

《程序设计基础》

上机

实验

指导

程序设计基础课程组

重庆理工大学

计算机科学与工程学院

2014

年

9

月

目录

1. 实验目的及要求 **................................................................................................................................................... 4**
   1. 实验目的 ......................................................................................................................................................... 4
   2. 实验具体要求 ................................................................................................................................................. 4
2. **C** 语言的运行环境 **................................................................................................................................................ 5**
   1. MICROSOFT VISUAL C++6.0 ............................................................................................................................ 5
      1. 启动 Microsoft Visual C++6.0 环境 ....................................................................................................... 5
      2. 建立工程 .................................................................................................................................................. 5
      3. 向已有工程中加入新文件 ...................................................................................................................... 7
      4. 编译、链接和运行程序 .......................................................................................................................... 8
   2. MICROSOFT VISUAL C++6.0 调式工具介绍.................................................................................................... 9
      1. 设置编译器选项 ..................................................................................................................................... 9
      2. 启动调试工具 Debugger .........................................................................................................................11
      3. 几种常用的 Debug 操作 .........................................................................................................................11
3. 上机实验内容 **..................................................................................................................................................... 13** 实验 1 C 程序编程环境认识和编制简单程序 .................................................................................................. 13 一、实验目的 ................................................................................................................................................. 13 二、必做题目 ................................................................................................................................................. 13 三、选做题目 ................................................................................................................................................. 14 四、实验结果 ................................................................................................................................................. 15 实验 2 数据类型、运算符与表达式 ................................................................................................................. 16 一、实验目的 ................................................................................................................................................. 16 二、必做题目 ................................................................................................................................................. 16 三、选做题目 ................................................................................................................................................. 17 四、实验结果 ................................................................................................................................................. 18 实验 3 顺序结构程序设计 ................................................................................................................................. 19 一、实验目的 ................................................................................................................................................. 19 二、必做题目 ................................................................................................................................................. 19 三、选做题目 ................................................................................................................................................. 19 四、实验结果 ................................................................................................................................................. 20 实验 4 分支结构程序设计 ................................................................................................................................. 21 一、实验目的 ................................................................................................................................................. 21 二、必做题目 ................................................................................................................................................. 21 三、选做题目 ................................................................................................................................................. 21 四、实验结果 ................................................................................................................................................. 22 实验 5 循环结构程序设计 ................................................................................................................................. 23 一、实验目的 ................................................................................................................................................. 23 二、必做题目 ................................................................................................................................................. 23 三、选做题目 ................................................................................................................................................. 24 四、实验结果 ................................................................................................................................................. 26 实验 6 数组 ......................................................................................................................................................... 27 一、实验目的 ................................................................................................................................................. 27 二、必做题目 ................................................................................................................................................. 27 三、选做题目 ................................................................................................................................................. 27 四、实验结果 ................................................................................................................................................. 29 实验 7 函数 ......................................................................................................................................................... 30 一、实验目的 ................................................................................................................................................. 30 二、必做题目 ................................................................................................................................................. 30 三、选做题目 1 .............................................................................................................................................. 30 四、选做题目 2 .............................................................................................................................................. 32 五、实验结果 ................................................................................................................................................. 34 实验 8 指针 ......................................................................................................................................................... 35 一、实验目的 ................................................................................................................................................. 35 二、必做题目 ................................................................................................................................................. 35 三、选做题目 1 .............................................................................................................................................. 36 四、选做题目 2 .............................................................................................................................................. 39 五、实验结果 ................................................................................................................................................. 42 实验 9 结构体、联合体 ..................................................................................................................................... 43 一、实验目的 ................................................................................................................................................. 43 二、必做题目 ................................................................................................................................................. 43 三、选做题目 ................................................................................................................................................. 44 四、实验结果 ................................................................................................................................................. 47 实验 10 文件 ....................................................................................................................................................... 48 一、实验目的 ................................................................................................................................................. 48 二、实验内容 ................................................................................................................................................. 48 三、实验结果 ................................................................................................................................................. 51 四、选做题目 ................................................................................................................................................. 49 实验 11 综合实验 ............................................................................................................................................... 52 一、实验目的 ................................................................................................................................................. 52 二、实验内容 ................................................................................................................................................. 52 三、实验结果 ................................................................................................................................................. 54 四、选做题目 ................................................................................................................................................. 52
4. 课程设计 **............................................................................................................................................................. 55**
   1. 课程设计的目的 ........................................................................................................................................... 55 4.2 课程设计流程 ............................................................................................................................................... 55
   2. 考核办法 ....................................................................................................................................................... 55
   3. 应提交的资料 ............................................................................................................................................... 56 附录 **A C** 语言常用调试技巧 **.............................................................................................................................. 57**

A.1 C 语言在编程中的特点............................................................................................................................. 57 A.2 C 语言编译时的常见错误 ......................................................................................................................... 57 A.2.1 源程序错误信息的分类 ........................................................................................................................... 57

A.2.2 C 程序的常见错误 .................................................................................................................................... 57 附录 **B** 课程设计报告文档格式 **........................................................................................................................... 69**

# **1** 实验目的及要求

## **1.1** 实验目的

《程序设计基础》是计算机专业的第一门专业基础课，课程开设的目的是帮助学生建立结构化程序设计的基本思想，通过课程的学习，使学生在尽可能短的时间里解决程序设计入门的问题，并为后继课程打下坚实的编程基础。

实践课是本门课程的重要环节。实验内容以验证性实验和设计性实验相结合为主，以综合性实验为辅。实验过程中要求使用 C 语言独立进行程序的编写和调试。通过实验，主要要达到下述目的：

1. 加深对结构化程序设计思想的理解。
2. 熟练掌握用结构化程序设计的思想解题的一般性方法，尽快解决程序设计的入门问题，以本门课程为基础，为后续课程做好准备。
3. 熟悉 VC++的编程环境，为今后学习 C++和 windows 程序设计做铺垫。
4. 熟悉 C 语言的语法要素。通过大量的验证性实验，巩固学生在理论课上所学习的知识点。
5. 初步建立算法的概念，通过设计性和综合性实验，培养算法分析的能力。
6. 熟悉高级语言程序设计中程序的编写、调试、连接、运行的全过程。掌握程序调试的基本步骤，通过实验积累程序调试的经验。
7. 学习如何选择测试数据、如何根据运行结果及错误提示发现并排除程序中的错误，建立初步的软件测试观念。

# **1.2** 实验具体要求

学生在实验前必须作好上机的各项准备，按教师的要求进行上机实验。具体要求如下：

1. 根据实验指导书提前作好上机预习。必须携带准备好的源程序上机。程序可以是由教师布置安排的题目和自选题目。

编写的程序应书写整齐，并经人工检查无误，以提高上机效率。对程序中自己有疑问的地方，应作出记号，以便在上机时给予注意。

1. 必须携带教材及相关参考资料上机。 3、调试程序的过程原则上应独立完成。

独立上机是培养学生独立思维能力的一个方面，上机过程中出现的问题，一般应自己独立处理，不要轻易举手问教师。对常见的“出错信息”应尽快熟悉其含义，并在总结经验的基础上迅速排除常见的语法错误。

1. 在程序调试通过后，应作好程序清单和运行结果的记录，实验结束后，形成实验报告。实验报告应包括如下内容：

⑴实验时间⑵实验内容⑶程序清单⑷运行结果及分析结论

1. 上机过程中严格遵守实验室纪律，养成良好的上机习惯。

**2 C** 语言的运行环境

目前，可以运行和调试 C 语言的环境有很多，比如传统的 Turbo C 环境，Boland C 环境，以及后来开发的 Microsoft VisualC++等等。Microsoft VisualC++6.0 提供了相对较好的编辑、编译和调试环境，所以，本书采用该环境进行 C 语言程序的编译、调试和运行环境。

## **2.1 Microsoft Visual C++6.0**

**2.1.1** 启动 **Microsoft Visual C++6.0** 环境

启动 Microsoft VisualC++6.0(为叙述方便，后面均简称 VC)环境的常用方法有 3 种；双击桌面图标；从桌面左下角的“开始”菜单进入；从桌面左下角的“运行”功能中进入。

1.通过双击中桌面图标直接启动 VC 环境

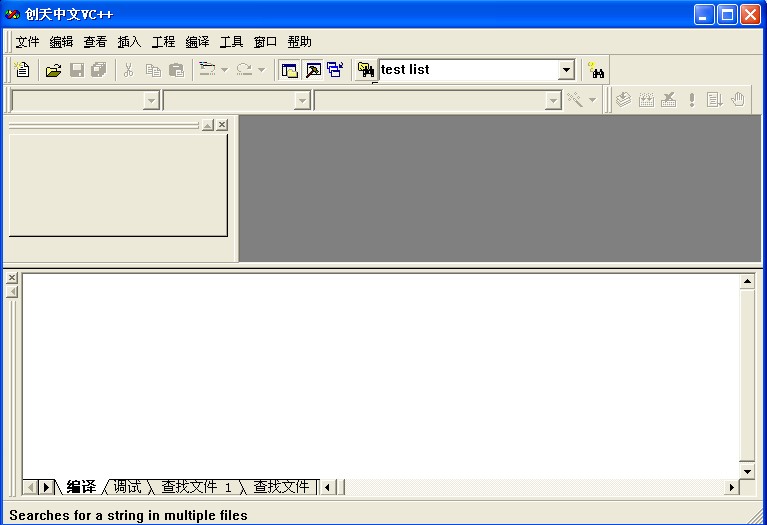
在桌面上找到 VC 的图标，使用鼠标左键（为叙述方便，后面若没有特殊说明均指左键）进行双击，即可打开 VC 环境。

2.从桌面左下角的“开始”菜单进入 VC 环境

1. 用鼠标单击左下角的“开始”菜单；
2. 将鼠标上移至“程序（P）”处；
3. 然后将鼠标再水平右移，在下一级菜单中移至 Microsoft Visual Studio6.0 处；
4. 再将鼠标右移至下一级菜单上，并将鼠标上下移动到 Microsoft VisualC++6.0 处，单击鼠标会出现如图 2.1 的情况，这就是编程时要用到的 VC 环境。

