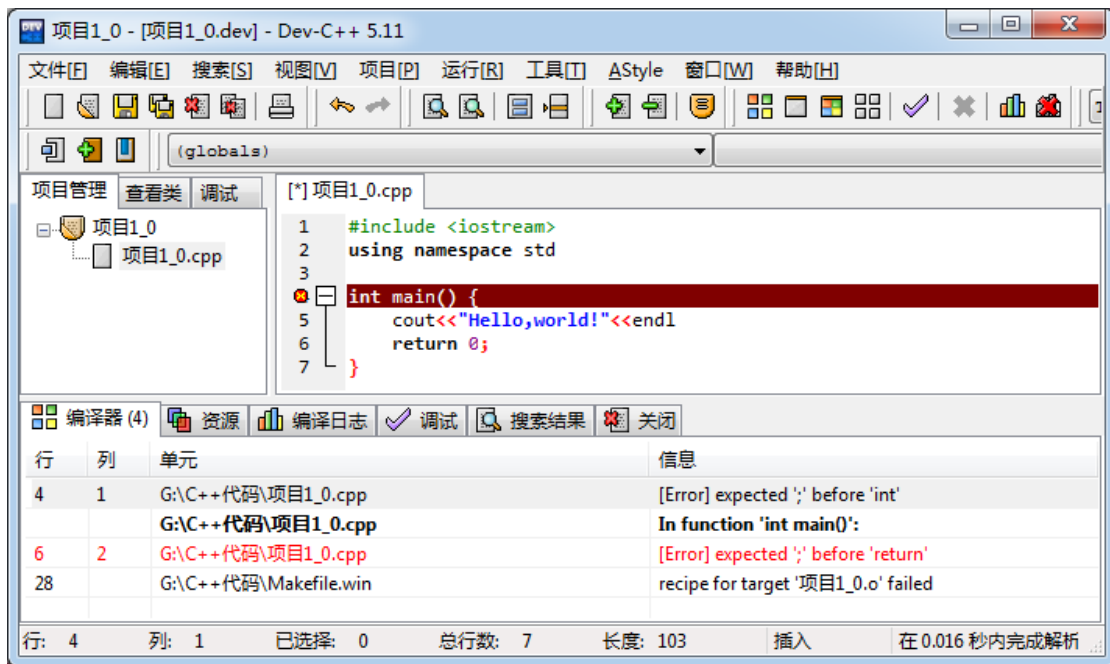


图 1.9 错误信息显示

需要重新打开某个项目时，可以直接在项目文件夹中双击扩展名为 dev 的项目文件；也可以先打开 Dev-C++ 界面再执行“文件”→“打开项目或文件”，在弹出的对话框中选择项目文件所在目录下的扩展名为 dev 的文件，点击“打开”即可。如图 1.10 所示。

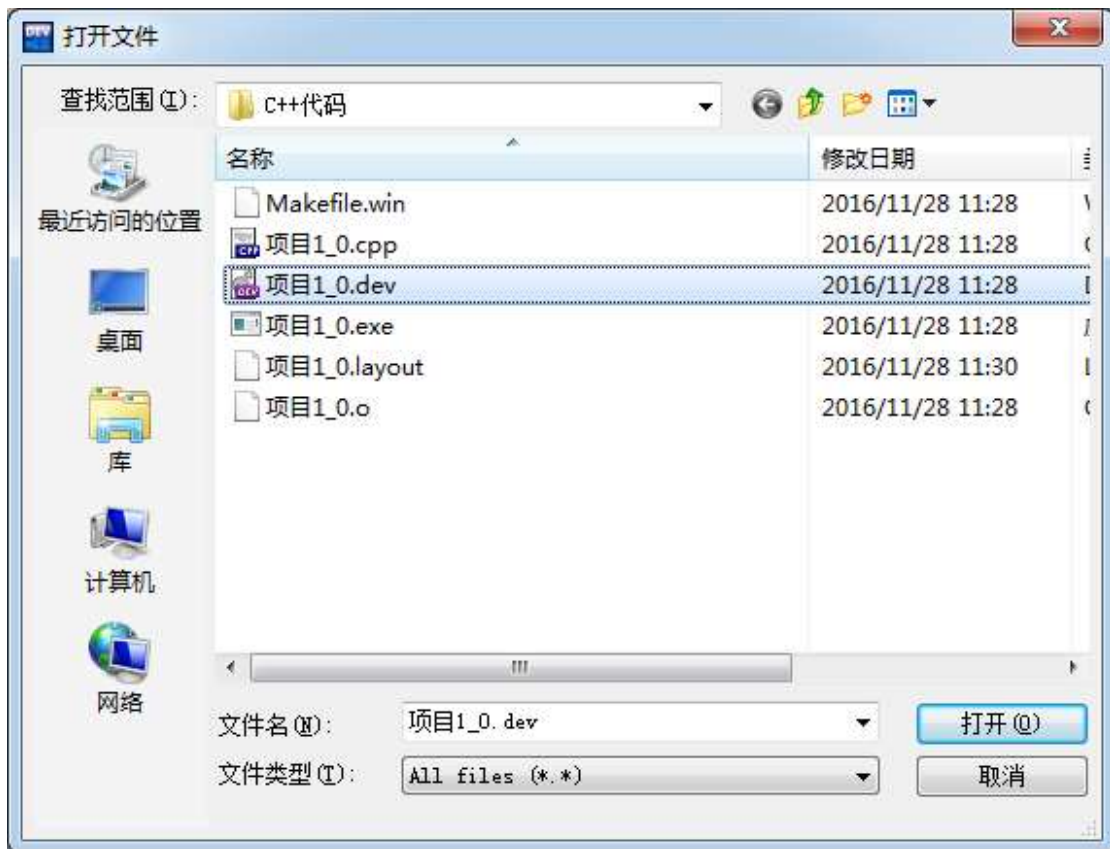


图 1.10 打开项目文件对话框

1.2.3 阅读程序一

阅读下面的程序，写出程序的输出结果。

```
#include <iostream>
using namespace std;
int main( )
{
    cout<<"I love 程序设计"<<"very much!"<<endl;
    cout<<"Programming C/C++ is easy and "
        <<"interesting.\n";
    return 0;
}
```

输出结果: _____

1.2.4 阅读程序二

阅读下面的程序，指出程序中的语法错误，修正所有的错误后，写出程序输出结果。

```
#include <iostream>
using namespace std;
int main
{
    int a=5,int b=5;
    cout<<" The Sum is: << a+b<<endl
    return 0;
}
```

输出结果: _____

1.3 实习任务

1.3.1 实习任务一

编写程序，按照下列格式输出由“*”组成的三角图案。记录程序代码。

**

*

程序代码： _____

1.3.2 实习任务二

编写程序，从键盘输入两个整数，输出这两个整数的和。记录程序代码及其运行结果。

程序代码： _____

运行结果： _____

1.3.3 实习任务三

编写程序，定义 `double` 变量记录圆的半径，定义常量记录圆周率，计算并输出圆的面积和周长。记录程序代码及其运行结果。

程序代码： _____

运行结果： _____

1.3.4 实习任务四

编写程序，输入千米数，输出显示对应的英里数。已知：1 英里=1.60934 千米（用符号常量表示）。记录程序代码及其运行结果。

程序代码： _____

运行结果： _____

1.3.5 实习任务五*

编写程序，从键盘输入一个小写字母，通过程序转换编程对应的大写字母，并输出显示。记录程序代码及其运行结果。

【提示】：每个小写字母 ASCII 码都比对应的大写字母 ASCII 码大 32。

程序代码： _____

运行结果： _____

1.4 综合实例——学生成绩管理系统设计

为了让同学们能把所学 C++语言的基础知识有机结合起来，最终能根据各专业需要独立完成程序设计，在 C++语言程序设计课程的实习过程中，我们安排了以学生成绩管理系统为综合实例的实习，同学们依据所学内容跟着实习指导将一步步完成该系统的设计。

【特别注意】

请同学们向本专业高年级的学长或专业老师了解各自的专业内容以及计算机在本专业的需求，在这门课程的最后请同学们分小组结合各自专业需要和所掌握的计算机语言的基本知识完成结合专业内容的综合实习。

1.5 拓展知识

1、转义字符

C++中通过双引号括起来的字符序列表示字符串常量，对于一些特殊字符，如不可见字符（回车、制表符等），往往通过转义字符来表示这些特殊字符。转

义字符由反斜杠加字符的形式给出，如'\n'表示回车换行。常见的转义字符如下表所示。请在字符串中嵌入一些转义字符，通过 cout 输出，观察输出结果。

表 1.1 转义字符

字符	含义	字符	含义
\n	换行	\a	响铃报警
\r	回车（不换行）	\'	一个单引号
\t	水平制表符	\"	一个双引号
\0	空字符，标志字符串结束	\\	一个反斜线
\b	退格	\ddd	1-3 位八进制 ASCII 码代表的字符
\?	问号	\xhh	1-2 位十六进制 ASCII 码代表的字符

2、命名空间

阅读运行下面的程序，理解后写出程序的输出结果。

```

1      #include <iostream>
2      namespace a
3      {
4          int b=5;
5      }
6      namespace c
7      {
8          int b=8;
9      }
10     int main()
11     {
12         using namespace a;
13         using namespace c;
14         int b=99;
15         std::cout<<b<< std::endl;
16         std::cout<<"a 中的 b\t"<<a::b<<std::endl;
17         std::cout<<"c 中的 b\t"<<c::b<<std::endl;
18         std::cout<<"main 中的 b\t"<<b<<std::endl;
19         return 0;
20     }

```

请注释掉 14、16、17、18 行后再运行，看结果，说明原因。

2 简单程序设计

2.1 实习目的

- 1) 熟悉简单表达式的构建与使用；
- 2) 掌握赋值、算术运算符的使用；
- 3) 熟练掌握输入输出语句的应用。

2.2 前导知识

2.2.1 阅读程序一

阅读下面的程序，写出程序的输出结果。

```
#include <iostream>
using namespace std;
int main()
{
    double numOne = 2.5;
    int numTwo = 3;
    double quotient = numOne/2;
    cout<<"Quotient: "<<quotient<<endl;
    quotient = numTwo/2;
    cout<<"Quotient: "<<quotient<<endl;
    return 0;
}
```

输出结果: _____

2.2.2 阅读程序二

阅读下面的程序，写出程序的输出结果。

```
#include <iostream>
using namespace std;
```

```

int main()
{
    int number = 103;
    int digit, tens, hundreds;
    digit = number % 10;
    tens = (number/10)%10;
    hundreds = (number/100)%10;
    cout<<"Hundreds: "<<hundreds<<", Tens: "<<
        tens<<", Digit: "<<digit<<endl;
    return 0;
}

```

输出结果: _____

2.2.3 阅读程序三

阅读下面的程序，写出程序的输出结果。

```

#include<iostream>
using namespace std;
int main()
{
    int a=0,b=1,c=2,d=3;
    a=b+c; b=a+c; c=a+b; d+=a+b+c;
    cout<<"a="<<a<<"\t";
    cout<<"b="<<b<<"\t";
    cout<<"c="<<c<<"\t";
    cout<<"d="<<d<<endl;
    return 0;
}

```

输出结果: _____

2.2.4 阅读程序四*

阅读下面的程序，写出程序的输出结果。

```

#include<iostream>

```

```
using namespace std;
int main()
{
    int i=5,j=6,a=12,x,y;
    x=i++;
    y=--j;
    x=j++;
    a+=a-a*a;
    cout<<"x="<<x<<"\t"<<"y="<<y<<"\t"<<"a="<<a<<endl;
    return 0;
}
```

输出结果: _____

2.3 实习任务

2.3.1 实习任务一

编写程序，输入两个正整数 m 和 n ，合并成一个正整数 p 。合并的方式是：将 m 的十位和个位数字放在 p 的百位和个位上，将 n 的十位和个位数字放在 p 的千位和十位上。输出显示 m 、 n 、 p 的值。如： $m=57$ ， $n=63$ ，则 $p=6537$ 。记录程序代码及其运行结果。

程序代码: _____

运行结果: _____

2.3.2 实习任务二

编写程序，输入 x 的值，输出多项式 $-4x^3 + 5.8x^2 - 2x + 2.6$ 的值。记录程序代码及其运行结果。

程序代码： _____

运行结果： _____

2.3.3 实习任务三

编程从键盘输入圆柱体的底面半径和高，计算并输出圆柱体的表面积和体积。记录程序代码及其运行结果。

程序代码： _____

运行结果： _____

2.3.4 实习任务四

编写程序，输入 float 类型的摄氏温度 C，输出华氏温度 F，记录程序代码及其运行结果。其中： $F=9/5C+32$

例如：人体温度摄氏 37° C 用华氏温度表示就是： $F=9/5*37+32=98.6$ 度。所以在西方国家看病医生说你的体温是 99 度，千万不要吃惊，他说的是华氏温度~~~。

程序代码： _____

运行结果： _____

2.3.5 实习任务五*

编写程序， 读入一个正 3 位整数，然后输出按位逆序的数字。注意：当输入的数字含有结尾的 0 时，输出不应带有前导的 0。比如输入 700，输出应该是 7。记录程序代码及其运行结果。

输入样例：123

输出样例：321

程序代码： _____

运行结果: _____

2.4 综合实例——学生成绩管理系统设计

根据本节实习所学的输入输出语句，完成学生成绩管理系统主界面的设计。程序执行结果如图 2.1 所示。也可以根据自己的喜好独立设计主界面，不缺少功能项目即可，尽量美观实用方便操作。

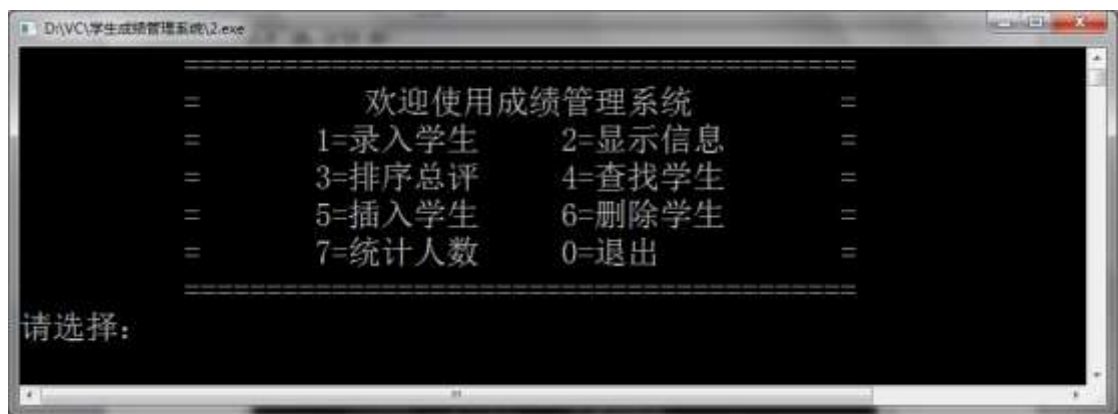


图 2.1 学生成绩管理系统主界面

2.5 拓展知识

通过 `cout` 的格式操纵符，可以控制输出格式。使用格式操纵符，需要包含如下头文件：`#include <iomanip>`

表 2.1 格式操作符

格式操纵符	说明	示例
<code>hex</code> 、 <code>oct</code>	将整数按十六、八进制输出	<code>cout<<hex<<x;</code>
<code>setw()</code>	指定输出数据的宽度	<code>cout<<setw(10)<<x;</code>
<code>setfill()</code>	指定宽度不足时的填充字符	<code>cout<<setfill('0')<<x;</code>
<code>setprecision()</code> 和 <code>fixed</code>	<code>setprecision</code> 和 <code>fixed</code> 配合使用，确定输出小数位数	<code>cout<<fixed<<setprecision(2)<<x;</code>