3、从桌面左下角的“运行（R）”功能中进入 VC 环境

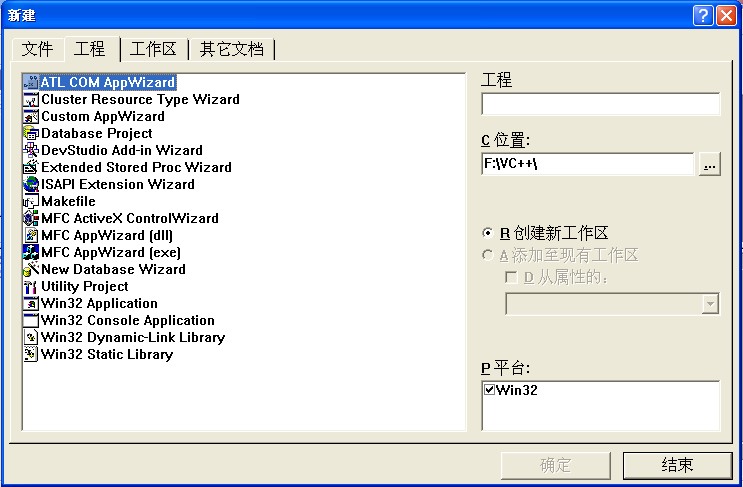
1. 用鼠标单击桌面左下角的“开始”菜单；
2. 将鼠标上移到“运行（R）”处，单击鼠标会出现一个“运行”对话框；
3. 在弹出的对话框中键入 msdev 字样，然后单击“确定”即会出现如图 2.1 所示的 VC 键编程环境。

 图 2.1 VC 环境窗口

**2.1.2** 建立工程

建立一个 VC 工程（Project）的步骤如下：

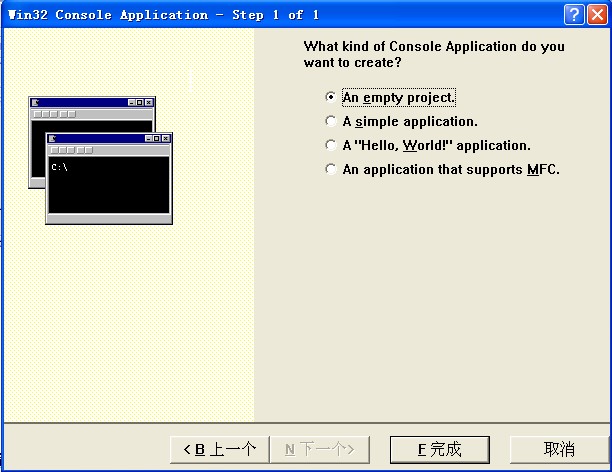
1. 在如图 2.1 所示的 VC 环境窗口中单击 File 菜单项，然后选择 New，就会出现如图 2.2 所示的对话框。

 图 2.2 新建工程的对话框

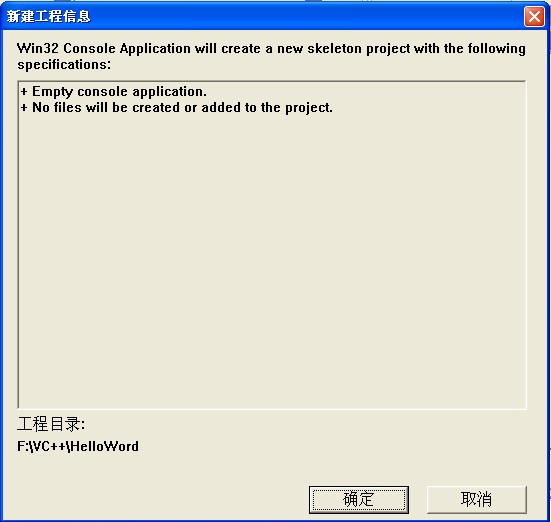
1. 此时编程者需要在工程类型列表中选择合适的工程类型，“程序设计基础”课程主要涉及到的是算法程序，所以一般选择 Win32 Console Application 类型的工程就可以了。
2. 在选择了合适的工程类型后，填写工程的名称，具体方法是在如图 2.2 所示对话框中的工程（或

Project name） 编辑框中输入工程名称，HelloWorld，编程者可以根据自已的需要和喜好输入其他名称。

1. 然后再在位置（或 Location）编辑框中的选择工程保存目录，并单击 OK 按钮，就会弹出如图 2.3 所示的对话框。

 图 2.3 选择 Win32 Console Application 后的模板

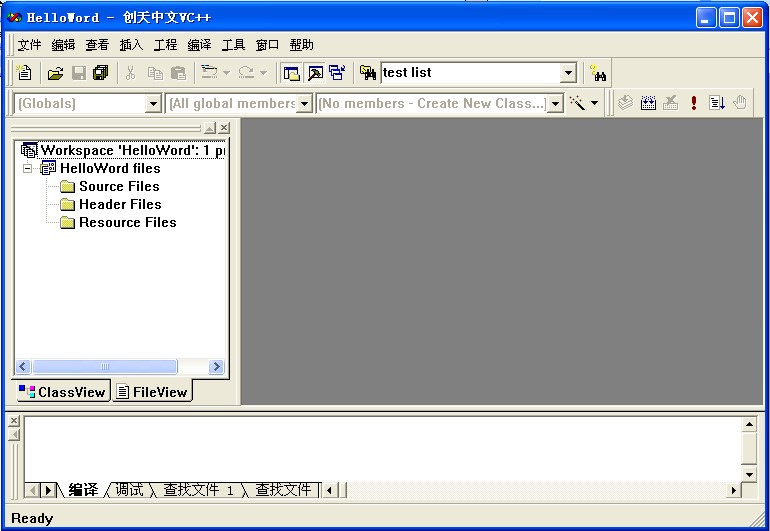
1. 在如图 2.3 所示的对话框中，列出了 4 种预设的工程模板，选择第 1 项 An empty project，然后单击完成（或 Finish）按钮，会弹出如图 2.4 所示的新工程报告对话框。

 图 2.4 新工程报告对话框

1. 从图 2.4 所示的对话框中可以看到新建工程的一些简要信息，确定信息准确无误后，单击确定（或

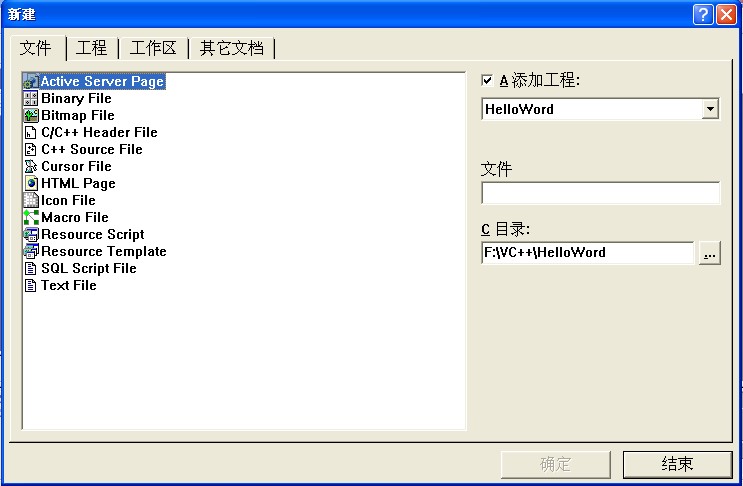
OK）按钮，这样就成功地新建了一个 VC 工程。

1. 现在观察一下 VC 的（New）Projects 做了什么工作，图 2.5 显示了刚刚新建的 HelloWord 工程在 VC 环境中的情况，窗口左边部分是 Workspace，它显示了有关工程的信息，包括类信息、资源信息、源文件信息等。单击下部的 FileView，在 Workspace 中可以看到 3 个目录：一般·cpp 文件放在 Source File 中；头文件·h 或·hpp 放在 Header File 中；资源文件放在 Resource Files 中；还可以建立自己的目录。

 图 2.5 新建 HelloWorld 工程的展开全貌

**2.1.3** 向已有工程中加入新文件

1. 如果希望在上面已建的工程 HelloWorld 之中新建一个文件，可以单击文件（或 File）菜单项，然后单击新建（或 New）子菜单项，就会弹出如图 2.6 所示的对话框。

 图 2.6 新建文件对话框

1. 在如图 2.6 所示的对话框中选择文件类型。例如：想要建立一个.c 或.cpp 文件，那么可以选择 C++

Source File 项目，如果想要建立一个.h 文件，那么可以选择 C/C++ Header File 项目。

1. 然后需要输入文件名称，这里输入 SubCall，再选择文件保存的路径，最后确认选中添加工程（Add to project）复选框，并单击确定（OK）按钮，这样就成功的新建了一个名为 SubCall.c 的文件，并已加入到了前面已创建的 HelloWorld 工程之中。

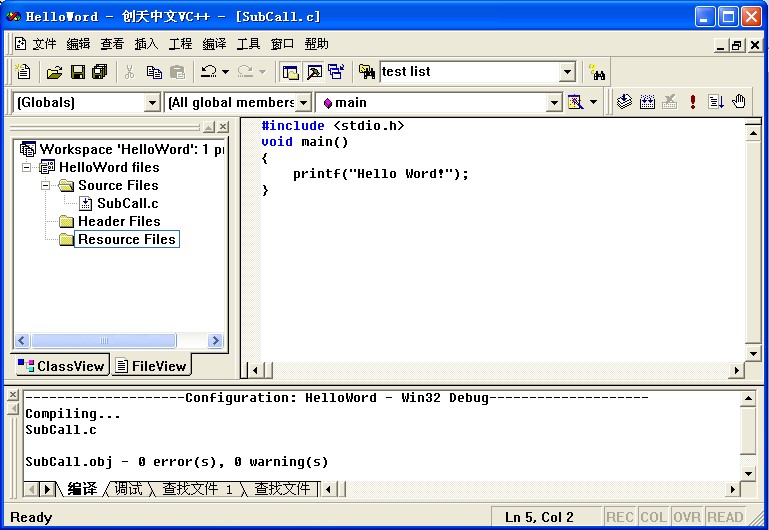
**2.1.4** 编译、链接和运行程序

按 Ctrl+F7（Compile）键可以进行程序的编译，按 F7(Build)键可以进行程序的编译、链接。

编译、链接后，如果出现错误（Error）、警告（warning）等信息，按 F4 键可以直接跳转到错误、警告信息在源文件中所对应的行，便于编程者进行查看和修改，若继续按 F4 键，可依次跳转到相邻的下一个警告或错误之处，比较方便快捷。

按 Ctrl+F5（Execute）(或者直接按快捷按钮“”)键，可以直接运行程序（忽略断点，后面在 2.2 中会详细解释有关断点的内容）。按 F5（GO）键可以在 Debug 模式下运行程序，会在一些预设的断点处停止运行。

图 2.7 重点显示了最常用的快捷键按钮。



图

2.7

编译、链接和运行程序

编译

链接

停止链接

直接运行

调试

插入、删

除断点

## **2.2 Microsoft Visual C++6.0** 调式工具介绍

**2.2.1** 设置编译器选项

VC 环境提供了编译器优化选项定制功能，系统的默认（缺省）设置已满足对程序性能（如运行速度和内存占用等方面）的一般需求，但是对于那些对程序性要求非常 高的程序来说，有必要通过手工设置和调整一个工程的统译器优化选项，来适应在特定运行环境中达到最高的性能，例如：对于特定的 CPU，选择不同的编优化方式可以达到比默认设置更好的性能，下面讲解如何调整几个常用的编译选项。

1. 首先在 VC 环境窗口中选择工程（Project）菜单项中的设置（Settings）子菜单，会弹出如图 2.8 所示的对话框，由于要修改的编译选项，所以选择 C/C++标签。

 图 2.8 工程设置（project setting）对话框

1. 当需要选择一个一般优化目标的方法时，可以在分类（Category）下拉列表框中选定 General。这个优化选项针对用户对应用程序的整体需求进行整体优化，用户可以通过调整编译选项使得程序的运行速度达到最优，或者是程序体积（文件的大小）实现最小，但如果用户还处于调试程序阶段。则应关闭所有的优化选项。
2. 可以看到在如图 2.9 所示的优化（Optimizations）下拉列表框中，共有 5 项选择：Default， Disable[Debug]， Maximize Speed， Minimize Size 和 Customize。其中 Disable[Debug]将关闭所有的优化选项；Default 是提供一些快速简单的代码优化：Maximize Speed 和 Minimize Size 分别以最快运行速度和最小程序休积为目标进行优化；Customize 优化选项提供了手工选项，适合高级用户的使用。

 图 2.9 优化（Optimizations）下拉列表框

1. 如果希望选择优化处理器和工程的默认调用规范，可以在分类（Category）下列表框中选定 Code Generation，如图 2.10 所示，这部份的优化选项是针对具体运行环境进行设置：例如可以选择在某种类型的 CPU 上的性能最优，另外，还能修改一些默认的调用规范编译规则。这些内容可以作为初学者的一般了解内容。

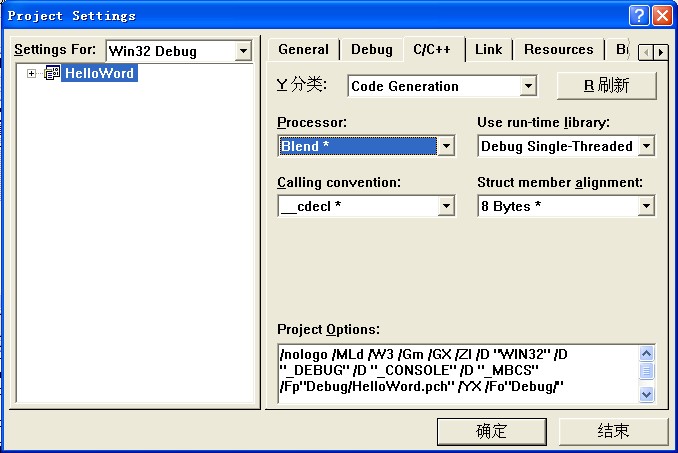


图 2.10 Code Generation 界面

1. 在 Processor 下拉列表框中可以选择 80386、80486 或 Pentium 等处理器代码生成。默认设置 Blend\* 表示了一种折中的处理，这种处理侧重于 Pentium 组但增加了对前代处理器的选择优化。
2. 在 Calling convention 下拉列表框中可以选择 3 种调用规范：\_cdecl\*，\_fastcall 和\_stdcall。调用规范决定了自左至右或自右至左的参数传递顺序、有关清除堆栈以及函数名字解释等。若想了解这 3 种调用规范的区别，可以通过查阅 VC 的联机帮助 MSDN 得到进一步的解释。
3. 在 Struct member alingment 下拉列表框中可以设置结构或联合成员的对齐边界，而 Use run-time library 下拉列表框的功能我们这里不做讨论。

**2.2.2** 启动调试工具 **Debugger**

要启动 Debugger，道先要确认工程类型 是 Win32 Debug。方法是选择工程（Project）菜单项中的设置（Settings）子菜单，在弹出如图 2.8 所示的对话框中，选择 Settings For 下拉列表框中的 Win32 Debug 选项。

确定工程类型为 win32 Debug 以后，选择 VC 菜单编译（Build）下的子菜单开始调试（Start Debug），再选择二级子菜单去（GO）；或者进接按 F5 键，就可以启动 Debugger 了。

**2.2.3** 几种常用的 **Debug** 操作

表 2.1 显示了最常用的几种 Debug 操作。

|  |  |  |
| --- | --- | --- |
| Debug 操作 | 动作说明 | 快捷键 |
| GO | 启动 Debugger，并执行程序，直到遇到一个断点或程序结束，或直到应用程序暂停等待用户输入 | F5 |
| Step Into | 启动 Debugger，并逐行单步执行源文件，当所跟踪的语句包含一个函数或一个方法调用时，Step Into 进入所调用的子程序中 | F11 |
| Step Out | 结束所调用子程序中的调试，跳出该子程序，与 Step Into 对应 | Shift+F  11 |
| Step Over | 启动 Debugger，并逐行单步执行源文件，当所跟踪的语句包含一个函数或一个方法调用时，Step Over 不进入所调用的子程序中，而是直接跳过。 | F10 |
| Run to Cursor | 启动 Debugger，并执行到光标所在的行 | Ctrl+F1  0 |
| Insert/Remove  Breakpoint | 在光标处插入/删除断点 | F9 |

为了 试调方便，也可以打开 Debug 工具栏，打开的方法是在 VC 环境窗口上部的菜单的空白处单击鼠标右键，在弹出的菜单中选择（）“调试”（“”）即可，图 2.11 显示了 Debug 工具栏的图示。

 图 2.11 Debug 工具栏

Debug 工具栏中所包括的其它常用功能如表 2.2 所示。

|  |  |  |
| --- | --- | --- |
| Debug 操作 | 动作说明 | 快捷键 |
| Restart | 从开始处调试程序，而不从当前所跟踪的位置开始调试 | Ctrl+shift+F5 |
| Stop Debugging | 结束调试，直接退出 Debugger | Shift+F5 |
| Quick Watch | 显示 Quick Watch 窗口，在该窗口中可以计算表达式的值 | Shift+F9 |
| Watch | 显示 Watch 窗口，该窗口包含关于当前和前面的语句中所使用的变量和返回值。当前函数的局部变量在 Local 标签中 | Alt+3 |
| Variables | 显示 Variables 窗口，该窗口包含关于当前和前面的语句中所使用的变量和返回值。当前函数的局部变量在 Local | Alt+4 |

|  |  |  |
| --- | --- | --- |
|  | 标签中 |  |
| Registers | 打开 Registers 窗口，显示微处理器的一般用途寄存器和  CPU 状态寄存器 | Alt+5 |
| Memory | 打开 Memory 窗口，显示该应用程序的当前内存内容 | Alt+6 |
| Call Stack | 打开 Call Stack， 显示该应用程序的当前内存内容 | Alt+7 |
| Disassembly | 打开一个包含汇编语言代码的窗口 | Alt+8 |

**3** 上机实验内容

## 实验 **1 C** 程序编程环境认识和编制简单程序

1. 实验目的
   1. 了解高级语言程序编写、编译、调试、和运行的基本过程
   2. 熟悉和掌握 Microsoft Visual C++6.0（以后均简称 VC） IDE 环境的操作和使用，进行简单的 C 语言程序的录入、调试、编译和运行。
   3. 认识常见的错误信息。
   4. 注意培养良好的编程习惯。
2. 必做题目
   1. 打开计算机，启动 VC++6.0，熟悉 VC++6.0 IDE 环境下各菜单命令的使用。
   2. 上机练习：
   3. 输入如下程序，并进行编译和运行。同时应了解所用的系统是用什么命令进行编译和连接运行的，编译和连接后得到的目标程序的后缀是什么形式，命令程序的后缀是什么形式。写出程序功能和运行结果。

#include <stdio.h> void main()

{

printf("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n")； printf("\n")；

printf(" Very Good! \n")； printf("\n")；

printf("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n")；

}

* 1. 输入如下程序，编译并运行。写出程序功能和运行结果。

#include <stdio.h> void main()

{

int a，b，c，d，sum； a=10； b=20； c=30； sum=a+b\*c； a=c； d=a+c；

printf("sum=%d， d=%d"，sum，d)；

}

* 1. 如下程序的功能是计算梯形面积 s，设梯形上底 a=5，下底 b=10，高 h=6。请完成程序并调试。 void min()

{

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

a=5； b=10； h=6；

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ printf(“a=%d，b=%d，h=%d，s=%f\n”，a，b，h，s)；

}

* 1. 编写程序：任意输入三个数，计算并输出它们的平均值以及三个数的乘积。
  2. 编写程序：已知匀加速运动的初速度为 10m/s，加速度为 2m/s2，求 20s 以后的速度、20s 内走过的路程及平均速度，写出程序。

1. 选做题目
   1. 编写程序：任意输入三个数，计算并输出它们的和以及这三个数的平均值。
   2. 输入一个整数 x，计算并输出 y=(x-10)3，编写程序，实现上述功能。
   3. 编一个程序，将任意一个两位正整数平方后，取其百位数和十位数，构成一个新的两位整数。
   4. 输入一个实数 x ，计算 y=x2，输出 x，y。
   5. 一个学生五门课的成绩为：70、89、65、100、78。编一个程序，计算并打印该学生的平均成绩。
   6. 找出程序中的错误并修改

void main()

{

int i1； i1=1000；

double f1； f1=123.4；

}

对于该程序，系统编译的时候会提示错误信息。如果程序修改为如下，再编译，错误信息消失，注

意体会其间的差别，看看上面的程序错在哪里。

Void main()

{

int i1； double f1； i1=1000；

f1=123.4；

}

（提示：C 语言规定，所有用到的变量都必须先定义，后使用；说明部分放在执行部分的前面。）

7、运行程序，给出其运行结果

#include <stdio.h> void main()

{

int i2；

int i1=1000；

{

int i1=100；

printf("i1=%d"，i1)；

}

i2=i1；

printf("i2=%d\n"，i2)；

}

1. 找出程序中的错误并修改

#include <stdio.h> void main()

{

int x，y，z；

scanf(“%d”，&z)； x=0； z=x+y

printf(“z=%d\n”，z)

}

1. 找出程序中的错误并修改

#include <stdio.h>

#define x 3.5

#define a 5.5

{ float x，y；

y=x+z； printf(“a=%f，x=%f，y=%f\n”，a，x，y)；

}

1. 运行程序，写出程序的运行结果

#include <stdio.h> void main()

{

int a，b，c，n，sum；

n=126； a=n/100； b=(n-a\*100)/10； c=n-100-b\*10； sum=a+b+c；

printf(“sum=%d”，sum)