缺省情况下，通过 `setw()` 设置输出宽度，实际输出宽度不足时，左侧填充空格，若实际输出宽度超过设置宽度时，按照实际宽度输出。

3 选择结构程序设计

3.1 实习目的

- 1) 掌握关系运算符和表达式的构建与使用;
- 2) 掌握逻辑运算符和表达式的构建与使用;
- 3) 掌握 if 语句的使用;
- 4) 理解多分支逻辑处理。

3.2 前导知识

3.2.1 阅读程序一

阅读下面的程序，写出程序的输出结果。

```
#include<iostream>
using namespace std;
int main()
{
    int a,b,c;
    a=10; b=20; c=30;
    if(a>b)
        a=b;
        b=c;
        c=a;
    cout<<a<<" "<<b<<" "<<c<<endl;
    return 0;
}
```

输出结果: _____

3.2.2 阅读程序二

阅读下面的程序，写出程序的输出结果。

```
#include<iostream>
```



```

using namespace std;
int main()
{
    int a, b, c;
    a=2;b=3;c=1;
    if(a>b)
        if(a>c)
            cout<<a;
    else
        cout<<b;
    cout<<c<<endl;
    return 0;
}

```

输出结果: _____

3.2.3 阅读程序三

下面的程序根据输入的百分成绩，输出不同的信息，大于等于 85 输出 Excellent，介于 70 和 85 之间输出 Pass，低于 70 输出 Fail。执行程序 3 次，分别用不同范围的成绩（90、78、60）进行测试，分析程序存在的问题并加以改正。

```

#include <iostream>
using namespace std;
int main()
{
    int grade;
    cout<<"Enter a grade(1-100): ";
    cin>>grade;
    if(grade>=85)
        cout<<"Excellent\n";
    else if(70<=grade<85)
        cout<<"Pass\n";
    else
        cout<<"Fail\n";
}

```

输入 1 (90): _____

输入 2 (78): _____

输入 3 (60): _____

[illegible]

$$y = \begin{cases} x^2 & (x < 0) \\ 1 & (x = 0) \\ x + 2 & (x > 0) \end{cases}$$

[illegible]

运行结果: _____

3.3.3 实习任务三

编写程序，输入 3 个整数，判断并输出以这 3 个整数作为边长能否构建 1 个有效的三角形。

若三个边长为 a 、 b 、 c ，则它们能构成有效三角形需同时满足：

- $a>0$ 且 $b>0$ 且 $c>0$
- 任意 2 边之和大于第 3 边长度

程序代码: _____

运行结果: _____

3.3.4 实习任务四

编写程序，输入 3 个整数，按照由小到大的顺序进行输出。

[illegible]

运行结果: _____

3.3.5 实习任务五*

计算工资：某公司员工的工资计算方法如下：一周内工作时间不超过 40 小时，按正常工作时间计酬；超出 40 小时的工作时间部分，按正常工作时间报酬的 1.5 倍计酬。员工按进公司时间分为新职工和老职工，进公司不少于 5 年的员

3.4 综合实例——学生成绩管理系统设计

根据本节实习所学的选择语句，完成学生成绩管理系统功能的选择。

- 1) 当用户输入 1-7 中的某个数字，输出用户选择的功能并退出系统，如图 3.1 所示：

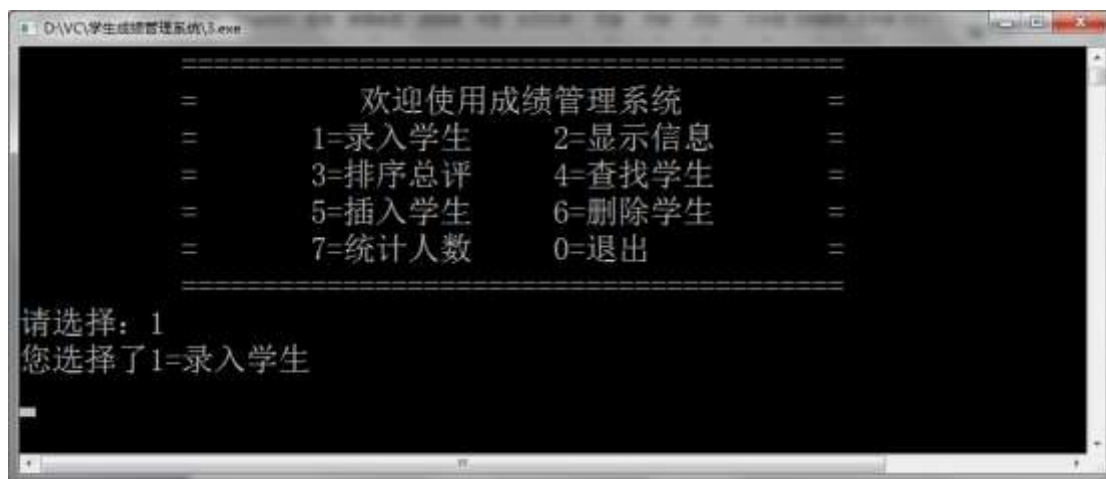


图 3.1 用户选择 1 效果图

- 2) 当用户输入数字 0，直接退出系统；
- 3) 当用户输入 0-7 以外的任何内容，提示用户输入错误并退出系统。

程序执行结果如图 3.2 所示：

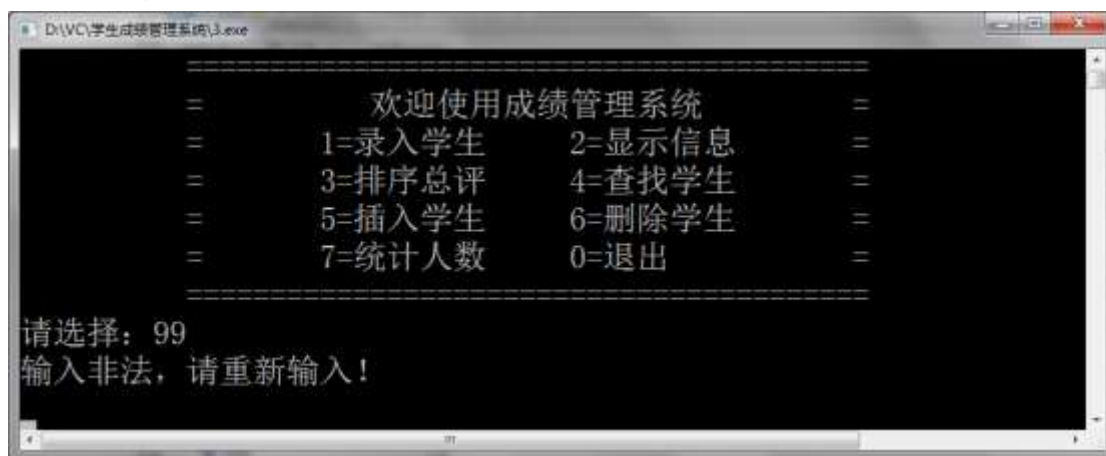


图 3.2 用户非法输入效果图

3.5 拓展知识

- 1) 调试运行下面的程序，记录程序的运行结果。

```
#include<iostream>
#include<iomanip>
```

```

using namespace std;
int main()
{
    int a=0,b=1,c=2,d=3,m,n,k;
    b=a++&&c++;
    d=a++||++c;
    n=b>a;
    k=n&&(m=a);
    cout<<setw(3)<<a<<setw(3)<<b<<setw(3)<<c<<setw(3)
    <<d<<setw(12)<<m<<setw(3)<<n<<setw(3)<<k<<endl;
    return 0;
}

```

- 2) 调试运行下面的程序，记录程序运行结果，如有错误，请予以修改。

```

#include <iostream>
using namespace std;
int main()
{
    int n=5,m=3,x;
    x=m>n?m:n;
    cout<<"x="<<x<<endl;
    if(m=n)    cout<<"m="<<m<<"\t"<<"n="<<n<<endl;
    return 0;
}

```

- 3) 调试运行下面的程序，记录程序的运行结果，如有错误，请予以修改。

```

#include<iostream>
using namespace std;
int main()
{
    int x,y;
    cout<<"please input a number:";
    cin>>x;
    switch(x)
    {

```

```

        case 1: y=10;
        case 2: y=20;
        case 3: y=30;
        default: y=50;
    }
    cout<<y<<endl;
    return 0;
}

```

- 4) 调试运行下面的程序，记录程序的运行结果。

```

#include<iostream>
using namespace std;
int main()
{
    int a, b, c, d, x;
    a=c=0;
    b=1;
    d=20;
    if(a) d=d-10;
    else if(!b)
    if(!c) x=15;
    else x=25;
    cout<<"d="<<d<<endl;
    return 0;
}

```

- 5) switch 语句

请用 switch-case 语句编写如下程序:输入一个同学的成绩,输出相应的等级。

score	等级
90~100	A
80~89	B
70~79	C
60~69	D
0~59	E

- 6) 阅读下面的程序，上机验证程序输出结果。输入数据时，依次输入第一个整数，输入表示运算的字符，再输入第二个整数，中间不用空格、回车等字符。


```

#include <iostream>
using namespace std;
int main()
{
    int num1, num2, result;
    char op;
    cout<<"Enter the expression:";
    cin>>num1;
    cin>>op;
    cin>>num2;
    if(op == '+' )
        result = num1 + num2;
    else if( op == '-')
        result = num1 - num2;
    else if( op == '*' )
        result = num1 * num2;
    else if ( op == '/' )
        result = num1 / num2;
    else
    {
        cout<<"Error input!\n";
        return 0;
    }
    cout<<num1<<op<<num2<<" = "<<result<<"\n";
    return 0;
}

```

4 函数与重载

4.1 实习目的

- 1) 掌握函数的定义、声明和调用方法；
- 2) 掌握函数按值传参和返回值；
- 3) 熟悉函数重载的用法。

4.2 前导知识

4.2.1 常用数学库函数

利用数学库函数，可以完成常用的数学运算，下面列出几个常用的数学函数。
要调用数学库函数，需要包含如下的头文件：

```
#include <cmath>
```

表 4.1 数据库函数

函数原型	说明
int abs(int i)	返回整型参数 i 的绝对值
double fabs(double x)	返回 double 类型参数 x 的绝对值
double sqrt(double x)	返回 \sqrt{x} 的值
double exp(double x)	返回 e^x 的值
double log(double x)	返回 $\log_e x$ 的值
double pow(double x, double y)	返回 x^y 的值
double sin(double x)	返回 x 的正弦值，x 为弧度值
double cos(double x)	返回 x 的余弦值，x 为弧度值

4.2.2 阅读程序一

阅读下面的程序，根据不同的输入写出相应的输出结果。

```
#include <iostream>
using namespace std;
void checkWeight( int height, int weight);
int main()
```

```

{
    int height, weight;
    cout<<"Enter your height(cm): ";
    cin>>height;
    cout<<"Enter your weight(Kg): ";
    cin>>weight;
    checkWeight( height, weight);
    return 0;
}

void checkWeight( int height, int weight )
{
    int idealWeight = height -105;
    if( idealWeight == weight )
        cout<<"Perfect!\n";
    else if( idealWeight > weight )
        cout<<"Slim!\n";
    else
        cout<<"Fatty!\n";
}

```

输入 1: 163 55 输出结果: _____

输入 2: 150 60 输出结果: _____

输入 3: 160 55 输出结果: _____

4.2.3 阅读程序二

阅读下面的程序，写出程序的输出结果。

```

#include<iostream>

using namespace std;

void add(int x,int y,int z);

int main()
{
    int x=2,y=3,z=0;

    cout<<"(1)x="<<x<<"\t"<<"y="<<y<<"\t"<<"z="<<z<<endl;

```

```

    add(x,y,z);

    cout<<"(3)x="<<x<<"\t"<<"y="<<y<<"\t"<<"z="<<z<<endl;

    return 0;
}

void add(int x,int y,int z)
{
    z=x+y;

    x=x*x;

    y=y*y;

    cout<<"(2)x="<<x<<"\t"<<"y="<<y<<"\t"<<"z="<<z<<endl;
}

```

输出结果: _____

4.2.4 阅读程序三

阅读下面的程序，写出程序的输出结果。

```

#include <iostream>

using namespace std;

int fun(int a);

double fun(double a);

char fun(char a);

int main()
{
    cout<<fun(3)<<endl;
    cout<<fun(3.6)<<endl;
    cout<<fun('A')<<endl;
    cout<<fun('g')<<endl;
    return 0;
}

int fun(int a)

```

```

{
    return a/2;
}
double fun(double a)
{
    return a/2;
}
char fun(char a)
{
    char result=a;
    if(a>='a' && a<='z')
        result=a-32;
    if(a>='A' && a<='Z')
        result=a+32;
    return result;
}

```

运行结果: _____

4.3 实习任务

4.3.1 实习任务一

编写函数 `getAbs`，传入 1 个 `double` 类型参数，返回其绝对值。在主程序中输入实数，调用 `getAbs` 计算并返回其绝对值，并在主函数中输出绝对值。函数声明如下：

```
double getAbs(double x);
```

`getAbs` 函数代码: _____

主程序代码：

4.3.2 实习任务二

编写函数 `getArea`，传入三角形的三条边长作为参数，利用海伦公式计算三角形的面积并返回。

【海伦公式】若三角形的三个边长为 a 、 b 、 c ，设 $s=(a+b+c)/2$ ，则三角形面积为：
$$\text{area} = \sqrt{s(s-a)(s-b)(s-c)}$$

在主程序中输入三个边长，调用 `getArea` 计算并显示三角形面积。函数声明如下：

```
#include <iostream>
#include <cmath>
using namespace std;
double getArea( double a, double b, double c );
getArea 函数代码：
```

【程序容错】

完善该实习任务中的容错处理，只有当三角形为有效三角形时，调用面积计算函数才有意义，否则程序会出现异常，应增加容错处理。

编写函数 `validate`，传入 3 个边长作为参数，判断是否构成有效三角形，返回 `bool` 型值，返回 `true` 表示构成有效三角形，返回 `false` 表示无效三角形。`validate` 函数的声明如下：

```
bool validate (double a, double b, double c);
```

`validate` 函数代码： _____

在主程序中输入 3 个边长，调用 `validate` 函数，若是无效三角形则输出提示信息然后结束程序；若是有效三角形，调用 `getArea` 计算并显示面积。测试的主程序如下，分别输入 3 组不同的边长，记录程序输出结果：

```
#include <iostream>
#include <cmath>
using namespace std;
int main()
{
    double a, b, c;
    cout<<"Enter 3 edges for the triangle: ";
    cin>>a>>b>>c;
    if(validate( a, b, c)== false )
        cout<<"Invlirate Triangle\n";
    else
        cout<<"The area is : "<< getArea(a,b,c)<<endl;
```

输入数据: _____

运行结果: _____

[illegible]


```

#include <iostream>
using namespace std;
bool narcissistic( int number );
int main()
{
    int m;
    cout<<"请输入一个三位正整数: ";
    cin>>m;
    if ( narcissistic(m) )
        cout<<m<<" is a narcissistic number."<<endl;
    else
        cout<<m<<" is not a narcissistic number."<<endl;
    return 0;
}

```

narcissistic 函数代码: _____

4.3.5 实习任务五*

阶梯电价：为了提倡居民节约用电，某省电力公司执行“阶梯电价”，安装一户一表的居民用户电价分为两个“阶梯”：月用电量 50 千瓦时（含 50 千瓦时）

输入格式：输入在一行中给出某用户的月用电量（单位：千瓦时）。

输出格式：在一行中输出该用户应支付的电费（元），结果保留两位小数，格式如：“cost = 应付电费值”；若用电量小于 0，则输出"Invalid Value!"。

输出格式：在一行中输出该用户应支付的电费（元），结果保留两位小数，
例如：“cost = 应付电费值”；若用电量小于 0，则输出“Invalid Value!”。

输出样例 1: cost = 5.30

输出样例 2: cost = 55.50

This image shows a single sheet of white paper with horizontal blue ruling lines. The lines are evenly spaced and run across the width of the page. There are approximately 20 lines visible. The paper has a slight shadow on the right side, suggesting it's resting on a surface.