}

四、实验结果

写实验报告。内容包括：

1、必做习题的运行结果，源程序； 2、选做习题的运行结果，源程序。 3、程序调试中出现的错误提示。（英文、中文对照） 4、若有没通过的程序，分析原因。

## 实验 **2** 数据类型、运算符与表达式

1. 实验目的
   1. 掌握 C 语言数据类型，熟悉如何定义整型、实型、字符型变量的方法，以及如何对其进行初始化和赋值操作。
   2. 熟悉并掌握常用的转义字符和常用的格式转换符。
   3. 学会使用 C 语言的有关算术运算符，以及包含这些运算符的表达式，特别是自加（++）和自减（--）运算符的使用。
   4. 注意各运算符的优先级和结合性。
   5. 注意不同数据类型的转换方法，以及转换的优先顺序。
2. 必做题目
   1. 分析以下程序结果，输入并运行该程序，进行验证。

#include <stdio.h> void main()

{

char c1，c2；

c1=97；c2=98；

printf("c1=%c c2=%c\n"，c1，c2)； printf("c1=%d c2=%d\n"，c1，c2)； c1=c1-32；c2=c2-32； printf("c1=%c c2=%c\n"，c1，c2)；

printf("c1=%d c2=%d\n"，c1，c2)；

}

2、分析以下程序结果，输入并运行该程序，进行验证。

#include <stdio.h> void main()

{

int i，m，n；

i=5；

printf("before i++ i=%d\n"，i)； m=i++；

printf("after i++ i=%d\n"，i)；

n=++i；

printf("after ++i i=%d m=%d n=%d\n"，i，m，n)；

}

3、计算以下表达式的值，并编写程序，验证你的计算结果。

1. (float)(a+b)/2+(int)x%(int)y

设 a=2，b=3，x=3.5，y=2.5

1. 假定 W， x， y， z， m 均为 int 型变量，有如下程序段：

W=1；x=2；y=3；z=4；

m=(w<x)?w:x；m=(m<y)?m:y；m=(m<z)?m:z；

则该程序运行后，m 的值是多少？

⑶设 a， b， c， d， m， n 均为 int 型变量，且 a=5，b=6，c=7，d=8，m=2，n=2，则逻辑表达式(m=a>b)&&(n=c>d)运算后，n 的值为多少？

4、以下程序的功能是：将输入的四位数 abcd 以反序输出 dcba，如输入 1234，输出 4321。完成程序，并进行编译和运行。 #include <stdio.h> void main()

{

int n，d1，d2，d3，d4； scanf(“%d”，&n)； d1=n%10； d2=(n%100)/10；

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_； \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_； printf(“n:%d change to:%d%d%d%d\n”，n，d1，d2，d3，d4)；

}

1. 选做题目
   1. 分析并写出以下程序的运行结果。

#include <stdio.h> void main()

{

int a=3；

printf("%d\n"， a+(a-=a\*a))；

}

* 1. 分析并设计程序验证结果:设 int x=1， y=1， 则表达式(!x||y)的值是多少？
  2. 写出下面赋值表达式运算后 a 的值，并编写程序上机验证。设原来 a=12， n=5。a 和 n 都定义

为整型变量。

⑴a+=a ⑵a-=2 ⑶a\*=2+3

⑷a%=(n%=2) (5)a/=a+a (6)a+=a-=a\*=a

* 1. 要将“China”译成密码，密码规律是：用原来的字母后面第 4 个字母代替原来的字母。例如，字

母’A’后面第 4 个字母是‘E’，用‘E’代替‘A’。因此，“China”应译为“Glmre”。编程，用赋初值的方法使 c1， c2，c3，c4，c5 这 5 个变量的值分别为’C’，’h’，’i’，’n’，’a’，经过运算，使 c1，c2，c3，c4，c5 分别变为’G’，’l’，’m’，’r’，’e’，并输出。

* 1. 分析下面程序的结果，并上机验证。

#include <stdio.h> void main()

{ int i，j，m，n； i=8； j=10； m=++i； n=j++；

printf(“%d，%d，%d，%d\n”，i，j，m，n)；

}

1. 实验结果

写实验报告。内容包括：

* 1. 必做习题的运行结果，源程序；
  2. 选做习题的运行结果，源程序。
  3. 程序调试中出现的错误提示。（英文、中文对照）
  4. 若有没通过的程序，分析原因。

## **3** 顺序结构程序设计

1. 了解 C 程序的基本结构。
2. 熟悉并掌握 C 的赋值语句和赋值表达式。
3. 熟悉并掌握几种输入输出函数的使用方法，包括 putchar，getchar，printf，scanf。
4. 熟悉并掌握常用的输入输出格式控制符。
5. 能够熟练使用简单的 C 语句编写顺序结构的程序。

二、必做题目

1. 输入一个圆的半径，求圆周长和圆面积。请在程序中将圆周率定义为符号常量。
2. 写一个程序，输入三个整数，分别表示日期的日、月、年，输出日期的形式为“日|月|年”。例如：输入： 1985 11 24

输出为：1985|11|24

1. 编写程序：输入摄氏温度 c，利用以下转换公式，将摄氏温度转换成华氏温度 f 输出。

转换公式为：f=(9/5)c+32

1. 输入一个直角三角形的两条直边的长度，求该直角三角形的斜边长。
2. 如下程序：输入二次方程的系数 a，b，c（输入数据的选取应保证二次方程为实根），计算并输出二次方程的两个实根。二次方程为：ax2+bx+c=0 完成并调试程序。

#include <stdio.h> void main()

{

int a，b，c； float d，v1，v2；

scanf(“%d %d %d”，&a，&b，&c)；

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_；

v1=(-1\*b+sqrt(d))/2/a；

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_；

printf(“v1=%f”，v1)；

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_；

}

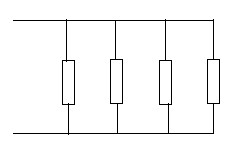
三、选做题目

1. 设圆半径 r=1.5，圆柱高 h=3，编程求圆周长，圆面积，圆球表面积，圆球体积。用 scanf 输入数据，输出计算结果，输出时要求有文字说明，取小数点后 2 位数字。
2. 编写程序：输入三个数字字符，输出由它们组成的一个整数。例如：输入‘2’、‘5’、‘7’，输出 257。
3. 编写程序：输入三个 1 位整数，输出由其组成的一个整数。例如：输入 2、4、7 输出：247。
4. 编写程序，输入三角形的三条边 a，b，c，计算并打印三角形的面积 S。 5、输入两个整数，计算它们的和数，整除后的商数。
5. 从键盘输入一个大写字母，要求改用小写字母输出。
6. 已知一个并联电路（如图所示）。如果两端电压 U=220V，R1=R2=10kΩ，R3=R4=200kΩ，求 i1， i2，i3，i4 及总电阻 r.

计算公式如下： 1/R=1/R1+1/R2+1/R3+1/R4

i1=U/R1， i2=U/R2， i3=U/R3； i4=U/R4

要求输出格式为：



R3

R2

R1

i1

i2

i3

i4

U

R= +

R1=

R2=

R3= R4

R4=

\_ 四、实验结果

写实验报告。内容包括：

1. 必做习题的运行结果，源程序；
2. 选做习题的运行结果，源程序。
3. 程序调试中出现的错误提示。（英文、中文对照）
4. 若有没通过的程序，分析原因。

## **4** 分支结构程序设计

1. 了解 C 语言表示逻辑量的方法（以 0 代表“假”，以 1 代表“真”）。
2. 学会正确使用逻辑运算符和逻辑表达式，包括掌握其运算规则，运算的优先级以及结合性。
3. 熟练掌握 if 语句和 switch 语句的语法规则。
4. 能熟练使用 if 和 switch 编制的分支程序。
5. 必做题目
   1. 编写程序：输入 a，b，c 三个整数，输出其中最大者。
   2. 编写程序：输入 3 个字母，输出 3 个字母中序号最小的字母。
   3. 编写程序：任意输入二次方程的系数 a，b，c，计算并输出方程所有可能的解。二次方程为： ax2+bx+c=0
   4. 输入一个学生百分制成绩，要求输出成绩等级‘A’、‘B’、‘C’、‘D’、‘E’。90 分以上为‘A’；80~89 分为‘B’；70~79 分为‘C’； 60~69 分为‘D’； 60 分以下为‘E’。
   5. 输入 3 个整数，要求按由小到大的顺序输出。
6. 选做题目
   1. 对一批货物征收税金，价格在 1 万元以上的货物征税 5%，在 5000 元以上、1 万元以下的货物征税 3%，在 1000 元以上、5000 元以下的货物征税 2%，1000 元以下的货物免税。编写一程序，读入货物价格，计算并输出税金。
   2. 输入 x，y，根据所处象限，用相应公式计算并输出 z。

lnx+lny 1 象限 z = sinx+siny 2 象限 e2x+e3x 3 象限 tan(x+y) 4 象限

* 1. 输入班号，输出该班学生人数。用 switch 语句和 if 语句编程。

|  |  |
| --- | --- |
| 班 号 | 91 92 93 94 95 |
| 人 数 | 30 32 30 30 32 |

* 1. 输入二次方程的 3 个系数 a，b，c，计算二次方程的根。程序只考虑实根和复根两种情形。二次方程如下：

ax2+bx+c=0

* 1. 输入一个数，验证该数能否被 2、3 和 5 整除，并显示结果。
  2. 输入 1 个字符，若是小写字母则变大写字母，若是大写字母则变小写字母。
  3. 有如下函数：
     1. (x<1)
     2. = 2x-1 (1<=x<15)

6x+x2 (x>=15) 从键盘输入 x，求函数 y 值。 8、输入一个整数，判断它能否被 3，5，7 整除，并输出以下信息：

* 1. 同时被 3，5，7 整除；
  2. 被其中两数（指出哪两个）整除；
  3. 被其中一个数（指出哪一个）整除；
  4. 能被 3，5，7 任一个整除。

9、给一个不多于 5 位的正整数，要求：

* 1. 出它是几位数；
  2. 别输出每一位数字；
  3. 逆序输出各位数字，例如原数为 321，应输出 123。

**[**解析**]**：该问题的核心是分解出每一位上的数字，然后通过检测各数字是否为零，便可知道该数字是几位数。

* 1. 编写程序：任意输入一个整数，判断它是奇数还是偶数。
  2. 编写程序：任意输入两个整数，求商（整数）和余数。如果除数为 0，给出错误提示。
  3. 输入 1 个字符，若是小写字母则转变为大写字母，若是大写字母则转杯为小写字母。
  4. 输入某年某月某日，判断这一天是这一年的第几天。
  5. 输入三个整数表示三条边的长度，判断它们能否组成一个三角形。如果能，组成的是直角三角形、等边三角形还是等腰三角形？
  6. 编写程序：要求从键盘输入两个数，并依据提示输入的数字，选择对这两个数的运算，并输出相应的运算结果。要求提示为：
  7. 作加法；
  8. 作减法；
  9. 作乘法； ⑷ 作除法。

**[**解析**]**：可使用 switch 语句，以提示输入的数字为依据，作选择结构设计，使得提示输入 1 时，将两个数之和输出；提示输入 2 时，将两个数之差输出；提示输入 3 时，将两个数之积输出；提示输入 4 时，将两个数字之商输出。注意除数不可为零的检测与提示。

1. 实验结果

写实验报告。内容包括：

* 1. 必做习题的运行结果，源程序；
  2. 选做习题的运行结果，源程序。
  3. 程序调试中出现的错误提示。（英文、中文对照）
  4. 若有没通过的程序，分析原因。

## **5** 循环结构程序设计

1. 进一步熟悉和掌握逻辑运算符和逻辑表达式的使用方法。
2. 熟练掌握 C 语言的几种循环结构：goto 语句、while 语句、do—while 语句、for 语句的特点和使用方法。
3. 熟悉并掌握 break 和 continue 的使用方法以及它们的区别。
4. 综合前面所学知识点，编写具有顺序、分支和循环结构的程序。

二、必做题目

1、下面程序可从键盘输入一个大于 0 的整数，然后输出此数的所有整数因子。例如：输入 12，则输出：12：1 2 3 4 6 12。完成程序，并进行编译和运行。

#include <stdio.h> void main()

{

int a，k；

scanf(“%d”，&a)； if (a>0) { printf(“%d:”，a)； for \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ if \_\_\_\_\_\_\_\_\_\_\_\_\_\_

printf(“%d”，k)；

}

printf(“\n”)；

}

1. 输入 10 个数，求它们的最大值并输出。
2. 求 100-999 中的水仙花数（若三位数 abc，a3+b3+c3=abc，则称 abc 为水仙花数。例如 153， 13+33+53=153，则 153 称为水仙花数。）

*x x*3 *x*5 *x*4*n*1 *x*4*n*1

1. 利用公式：sin *x*   ...  计算 sinx 的近似值，x 值由键盘输入，省

1! 3! 5! (4*n*1)! (4*n*1)!

略的项都<10-5

1. 打印九九表，格式如下：

1\*1=1 1\*2=2 1\*3=3 1\*4=4 1\*5=5 1\*6=6 1\*7=7 1\*8=8 1\*9= 9

2\*2=4 2\*3=6 …… 2\*9=18

3\*3=9 ……

……

9\*9=81

三、选做题目

1. 根据公式 e=1+1/(1!)+1/(2!)+1/(3!)+……。求 e 的近似值，精度要求为 10-6，即省略的项都<10-6。
2. 读入 10 个数，计算它们的和、积、平方和及和的平方。
3. 计算并输出 1!，2!，3!，4!，……，35!。
4. 输入 20 个数，统计其中正、负和零的个数。
5. 利用公式： 22  44  66 ...... (2*n*)2 计算的值(取前 100 项)。

2 1\*3 35 57 (2*n*1)(2*n*1)

1. 利用公式：1 1  1  1  1...... (1)*n*  1 计算的值(省略的项都<10-5)。

4 3 5 7 9 (2*n*1)

1. 编程计算 0\*1+2\*3+4\*5+……+100\*101+101\*102 之和。
2. 编程计算 5/6+6/7+7/8+……+99/100+5!+6!+7!之和。
3. 编程计算 1!-2!+3!-4!+5!-6!之和。
4. 编程计算 1/(1+2)+2/(2+3)+3/(3+4)+……100/(100+101)之和。
5. 编程计算(0+1)/1+(2+3)/3+(3+4)/4+……(99+100)/100 之和。
6. 求 100 ~ 200 中能被 3 或 7 整除的自然数。 13、统计 77 到 210 中偶数的个数。

14、统计 7 到 91 中能被 3 整除的奇数的个数。

15、7 到 91 中有多少能既能被 2 又能被 3 整除的数。

1. 显示 7 到 100 中所有不能被 5 整除的数，要求每行显示 5 个数。
2. 找出 1000 之内的所有完数。完数是指：该数的各因子之和正好等于该数本身，例如：6 的因子是 1，2，3，而 6 = 1+2+3，故 6 是完数。

**[**注意**]**：这里所说的因子，不包括该数本身。例如：6 的因子是 1，2，3。

1. 求 2~1000 中的所有亲密数对（亲密数对是指：如果 a 的因子和等于 b，b 的因子和等于 a，则（a，b）就是亲密数对）。

**[**注意**]**：这里所说的因子，不包括该数本身。例如：6 的因子是 1，2，3。

1. 用 100 元钱买 100 只鸡，已知公鸡 3 元 1 只，母鸡 1 元 1 只，小鸡 1 元 3 只，编程输出总的方案数以及每种方案中公鸡、母鸡、小鸡的数量。
2. 用 100 匹马驮 100 担货，大马驮 3 担，中马驮 2 担，小马驮 0.5 担，编程求大、中、小马的数量。
3. 编程输出如下序列的前 100 项，该序列的第一项为 0，第二项为 1，以后的奇数项为前两项之和，偶数项为前两项之差。
4. 编程验证：100 以内的奇数的平方除以 8 都余 1。
5. 编写一个程序，输出 1~100 之间所有的素数。
6. 编写程序，打印如下图案

\*

* + - \* \*
    - \* \* \* \*
    - \* \* \* \* \* \*
    - \* \* \* \*
    - \* \*

\*

1. 某化肥厂 1999 年生产化肥 3886 万吨，该厂计划年增长率为 10%，编程计算 2005 年生产多少万吨化肥。
2. 编程，求 100 到 300 之间有多少个其各位数字之和为 5 的整倍数的整数。
3. 输入两个正整数，求它们的最大公约数和最小公倍数。
4. 打印九九表，格式如下：

1\*1=1 1\*2=2 1\*3=3 1\*4=4 1\*5=5 1\*6=6 1\*7=7 1\*8= 8 1\*9=9

2\*2=4 2\*3=6 2\*4=8 …… 2\*9=18

3\*3=9 3\*4=12 …….

……

9\*9=81

1. 修改下列程序，使之实现输入 10-9999 范围内的一个整数，以该整数各位数字相反的顺序输出。

例如：若输入 934，则输出 439；若输入 120，则输出 021

#include <stdio.h> void main()

{ int a； scanf(“%d”，&a)；

while (a==0)

{

printf(“%d \n”，a/10)； a=a%10；

}

printf(“\n”)； }

30、某数列的前 3 个数为 0，0，1，从第 4 个数开始，每个数是它的前 3 个数之和。下面程序可求出并输出此数列的前 10 个数。完成程序并运行。

#include <stdio.h> void main()

{

int a，b，c，x，n； a=0； b=0； c=\_\_\_\_\_\_\_\_\_\_； n=3； printf(“%d %d %d”，a，b，c)； while (n<=10)

{

x=a+b+c； a=b； b=c；

\_\_\_\_\_\_\_\_\_\_\_\_\_\_； printf(“%d”，x)；

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_；

}

printf(“\n”)；

}

31、写出以下程序实现的功能。

#include <stdio.h>

void main()

{ int i； float fact； fact=2；

i=1； do

{

i=i+1；

fact=fact\*I； }while(i<=10)； printf(“%f”，fact)；

}

32、输入一行字符，分别统计出其中英文字母、空格、数字和其它字符的个数。

四、实验结果

写实验报告。内容包括：

1、必做习题的运行结果，源程序； 2、选做习题的运行结果，源程序。 3、程序调试中出现的错误提示。（英文、中文对照） 4、若有没通过的程序，分析原因。

## 实验 **6** 数组

1. 实验目的
   1. 熟练掌握一维数组的定义、初始化、赋值、输入输出的方法。
   2. 掌握二维数组的定义、初始化、赋值、输入输出的方法。
   3. 熟练掌握字符数组的定义、初始化、赋值的方法；
   4. 熟练掌握对字符数组逐个访问和整体访问的方法。
   5. 熟悉常用的字符串函数。
   6. 掌握与数组有关的程序与算法。
2. 必做题目
   1. 用冒泡排序法对 15 个整数由大到小进行排序。
   2. 输出如下杨辉三角。

1

[1 1](#_Toc113207)

[1 2 1](#_Toc113208)

[1 3 3 1](#_Toc113209)

1 4 6 4 1

1 5 10 10 5 1

* 1. 编写一个程序，将两个字符串连接起来，不要用 strcat 函数。
  2. 将字符串 a 中下标值为偶数的元素由小到大排序，其他元素不变。
  3. 用字符数组存储任意输入的 10 个字符，求其最大字符。

1. 选做题目
   1. 将一个数组的元素初始化为有序的，要求输入一个数后，按原来排序的规律将它插入数组中。

例如，数组中的元素为：1 2 4 4 5 10 12，输入整数 3 后，数组元素变为：1 2 3 4 4 5 10 12

* 1. 编写程序， 实现矩阵（3 行 3 列）的转置(即行列互换) ，例如， 输入下面的矩阵:

100 200 300

400 500 600

700 800 900

程序输出：

100 400 700

200 500 800

300 600 900

* 1. 编程：已知 5 个学生的 3 门功课的成绩。计算每一个学生的总分及平均分，统计平均成绩低于

60 分的学生人数。

4、编写程序，对任意输入的字符串，统计其中的大写字母和小写字母的个数。例如， 输入字符串：

AAaaBBb123CCccccd，则应输出结果：upper = 6， lower = 8。 5、下面程序输入一字符串，并将其中的大写字母改变成小写字母，完成程序。

#include <stdio.h> void main()

{ int i=0 ；

char s[120] ；

printf ( "Enter a string.\n") ； scanf( "%s"， s )； while ( \_\_\_\_\_\_\_\_\_\_ )

{ if ( \_\_\_\_\_\_\_\_\_\_\_ ) s[i]=s[i]-'A'+'a'；

i++； } printf( "%s\n"， s )；

}

6、定义两个二维数组（如 A[2][3]、B[2][3]），分别对这两个数组输入数据，求 A+B 和 A-B 的值。 7、求一个 3\*3 的整型矩阵对角线元素之和。

1. 将一个数组中的值按逆序重新存放。例如，数组元素原来的顺序为 8，6，5，4，1，运行程序，将数组元素修改为：1，4，5，6，8。
2. 有一篇文章，共有 3 行文字，每行有 80 个字符。要求分别统计出其中英文大写字母、小写字母、数字、空格以及其它字符的个数。
3. 编写一个程序，将字符数组 s2 中的全部字符复制到字符数组 s1 中，不用 strcpy 函数。注意，复制时，’\0’也要复制过去。‘\0’后面的字符不复制。
4. 输入 a1，a2，a3，……，a20，将它们从小到大排序后输出，并给出现在每个元素所对应的原来次序。例如：输入：27，3，25，27，14，39 输出： 3 2 14 5

25 3

27 1

27 4

39 6

1. 输入一串字符，以‘？’结束，统计各字母出现的次数，并按字母出现的多少输出（先输出字母出现多的，次数相同的按字母表顺序输出，不出现的字母不输出）。例：输入：5b3a+4-hdeh5dh?