根据本节实习所学的函数，将学生成绩管理系统的主界面写成独立的函数：

```
void Menu();
```

注意：Menu 中仅仅是主界面输出的内容，功能选择代码仍放在主函数中。
独立主界面函数的好处是为以后在需要的时候随时调用，避免代码重复。程序执行效果同第 3 章实习。

4.5 拓展知识

- 1) 【提示菜单】阅读下面的程序，上机验证程序输出结果。

```
#include <iostream>
using namespace std;
int menu()
{
    int choice;
    cout<<"1) add two number\n";
    cout<<"2) sub two number\n";
    cout<<"3) multiply two number\n";
    cout<<"4) divide two number\n";
    cout<<"0) quit\n";
    cout<<"Enter your choice:\n";
    cin>>choice;
    return choice;
}
int main()
{
    int num1=20, num2=6;
    int choice;
    int result;
    choice = menu();
    if( choice == 1)
        result = num1 + num2;
    else if( choice == 2)
        result = num1 - num2;
    else if (choice ==3 )
        result = num1 * num2;
```

```

else if (choice == 4 )
    result = num1 / num2;
else //非预期情况，通过 return 语句直接结束程序
    return 0;
cout<<"result: "<<result<<endl;
return 0;
}

```

2) 默认参数

调试运行下面的程序，运行程序 3 次，输入不同的值，记录程序的运行结果。

```

#include <iostream>
using namespace std;
bool check(int score, int baseLine = 60);
int main()
{
    int score;
    cout<<"Enter your score: ";
    cin>>score;
    if( check(score) == true)
        cout<<"Passed!\n";
    else
        cout<<"Failed!\n";
    if( check(score, 70) == true)
        cout<<"Passed!\n";
    else
        cout<<"Failed!\n";
    return 0;
}
bool check( int score, int baseLine)
{
    if( score >= baseLine )
        return true;
    return false;
}

```

5 简单循环结构程序设计

5.1 实习目的

- 1) 掌握 while 和 do~while 循环的使用;
- 2) 掌握 for 循环的使用;
- 3) 掌握 break 语句的用法;
- 4) 熟悉循环结构程序设计方法。

5.2 前导知识

5.2.1 阅读程序一

阅读下面的程序，写出程序的输出结果。

```
#include<iostream>
#include<iomanip>
using namespace std;
int main()
{
    int i,j;
    for(i=j=1;i<5;i++)
    {
        if(i%2==0)
            break;
        j++;
        cout<<setw(10)<<i<<setw(10)<<j<<endl;
    }
    return 0;
}
```

运行结果: _____

}

运行结果: _____

5.3.4 实习任务四

编写函数 `isPrime`，判断某个正整数（大于等于 2）是否为素数，函数返回 `bool` 类型参数。在主程序中，调用该函数，记录程序代码。

【素数的定义】对于正整数 n ，若 n 只能被 1 和 n 整除，则 n 为素数。判断 1 个正整数 n 是否为素数的方法如下：令 $r = [\sqrt{n}]$ ， $[x]$ 表示取不超过 x 的最大整数，若 n 不能被 $2 \sim r$ 之间的所有整数整除，则 n 是素数。

```
#include <iostream>
using namespace std;
#include <cmath>
```



```
bool isPrime(int num);
```

```
int main()
```

```
{
```

```
    int num;
```

```
    cin>>num;
```

```
        return 0;
```

```
}
```

```
bool isPrime(int num)
```

```
{
```

```
}
```

```
运行结果: _____
```

5.3.5 实习任务五 *

卡拉兹(Callatz)猜想：对任何一个自然数 n ，如果它是偶数，那么把它砍掉一半；如果它是奇数，那么把 $(3n+1)$ 砍掉一半。这样一直反复砍下去，最后一定在某一步得到 $n=1$ 。卡拉兹在 1950 年的世界数学家大会上公布了这个猜想，传说当时耶鲁大学师生齐动员，拼命想证明这个貌似很傻很天真的命题，结果闹得学生们无心学业，一心只证 $(3n+1)$ ，以至于有人说这是一个阴谋，卡拉兹是在蓄

意延缓美国数学界教学与科研的进展……

我们今天的题目不是证明卡拉兹猜想，而是对给定的任一不超过 1000 的正整数 n ，简单地数一下，需要多少步（砍几下）才能得到 $n=1$ ？

输入格式：每个测试输入包含 1 个测试用例，即给出自然数 n 的值。

输出格式：输出从 n 计算到 1 需要的步数。

输入样例：3

输出样例：5

程序代码: _____

5.4 综合实例——学生成绩管理系统设计

根据本节实习所学的循环语句，完成学生成绩管理系统功能的多次选择。

- 1) 当用户输入 1-7 中的某个数字，输出用户选择的功能后显示系统主界面；
- 2) 当用户输入数字 0，直接退出系统；
- 3) 当用户输入 0-7 以外的任何内容，提示用户输入错误并显示系统主界面。

程序执行结果如图 5.1 所示：

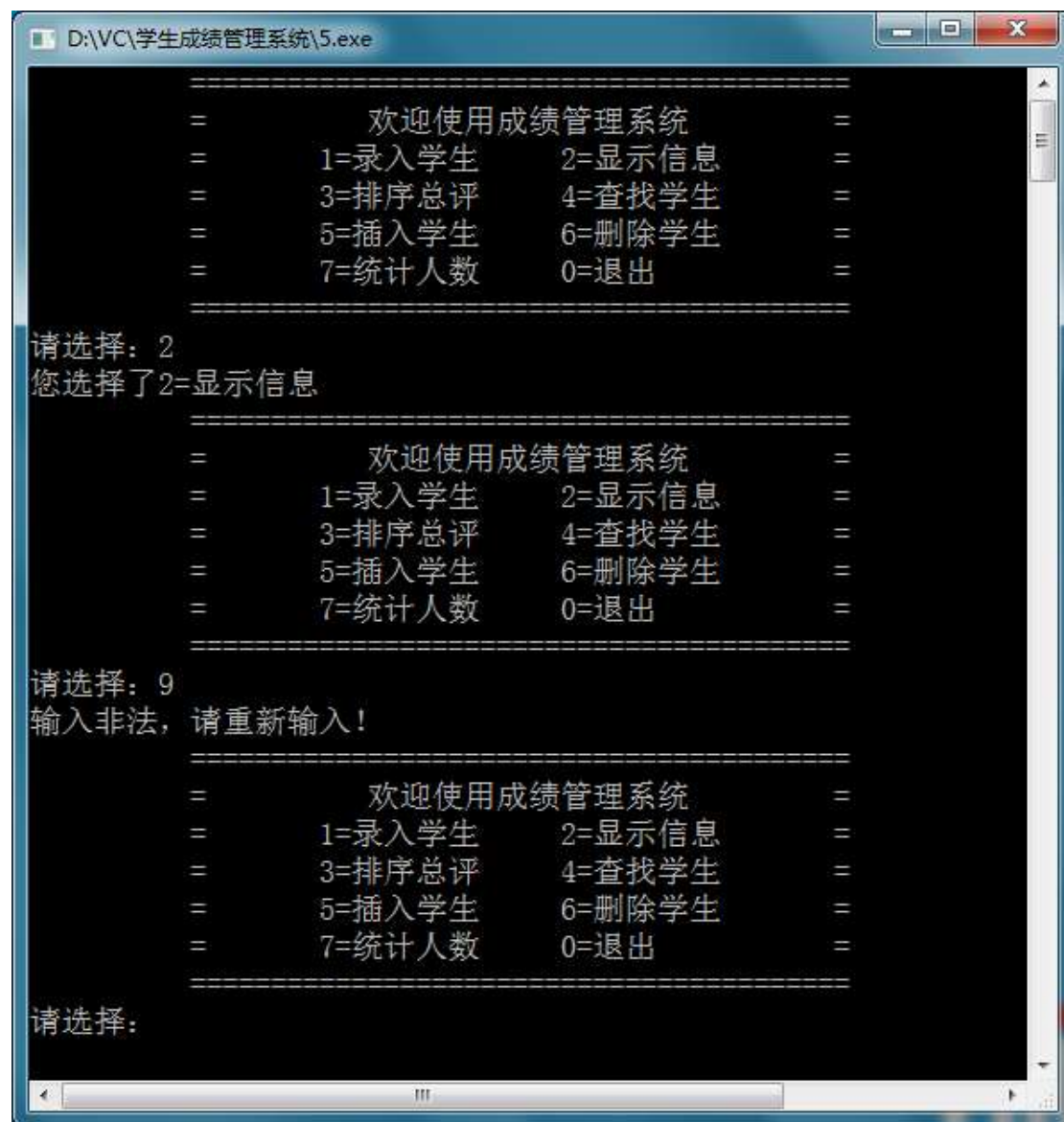


图 5.1 功能选择演示图

5.5 拓展知识

- 1) 编写函数，传入 double 型参数 x 和整型参数 k(k 可为负数)，计算并返回 x^k 。

函数原型如下：

double f(double x, int k);

- 2) 利用下面的泰勒级数公式，计算 e^x 的近似值，用循环对级数项累加求和，直到最后一项的绝对值小于 10^{-6} 时，认为达到精度要求。

$$e^x = 1 + \frac{x}{1!} + \frac{x^2}{2!} + \frac{x^3}{3!} + \cdots + \frac{x^n}{n!}$$

【提示】计算第 n 项时，无需重新计算 $n!$ ，在上一项的计算中，已经计算出 $(n-1)!$ ，而 $n! = (n-1)! * n$ 。

- 3) 阅读下面的程序，写出程序的输出结果。

```
#include<iostream>
#include<iomanip>
using namespace std;
int main()
{
    int i,j;
    for(i=j=1;i<5;i++)
    {
        if(i%2==0)
            continue;
        j++;
        cout<<setw(10)<<i<<setw(10)<<j<<endl;
    }
    return 0;
}
```

- 4) 最大公约数的辗转相除法

非负整数 m 和 n (m 和 n 不能同时为 0) 的最大公约数计算方法如下：

$$\gcd(m, n) = \begin{cases} m & \text{若 } n = 0 \\ \gcd(n, m \% n) & \text{若 } n \neq 0 \end{cases}$$

- 5) 最大公约数的递归调用实现

辗转相除法除了可以用循环实现，还可以通过递归调用的方法实现。递归调用指在函数中再次调用函数自身。

下面的函数是辗转相除法的递归实现，编写主程序调用该函数，测试程序功能是否正常。

```
int gcd(int m, int n)
{
    if(n==0)
        return m;
    return gcd(n, m%n);
}
```

递归函数不能无限调用下去，必须设置 1 个终止条件，以结束函数的递归调用，gcd 函数递归调用的终止条件为 n 等于 0，此时 m 即为最大公约数。

6 复杂循环结构程序设计

6.1 实习目的

- 1) 掌握多重循环程序设计的基本方法;
- 2) 学会合理运用选择结构和循环结构的组合解决较复杂的问题。

6.2 前导知识

6.2.1 阅读程序一

调试运行下面的程序，写出程序的运行结果。

```
#include<iostream>
using namespace std;
int main()
{
    int i, j, a=0;
    for(i=0; i<2;i++)
    {
        for(j=0; j<4;j++)
        {
            if(j%2)
                break;
            a++;
        }
        a++;
    }
    cout<<"a="<<a<<endl;
    return 0;
}
```

运行结果: _____

6.3 实习任务

6.3.1 实习任务一

用 100 匹马驮 100 担货物，大马驮 3 担，中马驮 2 担，2 匹小马驮 1 担，问大、中、小马各有多少匹？编写程序，输出所有的可能，大、中、小马的匹数可以为 0。

程序代码: _____

[illegible]

运行结果: _____

6.3.2 实习任务二

通过循环计算并输出从 100 开始到 200 之间的素数，记录程序代码。输出时每 5 个素数占 1 行。

```
#include <iostream>
```

```
#include <iomanip>
```

[illegible]
$$\}$$


```

using namespace std;

int main()
{
    int a,i,m;
    for(a=1;a<=1000;a++)

    _____

    _____

    _____

    _____

    _____

    _____

    _____

    _____

    _____

    _____

    return 0;
}

```

6.3.4 实习任务四

请输出下列图形：

```

1
2  2
3  3  3
4  4  4  4
5  5  5  5  5

```

.....

要求：从键盘输入行数。

程序代码： _____

[illegible]

6.3.5 实习任务五*

打印沙漏：本题要求你写个程序把给定的符号打印成沙漏的形状。

所谓“沙漏形状”，是指每行输出奇数个符号；各行符号中心对齐；相邻两行符号数差 2；符号数先从大到小顺序递减到 1，再从小到大顺序递增；首尾符号数相等。给定任意 N 个符号，不一定能正好组成一个沙漏。要求打印出的沙漏能用掉尽可能多的符号。

输入格式：输入在一行给出 1 个正整数 N (≤ 1000) 和一个符号，中间以空格分隔。

输出格式：首先打印出由给定符号组成的最大的沙漏形状，最后在一行中输出剩下没用掉的符号数。

输入样例：19 *

输出样例：

*

2

程序代码: _____

[illegible]

6.4 拓展知识

1) 编写程序，实现打印出以下图案

```
      *
    * * *
  * * * * *
* * * * * * *
  * * * * *
    * * *
      *
```

2) 验证随机数序列

编写程序，在程序中首先调用 `srand` 函数并设置固定的随机数种子，再通过循环调用 `rand` 函数输出前 100 个随机数。重复执行程序 3 次，验证输出结果。

【随机数生成】

很多程序都需要通过生成随机数，以模拟随机过程。系统构建随机数的基本方法如下，系统排列 0~RAND_MAX 之间的所有整数构成 1 个随机数序列（RAND_MAX 是系统预设的可生成最大随机数的上限）。0~RAND_MAX 的不同排列的数量很多，每种排列都是 1 个随机数序列，不同的序列通过不同的种子加以标识。在程序中，设置随机数种子后，便确定了 1 个随机数序列，之后生成随机数的过程便是顺序从设定的随机数序列中取数而已。

通过调用库函数 `srand` 和 `rand` 可以实现随机数的生成。`srand` 函数用于在程序中设置随机数种子，也就是确定一个随机数序列，`rand` 函数依次从指定的随机数序列中取 1 个整数。

如果在程序运行时，每次设置相同的种子，则每次确定的随机数序列是相同的，程序每次运行时取得的随机数是相同的，这种现象称为伪随机数。为了使程序每次运行时获得不同的随机数序列，一种常见的做法是程序启动时调用 `srand` 并传入系统当前时间作为参数，需要包含<ctime>头文件。

```
srand((unsigned) time(NULL));
```