输出： h 3 d 2 a 1 b 1 e 1

1. 输入矩阵 a（5 行 5 列），完成下列要求：
2. 输出矩阵 a；
3. 将 a 的第 2 行和第 5 行元素对调后，形成新的矩阵 a 并输出；
4. 用对角线（指矩阵的左上角到右下角的对角线）上的元素分别去除相应行的元素，又形成一个新的矩阵 a 并输出。
5. 在一个数组中存放 20 个整数，统计并输出该数组中素数的个数。
6. 任意输入 20 个数到一维数组 a 中，求 20 个数的平均值。
7. 用字符数组存储任意输入的一串字符，求其中最大字符。
8. 输入 a1，a2，a3，……，a10：
   1. 将它们从小到大排序输出：用改进的冒泡排序完成
   2. 将它们从大到小排序输出：用选择排序和改进的选择排序完成
9. 如下程序，程序功能是：在 n 行 n 列的矩阵中，每行都有最大的数，程序求这 n 个最大数中的最小值，完成程序并调试。 #include <stdio.h> #define N 5 int a[N][N]； void main()

{

int row，col，max，min；

//输入 N×N 个整数到数组 a 的代码略

for(row=0；row<N；row++)

{ for(max=a[row][0]，col=l；col<N；col++) if (\_\_\_\_\_\_\_\_\_\_) max=a[row][col]； if (\_\_\_\_\_\_\_\_\_\_\_) min=max； else if(\_\_\_\_\_\_\_\_\_\_) min=max；

} printf ("The min of max numbers is %d\n"，min)；

}

1. 编写程序， 程序的功能是: 移动一维数组中的内容； 若数组中有 n 个整数， 要求把下标从 0 到 p(含 p，p 小于等于 n-1)的数组元素平移到数组的最后。例如， 一维数组中的原始内容为: 1，2，3，4， 5，6，7，8，9，10； p 的值为 3。移动后， 一维数组中的内容应为: 5，6，7，8，9，10，1，2，3，4。
2. 编写程序，将两个两位数的正整数 a、b 合并形成一个整数放在 c 中。合并的方式是：将 a 数的十位和个位数依次放在 c 数的十位和千位上， b 数的十位和个位数依次放在 c 数的个位和百位上。例如，当 a＝45，b=12。则 c=5241。

四、实验结果

写实验报告。内容包括：

1、必做习题的运行结果，源程序； 2、选做习题的运行结果，源程序。 3、程序调试中出现的错误提示。（英文、中文对照） 4、若有没通过的程序，分析原因。

## 实验 **7** 函数

1. 实验目的
   1. 掌握函数定义的方法。
   2. 掌握函数实参与行参的对应关系，以及“参数传递”的方式。
   3. 掌握函数的嵌套调用和递归调用的方法。
   4. 掌握全局变量和局部变量、动态变量、静态变量的概念和使用方法。
2. 必做题目
   1. 分析并上机验证下列变量的值。

#include <stdio.h> void main()

{

int a=1，b=2，c=3；

printf("1---a=%d，b=%d，c=%d\n"，a，b，c)；

{

int a，b，c； a=10，b=20，c=30；

printf("2---a=%d，b=%d，c=%d\n"，a，b，c)；

}

printf("3---a=%d，b=%d，c=%d\n"，a，b，c)；

}

1. 写一个函数：求任意两个正整数最大公约数。在主函数 main 中输入两个正整数，使用该函数求其最大公约数。建议函数原型： int GreatestCommonDenominator(int intPositiveNumberA，int intPositiveNumberB)。
2. 编写一个函数，接收两点的坐标，计算并返回这两点之间的距离。建议函数原型： double Distance(int x1，int y1，int x2，int y2)；
3. 写一个温度转换函数，能将摄氏温度转换成华氏温度，也可以将华氏温度转换成摄氏温度。在主函数 main 中输入一个温度值并指明类型（‘C’或‘c’代表摄氏温度，‘F’或‘f’代表华氏温度），调用该函数，计算转换后的温度。建议函数原型：

double TemperatureTrans(double dblTemperature，char cType)；

5．编写一个函数：判断一个整数的奇偶性。建议函数原型： int isOdd\_or\_Even(int Number)；

三、选做题目 **1**

1、阅读以下程序，分析该程序的输出结果，思考函数 void s(int b[]， int n)的功能是什么。

void s(int b[]， int n)

{

int i，j，t，flg；

for(i=0；i<n-1；i++)

{

for(flg=0，j=0；j<n-i-1；j++) if(b[j]>b[j+1])

{

t=b[j]；b[j]=b[j+1]；b[j+1]=t；

flg=1；

}

if (!flg) break；

}

}

void main()

{

int i，a[]={11，9，2，5，3，7}；

s(a，6)；

for(i=0；i<6；i++) printf("%4d"，a[i])；

printf("\n")；

}

2、阅读以下程序，分析该程序的输出结果，思考函数 int m(unsigned k)的功能是什么。

int m(unsigned k)

{

int s=1； do

{

s\*=k%10； k/=10；

} while(k)； return s；

}

void main()

{

printf("%d，%d\n"，m(1234)，m(43209))；

}

1. 写一个函数：求一个实数的整数次幂。在主函数 main 中输入一个实数 x 和正整数 m，调用该函数求这个实数 x 的 m 次幂。建议函数原型：

double [Power(](http://dict.baidu.com/s?wd=factorial)double dblRealNumber，int intPositiveNumber)；

1. 编写函数，找出任意三个实数中的最大值。建议函数原型：

double FindMax(double dblOne，double dblTwo， double dblThree)；

1. 写一个函数，求任意两个正整数最小公倍数。在主函数 main 中输入两个正整数，使用该函数求其最小公倍数。建议函数原型：

int LowestCommonMultiple(int intPositiveNumberA，int intPositiveNumberB)；

1. 编写一个函数 CircleArea()，接收半径 r，计算并返回这个圆的面积。建议函数原型： double CircleArea(double r)；
2. 编写一个函数 CylinderVolume()，接收半径 r 和高 h，计算并返回圆柱的体积。建议函数原型： double CylinderVolume (double r，double h)；
3. 编写一个函数，使字符串的反序，在主函数 main 中调用该函数对输入的字符串进行反序。建议函数原型：

void stringInvert(char str[])；

1. 编写一个函数，将一个字符串中的元音字母复制到另一字符串，然后输出。建议函数原型： void strVowel(char strOld[]，char Vowel[])；
2. 编写函数，由实参传来一个字符串，统计该字符串中字母、数字、空格和其它字符的个数，在主函数中输入字符串并输出统计的结果。建议函数原型： void strStatistics (char strings[]，int pNum[])；
3. 编写一个求阶乘的函数。在主函数中输入正整数 n，调用该函数求 n 的阶乘。建议函数原型： long [Factorial(](http://dict.baidu.com/s?wd=factorial)long intPositiveNumber)；
4. 写一个函数，用“冒泡排序法”对 n 个整数从小到大排序。编写主函数 main，使用随机函数产生大小处于 100~350 之间的 20 个随机整数，调用该函数进排序并输出排序后的结果。建议函数原型： void Sort(int intArray[]，int intLength)；
5. 编写函数，实现方阵的转置（即实现矩阵第 i 行 j 列与第 j 行 i 列的互换）。建议函数原型： void MatrixInvert(double maxtrix[][]， int n)；

四、选做题目 **2**

1. 在一个数组 data 中存放 n(n<=1024)个整数，编写函数，判断该数组中哪些是素数，并统计素数的个数，在主函数中输出素数的个数和这些素数。建议函数原型： int IsPrime(int \*data， int \*primes，int len)；
2. 用字符数组存储任意输入的 n(n<=1024)个字符，编写函数，把该字符数组中的小写字母转换成大写字母。建议函数原型：

void [UpperCase(](http://dict.baidu.com/s?wd=factorial)char \*charArray)；

1. 在主函数中输入年、月、日，编写函数，判断输入的日期是该年的第几天，并在主函数中输出判断的结果。建议函数原型：

int CountDays(int year，int month，int day)；

1. 编写函数实现两个数的交换。在主函数中调用该函数对输入的两个数进行交换。建议函数原型： void Change(double \*pFirstNumber， double \*pSecondNumber)
2. 编写函数，求 n 阶矩阵中最大元素与最小元素的值。建议函数原型：

void MatrixSearch(double maxtrix[][]， double \*pMax，double \*pMin，int n)；

1. 二分法是一种快速查找算法。其方法为：初始查找区间的下界为 0，上界为 len-1，查找区间的中间部分，k = (下界+上界)/2 ； 若 list[k] 等于 c ，查找成功；若 list[k] > c，则新的查找区间的下界不变，上界改为 k-1； 否则新的查找区间的下界改为 k+1，上界不变。在新区间内继续用二分法查找。如下函数：在已按字母次序从小到大排序的字符数组 list[len] 中，查找字符 c，若 c 在数组中，函

数返回字符 c 在数组中的下标，否则返回 -1。将函数补充完整，并实现在主程序中的调用： int search(char \* list，char c，int len)

{

int low=0，high=len-1，k ；

while ( \_\_\_\_\_\_\_\_\_ )

{

k=(low+high)/2； if ( \_\_\_\_\_\_\_\_ ) return k； else if ( \_\_\_\_\_\_\_\_ ) high=k-1； else low=k+l；

}

return -1 ；

}

7、编写三个函数：

1. 输入 10 个职工的姓名和职工号，建议函数原型：

void [Input(](http://dict.baidu.com/s?wd=factorial)char Name[10][20]， int WorkNo[10])；

1. 按职工号由小到大排序，姓名顺序也随之调整， 建议函数原型：

void [Sort(](http://dict.baidu.com/s?wd=factorial)char Name[10][20]， int WorkNo[10])；

1. 要求输入一个职工号，用折半查找法找出该职工的姓名， 建议函数原型： int [Search(](http://dict.baidu.com/s?wd=factorial)char Name[10][20]， int WorkNo[10]，char \*pName)。

**[**要求**]**：在主函数中调用函数输入 10 个职工的姓名和职工号；调用函数进行排序；然后从键盘上输入要查找的职工号，输出该职工姓名。

1. 在主函数中定义二维数组 A[3][4]、B[3]，用子函数对数组 A 的每一行求和，将其值放在数组 B 中，在主函数中输出该值。建议函数原型： void [LineAdd(](http://dict.baidu.com/s?wd=factorial)int A[3][4]，int B[])；
2. 输入 10 个学生的 5 门课程的成绩，分别编写函数求；
3. 每个学生的平均分；建议函数原型：

void [StudentAverage(](http://dict.baidu.com/s?wd=factorial)double Score[10][5]，double stuAverage[10])；