`time(NULL)`函数调用获取系统当前的时间，并转换为无符号整数（1970 年 1 月 1 日的 UTC 时间从 0 时 0 分 0 秒算起到现在所经过的秒数），由于每次运行程序的时间不同，所以得到的随机数序列不同，从而实现真正的随机数生成。

`rand` 函数返回 0~RAND_MAX 之间的整数，如果程序需要生成指定范围的

随机数，需要相应的处理。比如，要生成 2 位随机整数（10~99），可通过下面的方法获得：

```
rand()%90+10;
```

如果要生成 0~1.0 之间的 double 类型浮点随机数，可通过下面的方法获得：

```
1.0*rand() / RAND_MAX;
```

3) 小学生算术测验

编写程序之前，一定要搞清楚要实现的功能和约束条件。本题要求编写针对小学生的四则运算测验（加、减、乘），系统根据用户的选择执行加、减、乘法练习或退出，每次随机生成 2 位整数进行运算（减法运算时被减数要大于减数），每次答题后及时给出正误提示信息，用户选择退出后，给出答题总数和答题正确数量信息。

以下是实现题目要求的参考程序代码，请认真阅读并上机调试程序，验证程序功能是否正确。

```
#include <iostream>
using namespace std;
#include <ctime>
int menu();
bool answerQuestion(int choice, int num1, int num2);
int main()
{
    int questionNum=0;
    int correctNum=0;
    srand((unsigned) time(NULL)); //设置随机数序列的种子
    while(true)
    {
        int choice=menu();
        if(choice==0)
            break;
        int num1, num2;
        num1=rand()%90+10;    //得到 10~99 的整数
        num2=rand()%90+10;
        if(answerQuestion(choice,num1,num2)==true)
            correctNum++;
    }
}
```

```

        questionNum++;
    }
    if(questionNum==0)
    {
        cout<<"You didn't answer any question!\n";
    }
    else
    {
        cout<<"Answer questions: "<<questionNum<<
            ", correct: "<<correctNum<<"\n";
    }
    return 0;
}
int menu()
{
    int choice;
    cout<<"1) add two number\n";
    cout<<"2) sub two number\n";
    cout<<"3) multiply two number\n";
    cout<<"0) quit\n";
    cout<<"Enter your choice:\n";
    cin>>choice;
    return choice;
}
bool answerQuestion(int choice, int num1, int num2)
{
    int result,answer;
    char operation;
    if( choice==2 && num1<num2)
    {
        int temp=num1;
        num1=num2;
        num2=temp;
    }

```

```

    }
    switch(choice)
    {
    case 1:
        result=num1+num2;
        operation='+';
        break;
    case 2:
        result=num1-num2;
        operation='-';
        break;
    case 3:
        result=num1*num2;
        operation='*';
        break;
    }
    cout<<num1<<operation<<num2<<"=?";
    cin>>answer;
    if(result==answer)
    {
        cout<<"Great! Your answer is absolutely correct!\n";
        return true;
    }
    cout<<"Sorry!\n";
    return false;
}

```

【试一试】试着按下面的需求修改程序，上机调试并验证程序是否正确：①每次练习固定完成 10 道习题；②在 main 函数中对用户输入的 choice 值进行容错，如果输入值不在 1~4 范围内，直接进入下一轮循环。

7 简单数组程序设计

7.1 实习目的

- 1) 掌握一维数组的定义、使用；
- 2) 掌握通过循环遍历一维数组元素的方法。

7.2 前导知识

7.2.1 阅读程序一

阅读下面的程序，写出程序的输出结果。

```
#include <iostream>
#include<iomanip>
using namespace std;
int main()
{
    int a[10]={ 1,2,3,4,5,6,7,8,9,10},i,j,t;
    for(i=0,j=9;i<j;i++,j--)
    {
        t=a[i];
        a[i]=a[j];
        a[j]=t;
    }
    for(i=0;i<10;i++)
        cout<<setw(5)<<a[i];
    cout<<endl;
    return 0;
}
```

运行结果： _____

7.3 实习任务

7.3.1 实习任务一

下面的程序输入某个同学的各科成绩并计算平均分，请将程序补充完整。

```
#include <iostream>
using namespace std;
int main()
{
    const int COURSE_NUM=5;
    int scores[COURSE_NUM];
    double aveScore=0;
    int i;
    for(i=0; i<COURSE_NUM; ++i) //通过键盘输入成绩数据
    {
        _____
    }
    _____
    _____
    _____

    cout<<"Average Score: "<<aveScore<<"\n";
    return 0;
}
```

运行结果： _____

7.3.2 实习任务二

本题要求实现求 Fabonacci 数列项的函数。Fabonacci 数列的定义如下：

$f(n)=f(n-2)+f(n-1)$ ($n \geq 2$)，其中： $f(0)=0$ ， $f(1)=1$ 。

测试程序样例：

```
#include <iostream>
using namespace std;
```

[illegible]

7.3.3 实习任务三

编写程序，将数组中的元素逆序。程序应包括以下几个部分：通过键盘输入数据、逆序处理、输出逆序后的结果。

【提示】定义 2 个索引变量初始指向数组的第 1 个元素和最后 1 个元素，交换它们，指向开始位置的索引变量递增，指向结束位置的索引递减。重复这 1 过程，直到 2 个索引位置重叠或错位。

7.3.4 实习任务四

编写程序，删除数组中指定位置处的元素。删除数组中的某个元素时，其后的所有元素需要顺序向前移动。

7.4 综合实例——学生成绩管理系统设计

根据本节实习所学的数组，完成学生成绩管理系统中简单数据处理功能：

1) 定义全局常整数，用以记录班级最多人数：

```
const int MAX_STUDENT_NUM=40; //测试时可以用 5
```

2) 在主函数中定义实型数组 `stu` 用以存储学生的成绩；并定义整型变量

currentN 用以记录当前学生实际人数:

```
float stu[MAX_STUDENT_NUM];
```

```
int currentN=0;
```

- 3) 自定义输入函数, 要求根据用户的需要可以连续输入, 直到达到上限;
并要求检测成绩是否在 0-100 之间:

```
void Input(float stu[],int &num);
```

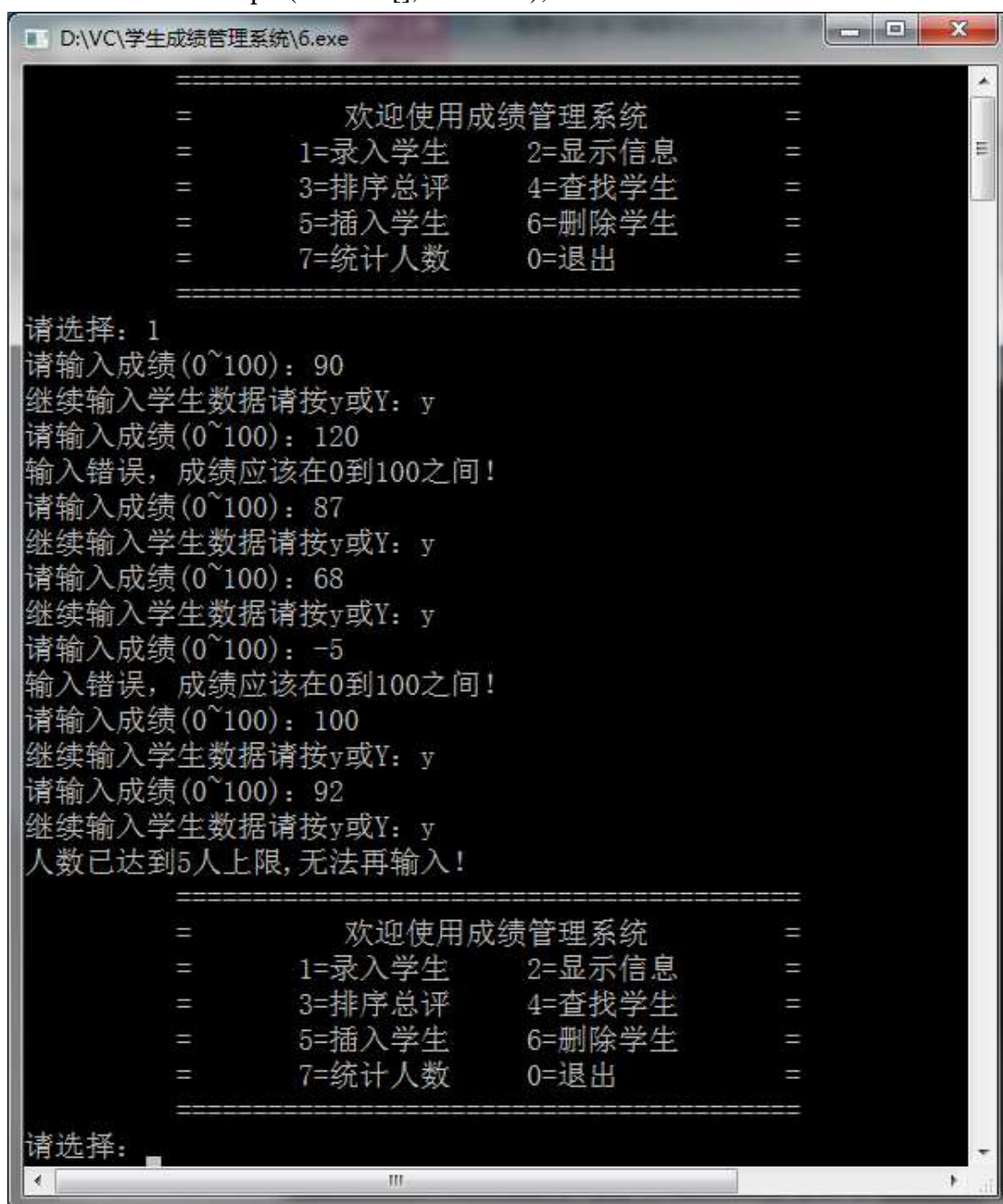


图 7.1 成绩输入演示图

注意: 如果 1 录入学生函数体设计合理, 5 插入学生可以和 1 录入学生共用一个函数体。

4) 自定义输出函数，要求输出美观，并保留一位小数：

```
void Output(float stu[],int num);
```

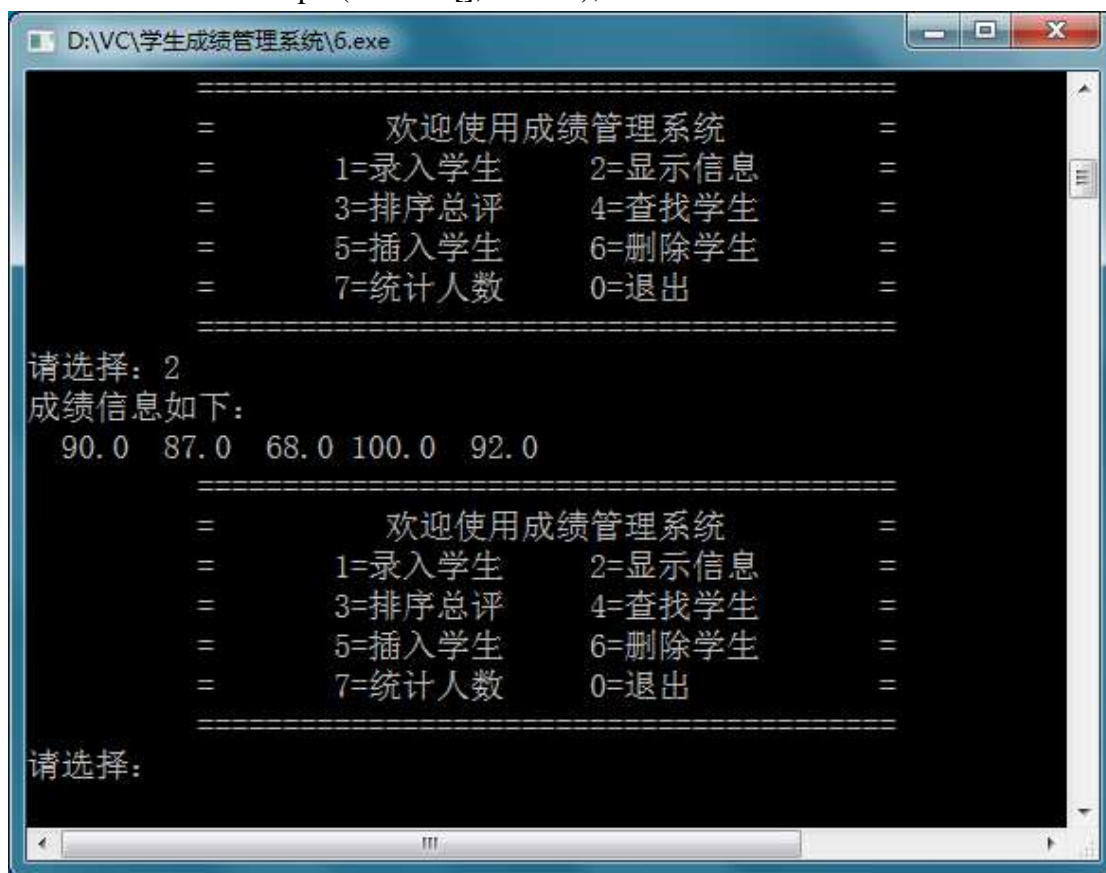


图 7.2 成绩输出演示图

5) 自定义统计函数，统计各分数段的人数：

```
void Statis(float stu[],int num);
```

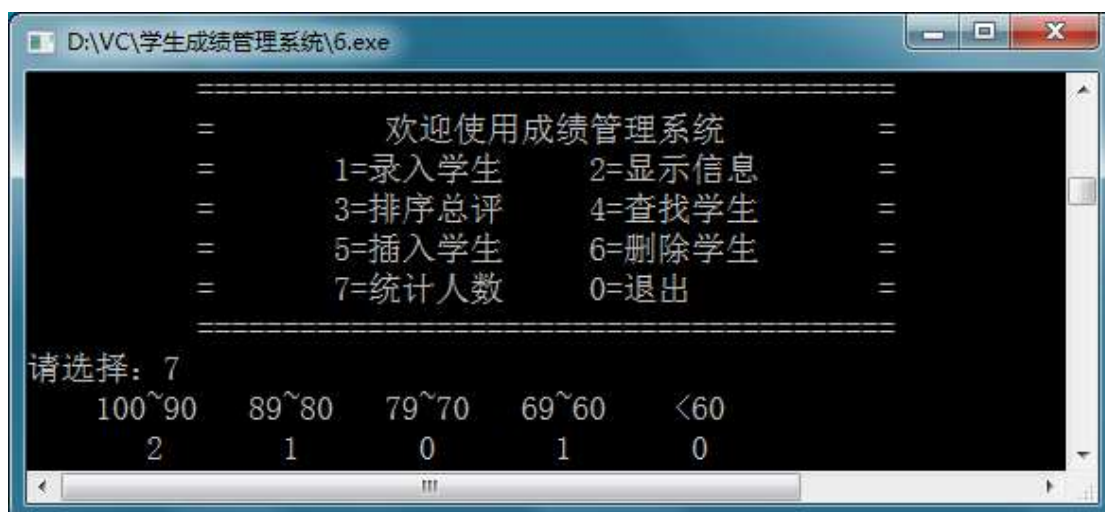


图 7.3 统计成绩演示图

7.5 拓展知识

- 1) 编写程序，定义数组，输入数组中各元素的值，输出数组中的最大值和最小值及其下标。
- 2) 编写程序实现裁判打分。由于各裁判打分时可能存在某些主观因素，因此在计算平时分时通常去掉一个最高分和一个最低分，程序读入 7 个裁判所打的分数，去掉一个最高分和一个最低分，计算剩余 5 个裁判的平均分。