1. 每门课程的平均分；建议函数原型：

void [ScoreAverage(](http://dict.baidu.com/s?wd=factorial)double Score[10][5]，double ScoAverage[5])；

1. 找出最高的分数所对应的学生和课程。建议函数原型：

void Max[Score(](http://dict.baidu.com/s?wd=factorial)double Score[10][5]，int \*pStuNo，double \*pScore)；

1. 编写一个名为 date 的函数，接收一个 yyyymmdd 形式的整数（如 20070412），确定相应的月、日、年的值，并向调用函数返回这个值。例如：如果函数通过语句调用 date(20120411，&month，&day， &year)，则数字 4 应在 month 中返回，数字 11 应在 day 中返回， 数字 2012 应在 year 中返回。建议函数原型：

void date(int intDateNumber，int \*pMonth，int \*pDay，int \*pYear)

1. 写一个函数，输入一行字符，将此字符串中最长的单词输出。
2. 函数 fun 的功能是：求 k!（k<13），所求阶乘的值作为函数值返回。例如：若 k = 10，则应输出： 3628800。请补充程序，使它能得出正确的结果。注意：不要改动 main 函数，不得增行或删行，也不得更改程序的结构！

#include <conio.h> #include <stdio.h> long fun (int k)

{

if(k>0)

\_\_\_\_\_\_\_\_\_\_ ； else if(k==0)

\_\_\_\_\_\_\_\_\_ ；

}

void main()

{ int k = 10；

printf("%d!=%ld\n"，k，fun (k)) ；

}

1. 若已定义：int a[10]，i；以下 fun 函数的功能是：在第一个循环中给前 10 个数组元素依次赋值

1、2、3、4、5、6、7、8、9、10；在第二个循环中使 a 数组前 10 个元素中的值对称折叠，变成 1、2、

3、4、5、5、4、3、2、1。完成下面函数并实现在主程序中的调用：

fun (int a[])

{

int i； for(i=1；i<=10；i++) =i； for(i=0；i<5；i++)

=a[i]；

}

1. 如下函数；递归函数 dec(int a[]，int n)是判断数组 a[]的前 n 个元素是否是不递增的。不递增返回 1，否则返回 0。将函数补充完整，并实现在主程序中的调用。

int dec(int a[]，int n)

{ if (n<=1) \_\_\_\_\_\_\_\_\_\_；

if (a[0]<a[1]) return 0；

return \_\_\_\_\_\_\_\_\_\_；

}

1. 写一个递归函数求一个实数的整数次幂。在主函数 main 中输入一个实数 x 和正整数 m，调用该函数求这个实数 x 的 m 次幂。建议函数原型：

double [Power(](http://dict.baidu.com/s?wd=factorial)double dblRealNumber，int intPositiveNumber)；

五、实验结果

写实验报告。内容包括：

1、必做习题的运行结果，源程序； 2、选做习题的运行结果，源程序。 3、程序调试中出现的错误提示。（英文、中文对照） 4、若有没通过的程序，分析原因。

## 实验 **8** 指针

1. 实验目的
   1. 掌握指针的概念，指针变量的定义和作用。
   2. 熟悉和了解 C 语言指针一般设计方法；学会使用 C 语言指针进行程序设计。
   3. 学会使用数组的指针和指向数组的指针变量。
   4. 学会使用字符串的指针和指针数组。
   5. 学会使用指向函数的指针变量。
   6. 了解指向指针的指针的概念及其使用方法。
2. 必做题目
   1. 运行下面的程序，写出运行结果，注意指针的使用方式。

#include<stdio.h> void main()

{

int a[]={1，3，5}，x=7，y=9； int \*p=NULL； p=a；

printf("%d， "，\*p)； printf("%d， "，\*(++p))； printf("%d， "，\*++p)； printf("%d， "，\*(p--))； printf("%d， "，\*p--)； printf("%d， "，\*p++)； printf("%d， "，++(\*p))； printf("%d\n "，(\*p)++)； p=&a[2]； printf("%d， "，\*p)； printf("%d， "，\*(++p))； p++；

printf("%d， "，\*p)；

}

2、运行下面的程序，写出结果，注意指针的使用方式。

#include<stdio.h> void main()

{

static int a[6]={1，2，3，4}；

int i，s=1； int \*p=NULL； p=a；

for(i=0；i<6；i++) s\*=\*(p+i)； printf("%d\n"，s)；

}

3、运行下面的程序，写出结果，注意指针的使用方式。

#include<stdio.h> void main()

{

char a[]="abcdef"； char \*b="ABCDEF"；

int i； for(i=0；i<3；i++)

printf("%d，%s\n"，\*a，b+i)；

printf("--------------------\n")； for(i=3；a[i]；i++){ putchar(\*(b+i))；

printf("%c\n"，\*(a+i))；

}

}

1. 输入两个数，求它们的和及乘积，用指针的方法实现。
2. 定义一个一维数组，输入、输出数组全部元素的值，用指针的方法实现。

三、选做题目 **1**

1、运行下面用指针描述的程序，写出运行结果，注意指针的使用方式。

⑴

#include <stdio.h> void main()

{

int i，j，a[4][3]={{1，2，3}，{4，5，6}，{7，8，9}，{10，11，12}}； int \*p=NULL；

printf("\n%d\t%d\t%d\t%d\n"，a[0]，a[1]，a[2]，a[3])； for(p=a[0]+2，i=0；i<10；i++) printf("%5d"，\*p++)； printf("\n")；

for(i=0；i<4；i++)

{

printf("%d"，\*(a+i))； for(j=0，p=\*(a+i)+j；j<3；j++)

printf("%5d"，\*p++)； printf("\n")；

}

}

⑵

#include <stdio.h> void main()

{

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

int (\*p1)[3]；

p1=a；

printf("\n")；

printf("1:%d\n"，\*(\*(p1+0)))；

p1++；

printf("2:%d\n"，\*p1[0])； printf("3:%d\n"，\*(\*(p1+1)+2))；

}

⑶

#include <stdio.h> void main()

{ void tran(int n，int x[])；

int a[4][4]={{3，8，9，10}，{2，5，-3，5}，{7，8，9，12}，{10，11，2，4}}； tran(2，a[0])； tran(0，a[2])；

tran(0，&a[2][0])；

}

void tran (int n，int arr[])

{ int i； for(i=0；i<4；i++) printf("%d"，arr[n\*4+i])；

printf("\n")；

}

2、阅读以下程序，思考该程序的作用及输出结果。

#include <stdio.h> void fun(int \*x，int \*y)

{

printf("%d %d"，\*x，\*y)；

\*x=3；

\*y=4；

}

void main()

{

int x=1，y=2； fun(&y，&x)； printf("%d %d"，x，y)；

}

1. 写一函数，求一个字符串的长度，在主函数中输入字符串，并输出其长度（不能用已有的字符串函数）。建议函数原型：

int StringLenth(char \* string)；

1. 任意输入 n 个数，定义函数，找出其中的最大值和最小值，并在主函数中输出这两个值。建议函数原型：

void MaxAndMin(int \*data，int \*max，int \*min，int len)；

1. 编写一个函数，判断任意输入的字符串中，有多少个字母。建议函数原型： int CountNumofLetter(char \*string)；
2. 定义可以存放 n 个元素的一维整型数组，编写函数，删除数组中指定的某元素（假设所有元素值是唯一的，且被删除元素存在于数组中）。建议函数原型： int Delete(int \*data，int len)；
3. 定义可以存放 n(n<=1024)个元素的一维整型数组，编写函数，在数组中查询指定的某元素（假设所有元素值是唯一的），函数返回该元素的位置信息。建议函数原型： int SearchKey(int \*data，int len)；
4. 定义可以存放 n(n<=1024)个元素的一维字符数组，编写函数，删除数组中指定的某字符（假设所有元素值是唯一的，且被删除元素存在于数组中）。建议函数原型： void Delete(char \*string)；
5. 定义可以存放 n(n<=1024)个元素的一维字符数组，编写函数，在数组中查询指定的某字符（假设所有元素值是唯一的），函数返回该元素的位置信息。建议函数原型： int SearchKey(char \*string)；
6. 将 n(n<=1024)个数按输入时顺序的逆序排列， 用函数实现。建议函数原型： void ReverseData(int \*data， int len)；
7. 将一个整数字符串转换为一个整数，如“-1234”转换为-1234。建议函数原型： int StringToInt(char \*string)；
8. 输入 n(n<=1024)个整数，将其中最小的数与第一个数对换，把最大的数与最后一个数对换。写 3 个函数：
9. 输入 n 个数。建议原型：void InputNumbers(int \*p，int len)；
10. 进行处理。建议原型：void SwapMaxMin(int \*p，int len) ；
11. 输出 n 个数。建议原型：void Display(int \*p， int len)；

13、阅读以下程序，分析其作用，学习其编程方法。

#include <stdio.h> #include <string.h> void strsort(char \*s[]，int n)

{

int i，j，k； char \*temp=NULL；

for(i=0；i<n-1；i++)

{

k=i；

for(j=i+1；j<n；j++) if(strcmp(s[j]，s[k])<0)

k=j；

if(k!=i)

{

temp=s[i]；s[i]=s[k]；s[k]=temp；

}

}

}

void main()

{

char \*s[10]，\*\*p=NULL； int i； for(i=0；i<10；i++) gets(s[i])； strsort(s，10)；

p=s；

for(i=0；i<10；i++)

printf("%s\n"，\*p++)；

}

四、选做题目 **2**

1. 如下函数：strcpy(char \*to，char \*from)将字符串 from 复制到字符串 to.将函数补充完整并实现在主程序中的调用。

void strcpy(char \*to，char \*from)

{

while (\_\_\_\_\_\_\_\_\_\_\_\_)；

}

1. 如下函数：merge(int a[]，int n，int b[]，int m，int \*c)是将两个从小到大有序数组，a 和 b 复制合并出一个有序整数序列 c，其中形参 n 和 m 分别是数组 a 和 b 的元素个数.将函数补充完整并实现在主程

序中的调用。

void merge(int a[]，int n，int b[]，int m，int \*c)

{ int i，j；

for (i=j=0；i<n&&j<m；) \*c++=a[i]<b[j]?a[i++]:b[j++]； while (\_\_\_\_\_\_\_\_\_\_\_\_) \*c++=a[i++]； while (\_\_\_\_\_\_\_\_\_\_\_\_) \*c++=b[j++]； }