8 常用算法

8.1 实习目的

- 1) 掌握折半查找算法;
- 2) 掌握常用的查找算法;
- 3) 学会合理运用选择结构和循环结构的组合解决较复杂的算法问题。

8.2 前导知识

8.2.1 程序填空题

以下程序分别在 a 数组和 b 数组中放入 a_{n+1} 和 b_{n+1} 个由小到大排好序的数据，程序把两个数组中的数按由小到大的顺序合并到 c 数组中，请填空。

```
#include <iostream>
using namespace std;
int main()
{
    int a[10]={ 1,2,5,8,9,10},an=5;
    int b[10]={ 1,3,4,8,12,18},bn=5;
    int i,j,k,c[20],max=9999;
    a[an+1]=b[bn+1]=max;
    i=j=k=0;
    while((a[i]!=max)||(b[j]!=max))
    {
        if(a[i]<b[j])
        {
            c[k]= 【1】 ;
            k++;
            【2】
        }
        else
```

```

        {
            c[k]= 【3】;
            k++;
            【4】
        }
    }
    for(i=0;i<k;i++)
        cout<<c[i]<<" ";
    cout<<endl;
    return 0;
}

```

8.3 实习任务

8.3.1 实习任务一

编写程序，在有序的数列中采用折半法查找数据，如果找到所需的数据，程序输出该数据的下标；如果没有找到所需的数据，输出“没有找到”。

【折半查找法】对于已经**有序**的数组，查找数据时可以采用折半查找法。假定数组按照从小到大顺序排列，折半查找时，先测试中间元素，可以判断出欲查找目标在列表的前半部分还是后半部分。如果中间元素比目标大，则目标应在前半部分；如果中间元素比目标小，则目标应在后半部分；如果中间元素与目标相等，则找到目标。重复折半过程，直到找到目标或确定目标不在列表中。

设 3 个整型变量：top，bot，mid。top 存储查找范围的顶部位置下标，bot 存储查找范围的底部位置下标， $mid=(top+bot)/2$ 存储查找范围的中间位置下标。假设待查找的数据放在 x 中，数列放在 a 数组中，且数列中的数按从小到大的顺序排序，要进行三种判断：

- (1) 判断 x 是否等于 a[mid]，如果等于，则已找到，结束循环。
- (2) 判断 x 是否小于 a[mid]，如果小于，则 x 必定落在 bot 和 mid-1 范围之内，下一步的查找应该在此范围内进行。
- (3) 判断 x 是否大于 a[mid]，如果大于，则 x 必定落在 mid+1 和 top 的范围之内，下一步的查找应该在此范围内进行。

top 大于 bot 则无法找到，结束搜索。

```
#include<iostream>
```



```
#include<iomanip>
using namespace std;
const int N=10;
int main()
{
    int x, mid, bottom, top, i, a[N]={6,9,15,26,38,47,55,63,77,90};
    cout<<"x=";
    cin>>x;

    _____
    _____
    _____
    _____
    _____
    _____
    _____
    _____
    _____
    _____
    _____
    _____
    _____
    _____
    _____

    return 0;
}
运行结果: _____
_____
```

8.3.2 实习任务二

编写程序,实现对数组的冒泡排序。程序由 3 部分构成:通过键盘输入数据,按照冒泡法对数组按照由小到大顺序排序,最后输出排序后的数组内容。

【冒泡排序】列表被分成两个子列表:已排序和未排序的;初始时已排序部分为空。在未排序的子列表中,最小的元素通过冒泡的方法选出来并移动到已排序子列表中,这一过程称为一次扫描。每次扫描,未排序元素增加 1 个,已排序元素减少 1 个。冒泡时,进行两两比较,如果前者较大,则交换数据。对于 n 个数据,需要进行 n-1 次扫描。

The diagram shows the progression of bubble sort through six stages:

- Original list:** A horizontal array containing the numbers 23, 78, 45, 8, 32, and 56. A vertical bar is positioned between 78 and 45, with arrows pointing to both, indicating a comparison.
- After pass 1:** The numbers 23 and 78 are highlighted in light blue and labeled "Sorted" below them. The vertical bar is now between 45 and 8.
- After pass 2:** The numbers 23, 45, and 78 are highlighted in light blue and labeled "Sorted" below them. The vertical bar is now between 8 and 32.
- After pass 3:** The numbers 8, 23, 45, and 78 are highlighted in light blue and labeled "Sorted" below them. The vertical bar is now between 32 and 56.
- After pass 4:** The numbers 8, 23, 32, 45, and 78 are highlighted in light blue and labeled "Sorted" below them. The vertical bar is positioned at the end of the list, after 56.
- After pass 5:** All numbers (8, 23, 32, 45, 56, 78) are highlighted in light blue and labeled "Sorted" below them. The vertical bar remains at the end of the list.

[illegible]

8.3.4 实习任务四

【选择排序】列表被分成两个子列表：已排序和未排序的；初始时已排序部分为空。排序时，从未排序列表中找到最小元素，与第一个元素交换。经过选择和交换后，将未排序列表中的第一个元素划入排序列表中，这一过程称为一次扫描。每次扫描后，未排序列表减少一个元素，已排序列表增加一个元素。对于 n

个数据，需要进行 $n-1$ 次扫描。

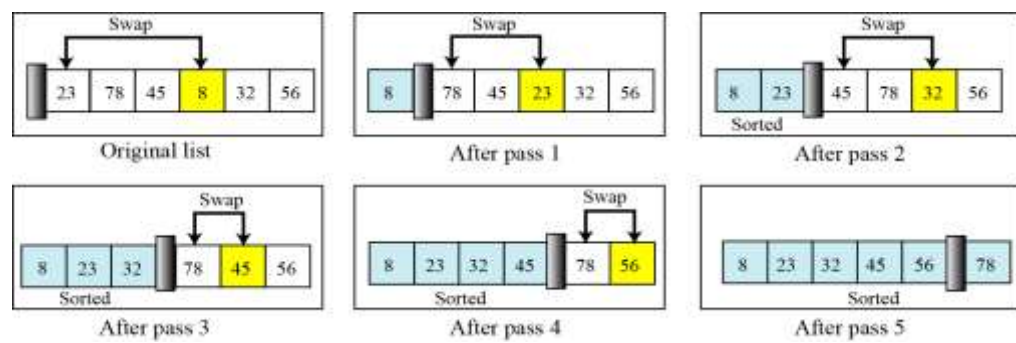


图 8.3 选择排序

程序代码：

```
#include<iostream>
#include<iomanip>
using namespace std;
void sort(int array[],int n);
int main()
{
    int a[10], i;
    cout<<"enter array:";
    for(i=0; i<10; i++)
        cin>>a[i];
    sort(a,10);
    cout<<"the sorted array:";
    for(i=0;i<10;i++)
        cout<<setw(5)<<a[i];
    cout<<endl;
    return 0;
}
void sort(int array[], int n)
{
    int i, j, k, t;
```

}

运行结果: _____

8.4 综合实例——学生成绩管理系统设计

根据本节实习所学的算法知识，完成学生成绩管理系统中数据处理功能：

1) 自定义排序函数，完成成绩由大到小的排序，排序后输出结果：

```
void Order(float stu[],int num);
```



图 8.4 排序成绩演示图

2) 自定义查找函数, 输入要查找的分数, 显示查找结果:

```
void Search(float stu[],int num);
```

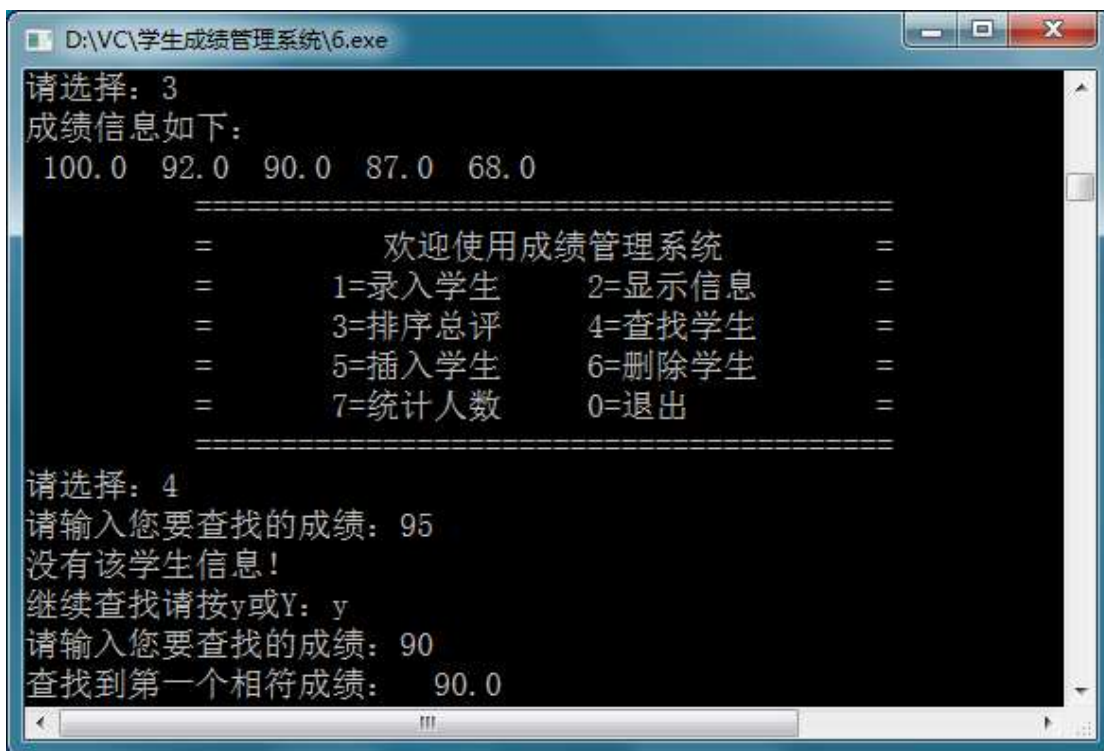


图 8.5 查找成绩演示图

3) 自定义删除函数, 输入要删除的成绩:

```
void Delete(float stu[],int &num);
```



图 8.6 删除成绩演示图

8.5 拓展知识

1) 简化的插入排序

本题要求编写程序，将一个给定的整数插到原本有序的整数序列中，使结果序列仍然有序。

输入格式：输入在第一行先给出非负整数 N (<10)；第二行给出 N 个从小到大排好顺序的整数；第三行给出一个整数 X 。

输出格式：在一行内输出将 X 插入后仍然从小到大有序的整数序列，每个数字后面有一个空格。

输入样例：

```
5
1 2 4 5 7
3
```

输出样例：

```
1 2 3 4 5 7
```

9 二维数组程序设计

9.1 实习目的

- 1) 掌握二维数组的定义和使用，二维数组在内存中的存储；
- 2) 熟练运用双重循环进行各种运算排序等操作；

9.2 前导知识

9.2.1 阅读程序一

阅读下面的程序，当从键盘输入 1 2 3 4 5 6 时，写出程序的输出结果。

```
#include<iostream>
using namespace std;
int main()
{
    int a[2][3];
    int i, j;
    for( i = 0; i < 2; i++ )
        for( j = 0; j < 3; j++ )
            cin >> a[i][j];
    for( i = 0; i < 2; i++ )
        for( j = 0; j < 3; j++ )
            cout << a[i][j] <<" ";
    return 0;
}
```

9.2.2 阅读程序二

二维数组的初始化分为两种，一种是顺序初始化，一种是按行初始化，阅读下段程序，写出程序的输出结果。

```
#include<iostream>
```



```

#include<iomanip>
using namespace std;
int main()
{
    int i,j,k,m;
    int array1[3][2]={4,2,5,6};
    int array2[3][2]={ {4,2},{5},{6}};
    cout <<"array1: " <<endl;
    for (i=0;i<3;i++)
    {
        for (j=0;j<2;j++)
            cout <<setw(2) <<array1[i][j];
        cout <<endl;
    }
    cout <<"array2:" <<endl;
    for ( k=0;k<3;k++)
    {
        for ( m=0;m<2;m++)
            cout <<setw(2) <<array2[k][m];
        cout <<endl;
    }
    return 0;
}

```

9.3 实习任务

9.3.1 实习任务一

求二维数组（3 行 3 列）的对角线元素的和。

10 12 13

14 15 16

17 18 19

This image shows a blank sheet of white paper with horizontal ruling lines. The lines are evenly spaced and run across the width of the page. There are no margins, text, or other markings on the paper.

9.3.2 实习任务二

将一个二维数组行和列元素互换，存到另一个二维数组中。例如：

$$\begin{bmatrix} 2 & 7 & 9 & 8 & 1 \\ 5 & 4 & 3 & 9 & 2 \\ 7 & 2 & 1 & 4 & 6 \end{bmatrix} \Rightarrow \begin{bmatrix} 2 & 5 & 7 \\ 7 & 4 & 2 \\ 9 & 3 & 1 \\ 8 & 9 & 4 \\ 1 & 2 & 6 \end{bmatrix}$$

最后请将转置前后的两个数组输出。

[illegible]

[illegible]

9.3.3 实习任务三

假设有一个 3×4 的矩阵，这个矩阵存入一个二维数组 `a` 中，数组中的数据如下：

```
int a[3][4]={ {5,12,23,56},{19,28,37,46},{-12,-34,6,8}};
```

要求编程序求出其中值最大的那个元素的值，以及其所在的行号和列号。

[illegible]

9.3.4 实习任务四

编写程序,从键盘输入杨辉三角的行数 n ,通过二维数组保存杨辉三角系数,并按如下形式进行输出杨辉三角。如 $n=5$,输出效果如下图所示:

```
1
1 1
1 2 1
1 3 3 1
1 4 6 4 1
```

【提示】杨辉三角每行的前后两个数字为 1,其他位置的数字为上一行相应位置和前一位置的和,即 $\text{array}[i][j]=\text{array}[i-1][j-1]+\text{array}[i-1][j]$ 。通过常量定义杨辉三角的行数,该常量也是保存杨辉系数的二维数组的行、列数。为了输出样式的美观,可以通过 `setw` 设置输出数字的宽度。

9.3.5 实习任务五*

在《射雕英雄传》中郭黄二人被裘千仞追到黑龙潭,躲进瑛姑的小屋。瑛姑出了一道题:数字 1~9 填到三行三列的表格中,要求每行、每列、及两条对角线

上的和都相等。这道题难倒了瑛姑十几年，被黄蓉一下子就答出来了。

4 9 2

3 5 7

8 1 6

编码实现 n 阶 (n 为正奇数) 幻方。

算法 (n 为奇数时):

- 1) 将 1 放在第一行中间一列;
- 2) 从 2 开始直到 $n \times n$ 止各数依次按下列规则存放: 按 45° 方向行走, 如向右上; 每一个数存放的行比前一个数的行数减 1, 列数加 1。
- 3) 如果行列范围超出矩阵范围, 则回绕, 例如 1 在第 1 行, 则 2 应放在最下一行, 列数同样加 1。
- 4) 如果按上面规则确定的位置上已有数, 或上一个数是第 1 行第 n 列时, 则把下一个数放在上一个数的下面。

- 2) 从 2 开始直到 $n \times n$ 止各数依次按下列规则存放: 按 45° 方向行走, 如向右上; 每一个数存放的行比前一个数的行数减 1, 列数加 1。

- 3) 如果行列范围超出矩阵范围, 则环绕, 例如 1 在第 1 行, 则 2 应放在最下一行, 列数同样加 1。

- 4) 如果按上面规则确定的位置上已有数, 或上一个数是第 1 行第 n 列时, 则把下一个数放在上一个数的下面。

This image shows a blank sheet of white paper with horizontal ruling lines. The lines are evenly spaced and run across the width of the page. There are no margins, text, or other markings on the paper.

9.4 拓展知识

- 1) 若矩阵 $A_{m \times n}$ 中存在某个元素 a_{ij} 满足： a_{ij} 是第 i 行中最大值且是第 j 列中的最小值，则称该元素为矩阵 A 的一个鞍点。试编写算法找出 A 中的所有鞍点及其坐标，没有鞍点的输出 no。

例如：

2 3 9 5

6 7 8 3

0 5 7 5

2 1 8 3

鞍点： $a[2][2]=7$

- 2) 试编写数独游戏程序。数独游戏就是把一个大正方形划成 3×3 的九个九宫格，每个九宫格又由 3 行 3 列共 9 个小方格构成，这样整个大正方形形成一个 9×9 的方格群。在这个大正方形内填满 1-9 的数字，要求大正方形每一行、每一列及每个九宫格内均必须包括 1 到 9 的每一个数字，既不能遗漏也不能重复。根据下面图左给出的例子试编写数独游戏程序输出下面右图结果。（超难，慎挑战!!）

	9		2		3		8	
1								6
	7		5		6		1	
2		9				7		3
		3				5		
8		5				1		4
	5		8		7		3	
9								5
	2		6		9		4	

5	9	6	2	1	3	4	8	7
1	3	2	4	7	8	9	5	6
4	7	8	5	9	6	3	1	2
2	4	9	1	8	5	7	6	3
7	1	3	9	6	4	5	2	8
8	6	5	7	3	2	1	9	4
6	5	1	8	4	7	2	3	9
9	8	4	3	2	1	6	7	5
3	2	7	6	5	9	8	4	1

10 字符数组和字符串

10.1 实习目的

- 1) 掌握字符串表示和使用，字符数组的定义和访问；
- 2) 掌握字符数组，以及字符串的读取和查找；
- 3) 了解 string 类字符串处理。

10.2 前导知识

10.2.1 阅读程序一

阅读下面的程序，根据输入的字符串，写出程序的输出结果。

注：小写字母比对应大写字母的 ASCII 码大 32，如'a'比'A'的 ASCII 码大 32。

```
#include <iostream>
using namespace std;
int main()
{
    char word[50];
    cout<<"Enter a word:";
    cin>>word;
    for(int i=0; word[i]!='\0'; i++)
        if(word[i]>='a' && word[i]<='z')
            word[i]-= 32;
    cout<<"Upper case: "<<word<<endl;
    return 0;
}
```

输入： Hello2016,C++100!

输出结果： _____

10.2.2 阅读程序二

阅读下面的程序，根据输入的字符串，写出程序的输出结果。

```
#include <iostream>
using namespace std;
int main()
{
    char word[50];
    cout<<"Enter a string:";
    cin>>word;
    int pos=0;
    for(int i=0; word[i]!='\0'; i++)
        if(word[i]<'0' || word[i]>'9' )
        {
            word[pos]=word[i];
            ++pos;
        }
    word[pos]='\0';
    cout<<"result: "<<word<<endl;
    return 0;
}
```

输入： Hello2016,C++100!

输出结果： _____

10.2.3 阅读程序三

阅读下面的程序，当输入： I love you,and you?写出程序的输出结果。

```
#include <iostream>
using namespace std;
int main()
{
    int i,flag=0,number=0;
    char str[100];
```



```

    gets(str);
    for(i=0;str[i]!='\0';i++)
    {
        if(str[i]==' ')
            flag=0;
        else
            if(flag==0)
            {
                flag=1;
                number++;
            }
    }
    cout<<number<<endl;
    return 0;
}

```

10.3 实习任务

10.3.1 实习任务一

编写函数 `strLength`，计算字符串的实际有效字符的个数，函数传入参数为字符串的起始地址，返回字符串中有效字符数量。函数原型如下：

```
int strLength(char str[]);
```

提示：系统存储字符串时，从首个字符开始，所有字符在内存中连续存储，最后以 '0' 作为字符串结束的标记。

10.3.2 实习任务二

10.3.3 实习任务三

编写函数 `insertString`，向已经有序的字符串中插入 1 个字符，使得插入后的字符仍然保持有序。传入参数为字符串起始地址和要插入的字符，假设存放字符串的数组有足够的空间以存放新插入的字符。函数原型如下：

```
void insertString (char str[], char c);
```

10.3.4 实习任务四

个位数统计：给定一个 k 位整数，请编写程序统计每种不同的数字出现的次数。例如：给定 $N = 100311$ ，则有 2 个 0，3 个 1，和 1 个 3。

输入格式：每个输入包含 1 个测试用例，即一个不超过 1000 位的正整数 N 。

输出格式：对 N 中每一种不同的个位数字，以 $D:M$ 的格式在一行中输出该位数字 D 及其在 N 中出现的次数 M 。要求按 D 的升序输出。

输入样例：100311

输出样例：

0:2

1:3

3:1

10.3.5 实习任务五*

D 进制的 A+B: 输入两个非负 10 进制整数 A 和 B, 输出 A+B 的 D ($1 < D \leq 10$)进制数。

输入格式: 输入在一行中依次给出 3 个整数 A、B 和 D。

输出格式: 输出 A+B 的 D 进制数。

输入样例: 123 456 8

输出样例: 1103

10.4 拓展知识

1) 单词转换

将 1 个英文单词转换为“Pig latin”形式的方法如下：如果单词以辅音字母开头，将第 1 个字符移到单词的最后，并在单词结尾添加 ay；如果单词以元音字母开头，则只在单词最后添加 ay。

利用 string 类型，编写程序，通过键盘读入若干单词，将每个单词转换为“Pig latin”形式，最后输出转换结果。示例如下：

输入：this is simple

输出：histay isay implesay

程序框架如下所示，补充遗漏的代码。

```
#include <iostream>
using namespace std;
#include <string>
bool isVowel(char letter);
int main()
{
    string word;
    string result;
    while(cin>>word)
    {
        //处理字符串转换
    }
    return 0;
}
//实现 isVowel 函数
```

【提示】对于字符串 word，word[0]可以得到字符串中的首字符，通过 word.substr(1)可以得到除第一个字符以外的子串；使用+运算符，可以将多个字符串或字符串与字符变量连接起来。

通过 while(cin>>word){}可以连续读入以空白字符（空格、回车等）分割的单词，直到遇到结束标志。在 Windows 环境下，通过输入 Ctrl+Z 组合键产生结束标志。

【string 类】

C++除了支持以'\0'结尾的字符数组表示字符串以外，还提供了更加方便的字符串类型：string，需要包含<string>头文件。

string 类封装了底层的字符串操作细节，使得用户无需关心字符数组的大小

和边界，以及最后的'\0'字符，string 类支持变长字符串，同时提供了多种操作字符串的方法，方便用户对字符串的各种处理。

要想使用标准 C++ 中 string 类，必须要包含 `#include <string>`
注意是 `<string>`，不是 `<string.h>`，带 .h 的是 C 语言中的头文件

字符串定义和初始化

```
string str1, str2; //创建名称为 str1 和 str2 的空串（内容为空）
string str3("China"); //创建初始内容为"China"，名字为 str3 的字符串
string str4="China"; //创建初始内容为"China"，名字为 str4 的字符串
string str5=str4; //创建内容和 str4 一样，名字为 str5 的字符串
```

常见的运算

```
getline(cin, str1, '\n') //从键盘输入一行字符（含空格），回车结束
getline(cin, str1, 'x') //从键盘输入一行字符（含空格），遇'x'或回车结束
str1=str2; //将字符串 str2 赋值给字符串 str1
str3=str1+str2; //连接 str1 和 str2，将连接后的结果赋值给 str3
str2[2]='a'; //将 str2 下标为 2 的字符修改为'a'
cin>>str1; //通过键盘读入字符串（只读入 1 个单词）
cout<<str1; //输出字符串 str1
```

常见的操作方法

```
str1.length(); //计算字符串 str1 的长度
str1.append(str2); //将 str2 内容追加追加到 str1 尾部
str1.empty(); //判断字符串 str1 是否为空串
str1.substr(2); //从下标 2 开始，截取余下的部分作为子串并返回
str1.substr(2,5); //从下标 2 开始，最多截取 5 个字符作为子串并返回
str1.insert(2, str2); //将 str2 插入到 str1 的下标 2 位置
str1.clear(); //将 str1 内容清空
```

【cin 流】运行以下程序：

```
#include<iostream>
using namespace std;
#include<cstring>
int main()
{
    int st=0;
```

```

while(st!=-1)
{
    cout<<"input to st:  ";
    cin>>st;
    cout<<"st= "<<st<<endl;
}
return 0;
}

```

分别输入：1；56；70；r;t;观察结果，发现当我们输入字符时产生了错误的输出。原因是 st 是 int 型，而输入时 char 型，出现类型匹配的错误。修改后：

```

#include<iostream>
using namespace std;
#include<cstring>
int main()
{
    int st=0;
    while(st!=-1)
    {
        cout<<"input to st:  ";
        cin>>st;
        while(cin.good()!=true)//判断标志位是否正确
        {
            cin.clear();//如果标志位不正确，修改为正确值
            cin.ignore();//清除缓冲区中的一个字符
            cout<<"input to st:  ";
            cin>>st;
        }
        cout<<"st= "<<st<<endl;
    }
    return 0;
}

```

11 复杂字符串程序设计

11.1 实习目的

- 1) 进一步掌握字符数组的双重循环操作；
- 2) 了解常用字符串函数。

11.2 前导知识

11.2.1 阅读程序一

阅读下面的程序，根据输入的字符串，写出程序的输出结果。

```
#include<iostream>
#include<cstring>
using namespace std;
int main()
{
    char arr[20];
    cin>>arr;
    int len=strlen(arr),i,j,k;
    int arr1[20] = {0};
    for(i=0;i<len;i++)
    {
        k=0;
        for(j=0;j<len;j++)
            if(arr[i]==arr[j])
                k++;
        arr1[i]=k;
    }
    for (i=0;i<len;i++)
        if(arr1[i]==1)
```



```

        {
            cout<<arr[i]<<endl;
            return 0;
        }
    cout<<"这个字符串组数中没有只出现一次的字符\n";
    return 0;
}

```

输入: eabefdffs

输出结果: _____

输入: eleloo

输出结果: _____

11.3 实习任务

11.3.1 实习任务一

字符串循环移位: 从键盘中输入一个正整数, 将给定的字符串顺时针循环移动输入的个数, 输出移动后的结果。例如: 字符串为"abcdefghijk", 循环次数: 4, 则移动后的结果为: "efghijkabcd"。

```

#include <iostream>
#include <cstring>
using namespace std;
int main()
{
    char s[] = "abcdefghijk";
    int len = strlen(s);
    int n;
    cout<<"请输入循环移动次数: ";
    cin>>n;

```

11.3.2 实习任务二

This image shows a blank sheet of white paper with horizontal ruling lines. The lines are evenly spaced and run across the width of the page. There are no margins, text, or other markings on the paper.

11.3.3 实习任务三

This image shows a blank sheet of white paper with horizontal ruling lines. The lines are evenly spaced and extend across the width of the page, providing a guide for writing. There are no margins or other markings on the paper.

11.3.4 实习任务四*

[illegible]

[illegible]

11.3.5 实习任务四

一个不超过 1000 位的非负整数 n ，将其各个位上的数字按升序排列，组成一个新数。注意，正整数的首位不为 0。

输入样例：265123

输出样例：122356

[illegible]

11.4 拓展知识

平时对字符串的操作的是很多的，了解下常用的字符串函数会使 c++ 编程变得很快捷。下面介绍几种常见的字符串操作函数。

1、字符串连接函数 strcat

其函数原型为：strcat(char[], const char[]);

strcat 是 string catenate(字符串连接)的缩写。该函数有两个字符数组的参数，函数的作用是：将第二个字符数组中的字符串连接到前面字符数组的字符串的后面。第二个字符数组被指定为 const，以保证该数组中的内容不会在函数调用期间修改。连接后的字符串放在第一个字符数组中，函数调用后得到的函数值，就是第一个字符数组的地址。例如：

```
char str1[30]= "People's Republic of";  
char str2[]="China";  
cout<<strcat(str1, str2); //调用 strcat 函数
```

输出：

People's Republic of China

连接前后的状况如下图所示。

tr1:	P	e	o	p	l	e	'	s	␣	R	e	p	u	b	l	i	c	␣	o	f	␣	\0	\0	\0	\0	\0	\0
tr2:	C	h	i	n	a	\0																					
tr3:	P	e	o	p	l	e	'	s	␣	R	e	p	u	b	l	i	c	␣	o	f	␣	C	h	i	n	a	\0

2、字符串复制函数 strcpy

其函数原型为：strcpy(char[], const char[]);

strcpy 是 string copy(字符串复制)的缩写。它的作用是将第二个字符数组中的字符串复制到第一个字符数组中去，将第一个字符数组中的相应字符覆盖。例如：

```
char str1[10], str2[]="China";
```

```
strcpy(str1, str2);
```

执行后，str2 中的 5 个字符"China"和"\0"(共 6 个字符)复制到数组 str1 中。

关于字符串复制函数 strcpy 的几点说明：

1) 在调用 strcpy 函数时，第一个参数必须是数组名(如 str1)，第二个参数可以是字符数组名，也可以是一个字符串常量。

2) 可以用 strcpy 函数将一个字符串中前若干个字符复制到字符数组中去。

3) 只能通过调用 `strcpy` 函数来实现将一个字符串赋给一个字符数组, 而不能赋值语句将一个字符串常量或字符数组直接赋给一个字符数组。

3、字符串比较函数 `strcmp`

其函数原型为: `strcmp(const char[], const char[]);`

`strcmp` 是 `string compare`(字符串比较)的缩写。作用是比较两个字符串。由于这两个字符数组只参加比较而不应改变其内容, 因此两个参数都加上 `const` 声明。以下写法是合法的:

```
strcmp(str1, str2);
```

```
strcmp("China", "Korea");
```

```
strcmp(str1, "Beijing");
```

比较的结果由函数值带回:

如果字符串 1=字符串 2, 函数值为 0。

如果字符串 1>字符串 2, 函数值为一正整数。

如果字符串 1<字符串 2, 函数值为一负整数。

字符串比较的规则与其他语言中的规则相同, 即对两个字符串自左至右逐个字符相比(按 `ASCII` 码值大小比较), 直到出现不同的字符或遇到 `\0` 为止。如全部字符相同, 则认为相等;若出现不相同的字符, 则以第一个不相同的字符的比较结果为准。

注意: 对两个字符串比较, 不能用以下形式:

```
if(str1>str2) cout<<"yes";
```

字符数组名 `str1` 和 `str2` 代表数组地址, 上面写法表示将两个数组地址进行比较, 而不是对数组中的字符串进行比较。对两个字符串比较应该用

```
if(strcmp(str1, str2)>0) cout<<"yes";
```

4、字符串长度函数 `strlen`

函数原型为: `strlen(const char[]);`

`strlen` 是 `string length`(字符串长度)的缩写。它是测试字符串长度的函数。其函数的值为字符串中的实际长度, 不包括 `\0` 在内。如:

```
char str[10]= "China";
```

```
cout<<strlen(str);
```

输出结果不是 10, 也不是 6, 而是 5。

以上是几种常用的字符串处理函数, 除此之外还有其他一些函数。

12 类的简单程序设计

12.1 实习目的

- 1) 掌握类的定义；
- 2) 掌握对象的定义和方法调用。

12.2 前导知识

12.2.1 阅读程序一

阅读下面的程序，写出程序的输出结果。

```
#include <iostream>
using namespace std;
class TestClass
{
public:
    TestClass(int a=0)
    {
        aa=a;
        cout<<aa<<" Constructed!\n";
    }
private:
    int aa;
};
int main()
{
    TestClass BB(5),a[5];
    return 0;
}
```

输出结果_____

12.3 实习任务

12.3.1 实习任务一

编写 Fraction 类，封装分数的基本功能，Fraction 的如下：

```
#include <iostream>
using namespace std;
class Fraction
{
public:
    Fraction();           //缺省构造函数设置分子为 0，分母为 1
    Fraction(int n);      //只有 1 个参数，设置分子为 n，分母为 1
    Fraction(int n, int d); //设置分子为 n，分母为 d
    void setValue(int n, int d); //设置分子和分母
    int getNum();          //获取分子值
    int getDen();          //获取分母值
    double getDoubleValue(); //获取分数对应的小数值
    void output();         //按分数形式显示分数
private:
    int num;    //分子
    int den;    //分母
};
```

给出所有成员函数的实现：

按照下面的主程序测试所写的分数类，检查结果是否正确。

```
int main()
{
    Fraction f1, f2(2), f3(3,4);
    f1.output();
    f2.output();
    f3.output();
    f1.setValue(6, 4);
    f1.output();
    cout<<" double value of f1: "<<f1.getDoubleValue( )<<endl;
    return 0;
}
```

运行结果: _____

12.3.2 实习任务二

实现 1 个简单的二维点类。构造函数可以用初始化列表实现对数据成员的初始化。在 Point 类中添加成员函数 getDistance，计算两点之间距离，类的定义如下，实现成员函数。

```
class Point
{
public:
    Point( double newX=0, double newY=0);
    void setValue(double newX, double newY);
    double getX( );
```

This image shows a blank sheet of white paper with horizontal ruling lines. The lines are evenly spaced and extend across the width of the page. There are no margins, text, or other markings on the paper.

测试的主程序如下：

```
int main()
{
    Point p1(3,4),p2(5,2);
    double dis1 = p1.getDistance(p2);    //成员函数版本计算
    cout<<"Distance: "<<dis1<<endl;
    double dis2 = getDistance(p1,p2);    //普通函数版本计算
    cout<<"Distance: "<<dis2<<endl;
    return 0;
}
```

12.4 综合实例——学生成绩管理系统设计

根据本节实习所学的类，完成学生成绩管理系统中全部数据处理功能：

（一）设计一个类 Student：

1. 数据成员：

- a) 学号 (id)
- b) 姓名 (name)：字符数组（说明：string 类型可以根据需要动态申请大小，但存取到文件的时候会有问题，所以本例用字符数组）
- c) 平时成绩 (regularGrade)
- d) 期末考试成绩 (finalExam)

2. 成员函数：

- a) 输入学号 (inputId)：本系统只管理一个班的学生成绩信息，所以这里要求控制输入的学号在 1-40 之间，如果超出这个范围，要求重新输入直到输入正确为止；ID 不能重复。
- b) 输入姓名 (inputName)
- c) 输入平时成绩 (inputRegularGrade)：要求成绩在 0-100 之间，如果超出这个范围，要求重新输入直到输入正确为止。
- d) 输入期末考试成绩 (inputFinalExam)：要求成绩在 0-100 之间，

如果超出这个范围，要求重新输入直到输入正确为止。

- e) 获得学号 (getId): 在后续排序、查找、删除等操作中均以学号为关键字。
- f) 输出成绩标题 (heading): 要求美观的输出标题。
- g) 输出成绩信息 (output): 要求与标题对应，并且所有成绩都保留一位小数。
- h) 计算总评成绩 (CalTotalMark): 计算总评成绩=平时成绩*50%+期末考试成绩*50%，并返回。

```
class Student
{
public:
    void inputId();
    void inputName();
    void inputRegularGrade();
    void inputFinalExam();
    int getId();
    static void heading();//输出成绩标题
    void output()const; //输出成绩信息
    float CalTotalMark();//计算总评成绩=平时*50%+期末*50%
private:
    int id;           //学号
    char name[20];     //姓名
    float regularGrade; //平时成绩
    float finalExam;   //期末考试成绩
};
```

- (二) 修改主函数中原来定义的实型数组 stu 的类型为 Student，用以存储学生的全部信息：

Student stu[MAX_STUDENT_NUM];

- (三) 修改自定义输入函数，保留根据用户的需要可以连续输入，直到达到上限的功能代码；删除检测成绩是否在 0-100 之间的代码，同时调用类的成员函数分别完成学号、姓名、平时成绩、期末考试成绩的输入，并控制学号不能重复。

void Input(**Student** stu[],int &num);

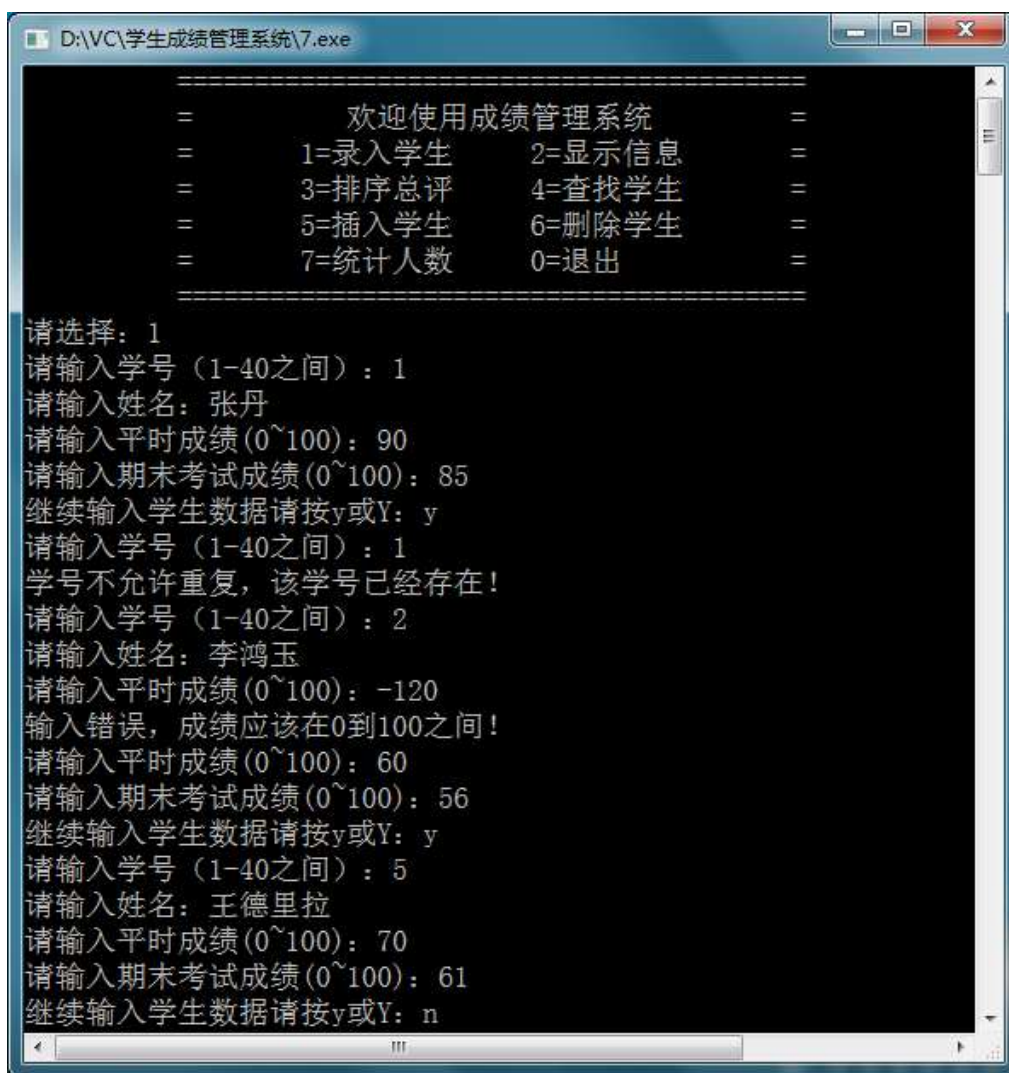


图 12.1 录入学生信息演示图

(四) 修改自定义输出函数，输出目前所有学生的信息。

void Output(*Student* stu[],int num);



图 12.2 输出学生信息演示图

(五) 修改自定义排序函数，完成总评成绩由大到小的排序，排序后输出结果：

```
void Order(Student stu[],int num);
```

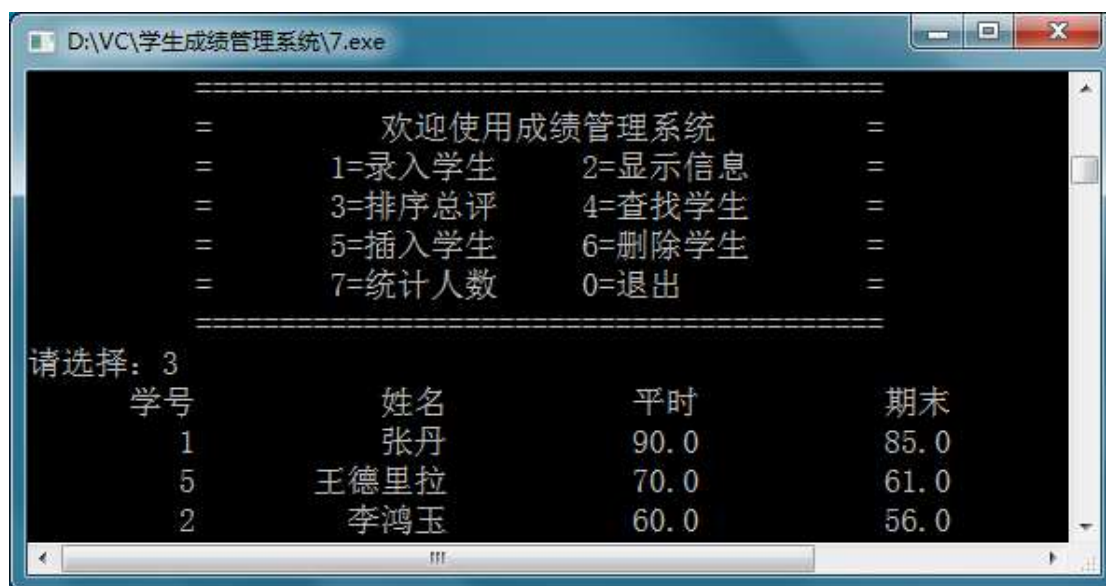


图 12.3 排序学生信息演示图

(六) 修改自定义查找函数，输入要查找的学号，显示查找结果。

```
void Search(Student stu[],int num);
```



图 12.4 依据学号查找学生信息演示图

(七) 修改自定义删除函数，输入要删除的学号。

```
void Delete(Student stu[],int &num);
```



图 12.5 删除学生信息演示图

(八) 修改自定义统计函数，统计总评成绩的各分数段的人数。

```
void Statis(Student stu[],int num);
```

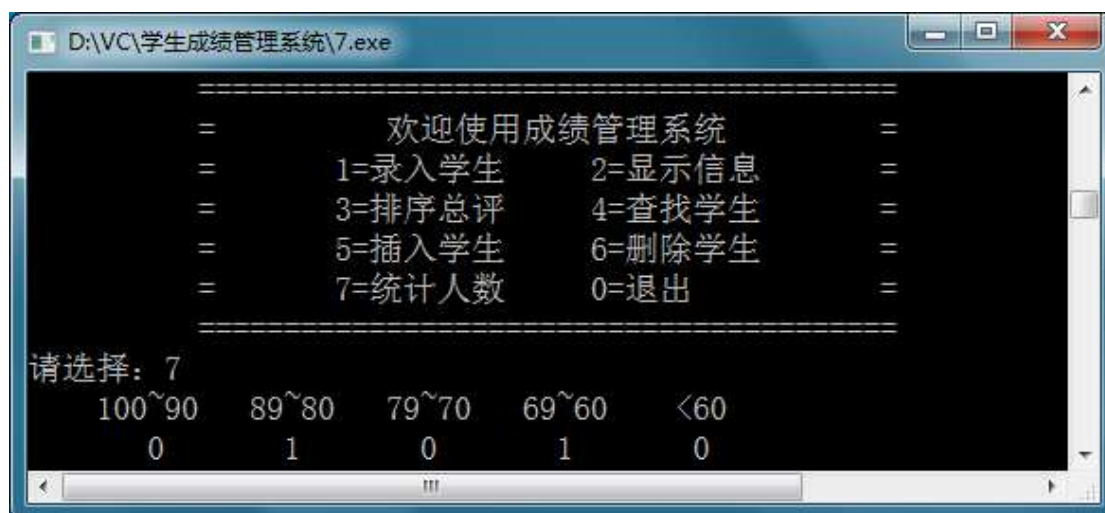


图 12.6 统计学生总评成绩演示图

12.5 拓展知识

1) 阅读下面的程序，写出程序的输出结果。

```

#include <iostream>

#include <string>

using namespace std;

class Person
{
private:
    string name;
    int age;
public:
    Person(string n,int a);
    void growup()
    {
        age++;
    }
};

Person::Person(string n, int a)
{
    name=n;
    age=a;
    cout<<"Hello,"<<name<<" is comming!"<<endl;
}

int main()
{
    Person p("zhang",1);
    for(int i=0;i<90;++i)
        p.growup();
    return 0;
}

```


13 文件

13.1 实习目的

- 1) 掌握文本文件读写的方法；
- 2) 掌握二进制文件的读写方法。

13.2 前导知识

13.2.1 阅读程序一

```
#include <iostream>
#include <iomanip>
#include <string>
using namespace std;
int main()
{
    double values[] = { 2.6,789,-13.6, };
    string names[] = { "Cat", "Animal", "Simth" };
    for (int i=0;i<3;i++)
        cout << setiosflags(ios_base::left)           //左对齐
              << setw(7) << names[i]
              << resetiosflags(ios_base::left)         //恢复默认右对齐
              << setw(9) << values[i] << endl;
    return 0;
}
```

13.3 实习任务

13.3.1 实习任务一

创建一个文本文件并写入信息。具体要求如下：

- 1、打开 d:\me.txt 文件用于写，若该文件不存在就创建它；

2、运行后打开文件 me.txt，其内容如下：

name: 张三

adress: 北京海淀区

13.3.2 实习任务二

读写文本文件的练习：

1、创建 d:\\try.txt 文件，若文件已存在则清空原内容；

2、打开该文件并写入以下内容：

运行结果：

2016

3.14

A

How are you

13.3.3 实习任务三

二进制文件拷贝文件：打开一个存在的二进制文件 d:\1.exe，将该文件中的内容拷贝到 d:\2.exe 中。

13.4 综合实例——学生成绩管理系统设计

在前面几个实习中，我们完成了一个班的成绩的增删改查统计等基本的数据处理功能，但是每次运行系统都需要重新输入新的数据，系统关闭后所有数据都丢失了，这样的系统作为测试可以，但是并没有实际的用处，为了让系统更实用，我们需要把处理过的数据长久的保存在外存储器上，以便以后实用。本次实习根据本节实习所学的文件知识，完成学生成绩管理系统中数据的存储功能。

- (1) 数据保存在外存储器：在用户即将退出系统之前把最新数据保存在外存储器上，即在用户选择 0 退出之前添加如下代码：

```
ofstream fileout("students.dat",ios::binary);
if(!fileout)
{
    cout<<"文件不能被打开！"<<endl;
    return 0;
}
fileout.write((char*)&currentN,sizeof(currentN));
for(int i=0;i<currentN;i++)
    fileout.write((char*)&(stu[i]),sizeof(stu[i]));
```

```
fileout.close();
```

```
return 0;
```

- (2) 从外存储器读入数据：在用户进入系统主界面之前把外存储器上的数据读入，即在主函数中定义完 `Student` 数组和 `currentN` 之后添加如下代码：

```
ifstream filein("students.dat",ios::binary);
if(filein)
{
    filein.read((char*)&currentN,sizeof(currentN));
    for(int i=0;i<currentN;i++)
        filein.read((char*)&(stu[i]),sizeof(stu[i]));
}
filein.close();
```

13.5 拓展知识

- 1) 显示文本文件的内容，使用 `get()` 一次读一个字符：打开一个已经存在的文本文件 `d:\\123.txt`，使用 `get()` 一次读一个字符，输出文件内容。

```
#include<fstream.h>
int main()
{
    ifstream fin("d://123.txt",ios::nocreate);
    if(!fin){
        cout<<"File open error!\n";
        return 0;
    }
    char c;
    while((c=fin.get())!=EOF)cout<<c;        //注意结束条件的判断
    fin.close();
    return 0;
}
```

- 2) 使用 `get(char *buf,int n,char delim='\n')` 一次读多个字符：打开一个已经存在的文本文件 `d://123.txt`，使用 `get(char *buf,int n,char delim='/n')` 一次读多个字符。

利用文本文件中不会有字符'\0'的特点进行读取该文件内容。

```
#include<fstream.h>

int main()
{
    ifstream fin("d://123.txt",ios::nocreate);
    if(!fin){
        cout<<"File open error!\n";
        return 0;
    }
    char c[80];
    while(fin.get(c,80,'\0')!=NULL)cout<<c; //注意结束条件的判断
    fin.close();
    return 0;
}
```

- 3) 使用 read(char *,int n)读文件：打开一个已经存在的文本文件 d:\\123.txt，使用 read(char *,int n)读文件。

```
#include<fstream.h>

int main()
{
    ifstream fin("d://123.txt",ios::nocreate);
    if(!fin){
        cout<<"File open error!\n";
        return 0;
    }
    char c[80];
    while(!fin.eof()) //判断文件是否读结束
    {
        fin.read(c,80);
        cout.write(c,fin.gcount());
    }
    fin.close();
    return 0;
}
```

```
}
```

4) C++按行读取文本文件: 假设有一个文本文件 `split.txt`, 该文件内容如下所示:

```
1 2 3
```

```
2 3 4
```

```
3 4 5
```

```
5 6 7
```

```
7 8 9
```

要求按照行读取数据, 并一个个的显示出来。

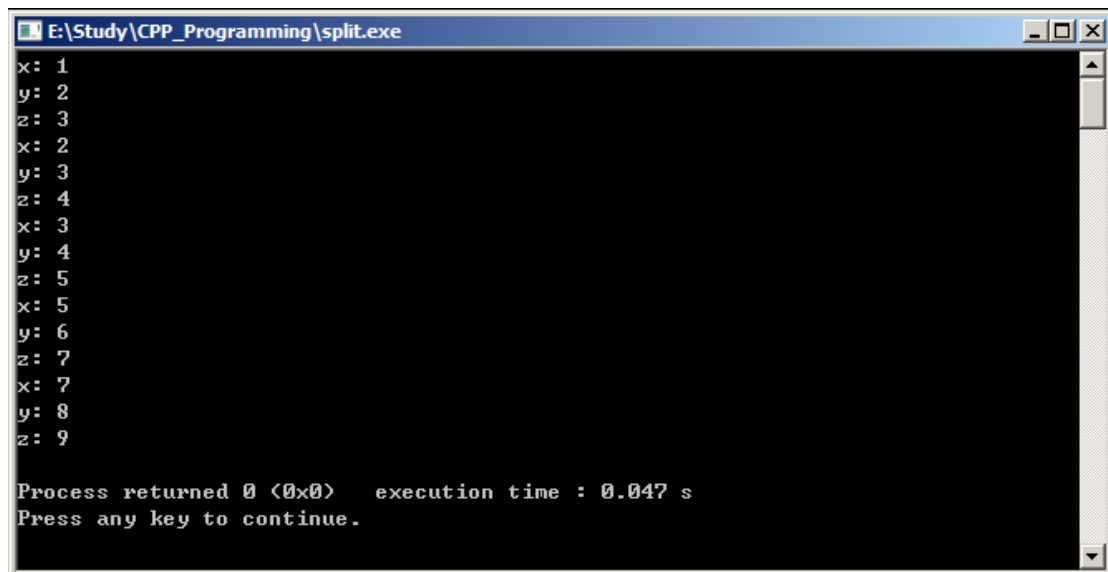
读取行的函数是 `getline()` 函数, 然后利用 `stringstream` 将每行文本自动按照空格分列, 并分别存放到对应的三个字符串变量中

```
#include <iostream>
#include <sstream>
#include <fstream>
#include <string>
int main(int args, char **argv)
{
    std::ifstream fin("split.txt", std::ios::in);
    char line[1024]={0};
    std::string x = "";
    std::string y = "";
    std::string z = "";
    while(fin.getline(line, sizeof(line)))
    {
        std::stringstream word(line);
        word >> x;
        word >> y;
        word >> z;
        std::cout << "x: " << x << std::endl;
        std::cout << "y: " << y << std::endl;
        std::cout << "z: " << z << std::endl;
    }
    fin.clear();
    fin.close();
}
```

```
return 0;
```

```
}
```

运行结果如下：



```
E:\Study\CPP_Programming\split.exe
x: 1
y: 2
z: 3
x: 2
y: 3
z: 4
x: 3
y: 4
z: 5
x: 5
y: 6
z: 7
x: 7
y: 8
z: 9

Process returned 0 (0x0)   execution time : 0.047 s
Press any key to continue.
```

14 指针与动态分配内存

14.1 实习目的

- 1) 掌握指针的定义和使用；
- 2) 掌握在堆中创建数据的方法；
- 3) 熟悉通过指针访问数组的方法；
- 4) 了解多文件的应用。

14.2 前导知识

14.2.1 阅读程序一

- 1) 阅读下面的程序，写出程序的输出结果。

```
#include <iostream>
using namespace std;
int main()
{
    double scores[4]={84, 76, 82, 91};
    double total=0;
    for(int i=0; i<4; i++)
        total+= scores[i];
    cout<<"Total score: "<<total<<endl;
    return 0;
}
```

输出结果_____

- 2) 将 1)的程序修改如下

```
#include <iostream>
using namespace std;
int main()
```



```

{
    int scoreNum = 4;
    double scores[scoreNum]={84, 76, 82, 91};
    double total=0;
    for(int i=0; i<scoreNum; i++)
        total+= scores[i];
    cout<<"Total score: "<<total<<endl;
    return 0;
}

```

请指出程序中的错误，并说明原因。

程序错误： _____

- 3) 将 1)的程序修改如下，写出程序的运行结果。

```

#include <iostream>
using namespace std;
int main()
{
    int scoreNum;
    cout<<"Enter the num of score: ";
    cin>>scoreNum;
    double total=0;
    double *pScores = new double[scoreNum];
    for( int i=0;i<scoreNum;++i )
    {
        cout<<"Enter score "<<i<<":";
        cin>>pScores[i];
        total+= pScores[i];
    }
    cout<<"Total score: "<<total<<endl;
    delete []pScores;
    return 0;
}

```

运行结果： _____

14.2.2 阅读程序二

阅读下面的程序，写出程序的输出结果。

```
#include <iostream>
using namespace std;
int main()
{
    int a=5, *pa=&a;
    cout<<"&a="<<&a<<",&a="<<a<<endl;
    cout<<"pa="<<pa<<",&pa="<<&pa<<",&pa="<<*pa<<endl;
    int b[]={2, 4, 6, 9, 3};
    int *pb=b;
    cout<<"b="<<b<<",&b[0]="<<&b[0]<<endl;
    cout<<"pb="<<pb<<",&pb="<<&pb<<",&pb="<<*pb<<endl;
    char s[]="Hello";
    cout<<"s="<<s<<",&s="<<*s<<endl;
    return 0;
}
```

运行结果： _____

14.2.3 阅读程序三

阅读下面的程序，写出程序运行结果。

```

#include <iostream>
using namespace std;
class TestClass
{
public:
    TestClass()
    {
        cout<<"Constructed!\n";
    }
    ~TestClass()
    {
        cout<<"Destructed!\n";
    }
};
int main()
{
    TestClass t1,t[3];
    TestClass *p;
    p=new TestClass;
    delete p;
    return 0;
}

```

运行结果： _____

【想一想】构造函数在什么情况下被调用？析构函数在什么情况下被调用？定义指针变量会调用构造函数吗？请说明程序中的每次输出都是由哪条语句触发引起的。

14.3 实习任务

14.3.1 实习任务一

编写函数 `getMax`，传入数组起始地址及数组元素个数，返回该数组的最大值，函数原型如下。

```
int getMax(int *pa, int size);
```

14.3.2 实习任务二

编写函数 `deleteChar`，用于删除字符串中指定的字符，将字符串中与指定字符相同的字符全部删除，函数原型如下：

```
void deleteChar(char *s, char c);
```

14.3.3 实习任务三

编写函数，实现 2 个字符串按照词典顺序的比较，函数原型如下：

```
int strcmp(char *s1,char *s2);
```

s1 指向字符串 1，s2 指向字符串 2，如果 s1 和 s2 相等，函数返回 0，如果 s1 和 s2 不等，返回 2 个字符串中从左向右第一个不相等字符的 ASCII 码差值，若果 $s1 > s2$ ，返回正值，若果 $s1 < s2$ ，返回负值。

14.4 综合实例——学生成绩管理系统设计

在前面实习中，我们用来存储学生成绩信息的是对象数组，数组的最大缺点是要求在定义的时候就确定大小，而每个班的人数是不一样的，而且可能存在转专业、退学等使班级人数发生变化的情况，所以我们在定义数组时为满足班级的各种人数情况只能定义的足够大，导致严重浪费。

在我们学习指针和动态内存分配后，同学们可以试着将数组改成根据需要动态申请内存空间，以达到程序灵活使用的目的。这里提供两种方案：

1、使用链表的形式

完全根据需要动态申请需要的内存，即每新增一名学生动态申请空间，每删除一名学生则回收该空间，几乎没有多余的浪费；但是链表的查询排序都比较麻烦，我们前面的程序要做比较大的改动，排序查找几乎要推翻重新做。

2、用数组的形式，但是可以动态扩大数组大小

数组创建时必须指定元素上限，之后不能动态变化，难以满足实际变化的情况，需要对其进行扩展。开始可以定义相对比较少数组空间，比如测试的时候可以用 3，实际应用的时候可以定义常见班型的 30。以后每超过一人都可以调用函数把数组大小增加 1，但是频繁的增加需要对数组元素移动，影响效率，所以在实际应用中一般达到数组上限可以考虑直接将原数组大小翻倍或者增加指定大小（这里测试时建议每次增加 2）。

（1） 修改班级最多人数全局常整型为变量：

```
const int MAX_STUDENT_NUM=3;//班级最多人数
```

（2） 增加每次扩容大小全局常整型：

```
const int reAllocMemoryNum=2;//每次扩容大小
```

（3） 将主函数中学生数组的定义修改为指针：

```
Student stu[MAX_STUDENT_NUM];
```

```
Student *stu;
```

注意：一般习惯将指针命名以 p 开头，本软件为了原有代码尽量少改动，指针保持与原数组同名。

（4） 修改文件读入代码：

```

if(filein)
{
    filein.read((char*)&currentN,sizeof(currentN));
    MAX_STUDENT_NUM=currentN;
    stu=new Student[MAX_STUDENT_NUM];
    for(int i=0;i<currentN;i++)
        filein.read((char*)&(stu[i]),sizeof(stu[i]));
}
else
    stu=new Student[MAX_STUDENT_NUM];
filein.close();

```

- (5) 在用户选择 0 退出系统之前的 return 0;语句之前添加一行回收内存代码语句:

```
delete[] stu;
```

- (6) 添加重新申请扩大内存函数:

```

Student* reAllocMemory(Student stu[],int num)
{
    Student *temp=stu;
    MAX_STUDENT_NUM+=reAllocMemoryNum;
    stu=new Student[MAX_STUDENT_NUM];
    int i;
    for(i=0;i<num;i++)
        stu[i]=temp[i];
    delete[]temp;
    return stu;
}

```

- (7) 修改输入数据函数:

```

Student* void Input(Student stu[],int& num)
{
    .....
    if(num>=MAX_STUDENT_NUM)
    {
        //cout<<"人数已达到"<<MAX_STUDENT_NUM<<"人上

```

```

限,无法再输入!"<<endl;

    stu=reAllocMemory(stu,num);

    //return;

}

.....

return stu;

}

```

(8) 修改主函数中用户选择 1 录入学生和 5 插入学生的调用函数:

```
stu=Input(stu,currentN);
```

14.5 拓展知识

1) 编写函数，计算 1 个数组的标准差，计算公式如下：

$$\delta = \sqrt{\frac{\sum_{i=1}^n (\bar{x} - x_i)^2}{n}}$$

其中 \bar{x} 是数组的均值。

函数传入参数为数组起始地址及数组元素个数，返回值为所计算的标准差。

函数原型如下：

```
double deviation( double *x, int n);
```

2) 计算平均学分积点

假定学期内有 5 门课程，各门课程的学分为 `credit[i]`，对应课程的学分积点为 `gradePoint[i]`，则学期平均学分积点的计算规则如下：

$$aveGP = \frac{\sum_{i=1}^5 credit[i] * gradePoint[i]}{\sum_{i=1}^5 credit[i]}$$

某课程的考试成绩为 `score`，则该课程的积点计算规则如下：

$$gradePoint = \begin{cases} 4 & score \geq 90 \\ 3 & score \geq 80 \\ 2 & score \geq 70 \\ 1 & score \geq 60 \\ 0 & score < 60 \end{cases}$$

编写程序，计算学期的平均学分积点，定义各课程的学分数组并初始化，定

义课程积点数组，通过键盘输入各科成绩并计算对应课程的积点，最后计算学期内所有课程的平均学分积点。

【提示】可以将根据成绩计算课程积点的功能定义为独立的函数，在主程序中调用，将来成绩转换积点的规则发生变化后，只需要修改该函数即可，而不用修改主程序。

程序代码框架已经给出，请补充遗漏的代码。

```
#include <iostream>
using namespace std;
int scoreToGradePoint(int score);
int main()
{
    const int CourseNum=5;
    int credit[CourseNum]={4, 4, 3, 2, 4};
    int scores[CourseNum];
    int gradePoint[CourseNum];
    double aveGP=0;
    int i;
    for(i=0;i<CourseNum;i++)
    {
        cout<<"Enter score "<<i<<": ";
        cin>>scores[i];
        gradePoint[i]=scoreToGradePoint(scores[i]);
    }
    // 补充计算并输出平均学分积点的功能
    return 0;
}
// 给出从成绩到课程积点的计算函数 scoreToGradePoint
```

【试一试】将 CourseNum 定义为普通变量，用户输入课程数量，动态分配内存，用户输入各门课程的成绩与学分。