1. 有一字符串，包含 n(n<=1024)个字符，写一函数，将此字符串中从第 m 个字符开始的全部字符复制成为另一个字符串。建议函数原型：

void Mystrcpy(char \*src， char \*des， int m)；

1. 分别用行指针和列指针法寻找 3\*4 矩阵中的最大值。建议函数原型：
2. int FindMax(int (\*data)[])； //行指针法
3. int FindMax(int \*data)； //列指针法
4. 编写函数实现 n 阶方阵的转置（即实现矩阵第 i 行 j 列与第 j 行 i 列的互换），并编写主函数进行测试。建议函数原型：

void MatrixInvert(double (\* maxtrix)[]，int n)；

1. 定义一个具有 n(n<=1024)个元素的一维整型数组，编写函数，将其中最小的数与第一个数对换。

建议函数原型：

void MinChange(int \*data，int len)；

1. 编写函数，对具有 n 个整数的一维数组中的元素进行重新排列，要求：奇数序号的放在数组的前面，偶数序号的放在数组的后面。例如：a[4]={1，2，3，4}，排列后为：a[4]={1，3，2，4}。建议函数原型：

void DataResort(int \*data，int len)；

1. 将 5 x 5 整数矩阵中的最大数放在中心位置，4 个角上分别放上 4 个最小的元素（顺序为从左到右，从上往下，依次从小到大存放），用函数与指针实现。建议函数原型：
2. void InputMatrix(int \*p， int n)；//输入整数矩阵
3. void MatrixMaxMin(int \*p)；//调整元素位置
4. void DisplayMatrix(int \*p， int n)；//输出矩阵元素值
5. 有 4 个学生，5 门课程。输入所有学生的所有课程成绩。针对如下每个功能分别编写一个函数，尽量以指针方式实现。
6. 求第一门课程的平均分，建议函数原型：

double Average1(double (\*p)[5])；

1. 统计有两门及其以上课程不及格的学生人数，建议函数原型：

int CountLowScore(double (\*p)[5])；

1. 统计平均成绩在 90 分以上的学生人数，建议函数原型：

int CountAverScore(double (\*p)[5])；

1. 在主函数中输入 n(n<=100)个等长的字符串，用另一个函数对它们排序，然后在主函数中输出这 10 个已经排好序的字符串。建议函数原型：
2. void InputString(char \*(p[])， int n)； //输入 n 个字符串
3. void SortString(char \*(p[])，int n)； //对 n 个字符串进行排序
4. void DisplayString(char \*(p[])，int n)； //输出排序后的字符串
5. 输入一个字符串，将该字符串中连续的数字字符作为一个整数，依次存放到一个数组中。统计该字符串中有多少整数，并输出这些整数。

例如： 字符串“d43we56hes 5 897^&9kl”中有整数：43，56，5，897，9

建议函数原型：

int Numbers(char \*pStr， int \*pNum)； //统计字符串 pStr 中的数字，将数字存入 pNum 中，并将数字个数通过函数返回

1. 编程实现字符串比较函数: int MyStrcmp(char \*p1，char \*p2)。其中，p1 和 p2 分别指向两个字符串，如果这两个字符串相当则函数返回 0，如果这两个字符串不相等则返回二者第一个不同字符的 ASCII

差值。例如：“asd”和“awe”第二个字符不同，则返回：'s'-'w'=115-119 = -4

1. 以下程序用指针数组编程实现：输入月份号，输出该月的英文名：例如：输入数字 4，输出显示“April”。阅读程序并填空。

#include <stdio.h>

//字符指针数组，用于存放月份的英文单词

char \*pMonth[12]={{"January"}，{"February"}，{"March"}，{"April"}，{"May"}， {"Jule"}， {"July"}，{"August"}，{"September"}，{"October"}，{"November"}，{"December"}}； void main()

{

while⑴ //无限循环月份输入

{ printf("请输入月份数字(非法月份自动退出)：\n\t")； int nMonth； scanf("%d"，&nMonth)；

if(nMonth<1 || nMonth>12)//非法月份输入，退出循环 break；

printf("\t%s\n"，\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_)；//显示数字对应的单词

}

}

14、函数指针的练习：如下程序的功能是：用矩形法分别求取正弦、余弦、以及自然指数在区间[0， 1]的定积分。用函数指针来实现。请阅读程序并将其补充完整。注：程序包含头文件 math.h，因为里面有各类已定义好的数学函数

#include <stdio.h>

#include <math.h>

#define SIN 1 //正弦标志常量

#define COS 2 //余弦标志常量

#define EXP 3 //自然指数标志常量

//求 sinx，cos， exp 在 min 和 max 之间的定积分，矩形宽度为 h 越小精度越高

double Integral(int flag， double min， double max， double h)

{ double result=0； //积分结果 double (\*pfn)(double)； //函数指针 if(flag==SIN) //正弦 pfn = sin；

else if(flag==COS)//余弦 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_；

else if(flag==EXP)//自然指数 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_； else

return 0；

while(\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_)

{

result += \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_； min += h；

}

return result；

}

void main()

{ double x = Integral(SIN，0，1，0.001)； double y = Integral(COS，0，1，0.001)； double z = Integral(EXP，0，1，0.001)； printf("x∈[0，1]， ∫sinxdx = %f\n"，x)； printf("x∈[0，1]， ∫cosxdx = %f\n"，y)；

printf("x∈[0，1]， ∫expxdx = %f\n"，z)；

}

15、通过键盘输入任意个整数存放于动态分配的连续内存中，并把该连续内存中所有为偶数的数，放在连续内存的前部，奇数存放于连续内存的后部。并分别对奇数部分和偶数部分进行排序。例如，指针\*p 动态分配内存并输入数据{7，4，3，2，5，12，1，11，8，6}，将偶数放在前部并排序，奇数放在后部并排序，由此得到的结果为：{2，4，6，8，12，1，3，5，7，11}。建议函数原型：

1. void InputNumbers(int \*pData， int length)； //输入 length 个整数 到指针 pData 指向的内存中
2. void ClassifyAndSort(int \*pData， int length)； //对 pData 指向的 length 个整数进行奇偶分类分离，并分别排序
3. void DisplayNumbers(int \*pData， int length)； //显示奇偶分离及排序后的结果

16、编写函数，用指向指针的指针的方法将 5 个字符串排序输出。建议函数原型：

1. void InputString(char \*\*p， int n)； //输入字符串
2. void DisplayString(char \*\*p，int n)； //显示字符串
3. void SortString(char \*\*p， int n， int dir)； //排序字符串，其中，p: 双重指针，为待排序的字符串； n: 待排序字符串的个数；dir: 升序或降序标记。
4. 有 n 个人围成一圈，顺序排号。从第 1 个人开始报数（从 1 到 3 报数），凡报到 3 的人退出圈子，问最后留下的是原来第几号的那位。
5. 编写函数将整数 x 的所有偶数因子存放在内存中（假设内存足够大）；将偶数因子的个数通过函数返回值返回。在主函数中输入整数 x，并输出 x 所有的偶数因子。建议函数原型：

int EvenFactor(int x， int \*pEF)； //将 x 的所有偶数因子，放到指针\*pEF 指向的内存中

1. 编写函数，实现如下功能：
2. 动态输入 n 个整数存放在动态分配的内存中，建议函数原型： void InputNumbers(int \*pData， int n)
3. 将整数序列中的 0 元素移到最后，其余非 0 元素按原次序依次向前移动，例如：{1，0，9，0，5，

7，6，0，5} 转换为 {1，9，5，7，6，5，0，0，0}。建议函数原型：

void ZeroProcess(int \*pData， int n)；// pData 指向输入的整数序列，n 为整数的个数

1. 将处理后的整数输出显示，建议函数原型： void ShowNumbers(int \*pData， int n)

20、编写函数，完成如下功能：

1. 在内存中圈定连续固定长度的内存空间（如：定义数组圈定 1000 字节的内存）
2. 在该圈定内存范围内，进行模拟动态内存管理（即动态分配与释放）
3. 自定义函数: char\* MyAlloc(int len)； 进行动态内存分配，len 为分配字节个数，返回值为分配内存首地址，内存分配失败返回 NULL

⑷自定义函数：void MyFree(char \*p)；进行动态内存释放

**[**要求**]**：进行完善的动态内存管理，特别是内存的反复分配与释放，如何标识内存分配与否状态，如何判定是否有足够内存用于分配（即剩余的连续内存是否足够用于分配）。

**[**注意**]**：不能使用已有的动态内存管理函数 alloc()，frre()等。

**[**提示**]**：可另外定义一个等长度的 int 数组，用于标识圈定内存中每个内存空间的分配与否的状态，这样方便进行处理：如计算当前最大剩余连续空间长度

五、实验结果

写实验报告。内容包括：

1. 必做习题的运行结果，源程序；
2. 选做习题的运行结果，源程序。
3. 程序调试中出现的错误提示。（英文、中文对照）
4. 若有没通过的程序，分析原因。

## 实验 **9** 结构体、联合体

1. 实验目的
   * 1. 掌握结构体类型变量的定义和使用方法。
     2. 会使用指向结构体的指针对结构体的成员进行操作。
     3. 掌握联合体类型变量的定义和使用方法。
     4. 理解结构体和联合体相互嵌套的使用方法。
     5. 理解链表的概念和结构。
2. 必做题目
   * 1. 编写程序，将下列数据赋予结构体变量，并将它们输出。

|  |  |  |
| --- | --- | --- |
| 姓名 | 年龄 | 月薪 |
| 李明 | 25 | 2500 |
| 王利 | 22 | 2300 |
| 赵勇 | 30 | 3000 |

* + 1. 有五个学生，每个学生的数据包括学号、姓名、三门课的成绩。从键盘输入 5 个学生的相关数据，要求打印出三门课的总平均成绩，以及最高分的学生的数据（包括学号，姓名、三门课的成绩，三门课的平均分）。要求使用结构体变量。
    2. 阅读以下程序，分析其输出结果。

#include <stdio.h>

union myun

{

struct

{int x，y，z；} u； int k；

}a；

void main()

{

a.u.x=4；

a.u.y=5；

a.u.z=6；

a.k=0；

printf("%d\n"，a.u.x)；

}

* + 1. 编写一个 C 语言程序，提示用户输入当前的年、月和日期，把输入的数据保存到一个恰当定义的结构体中，并用一种恰当的方式显示数据。
    2. 编写一个程序，使用一个结构体保存股票名称、每股预估的收益以及预估的股价收益比。使这个程序提示用户输入 5 支不同的股票项，每次使用相同的结构保存这些输入的数据。当输入某个特定的股票数据时，让程序根据输入的收益和股价收益比计算并显示预期的股票价格。例如，如果用户输入数据

XYZ 1.56 12，则 XYZ 股票的每股预期价格是 1.56\*12=18.72 元。

1. 选做题目

1、为下列每个记录声明一个结构类型，在主函数中初始化后用 printf 输出结果。

* + - 由学生的学号、完成的学分和累积成绩分数平均值组成的学生记录。
    - 由学生的姓名、出生日期、完成的学分数和累积成绩分数平均值组成的学生记录。
    - 由人名和地址（街道、城市、国家和邮编）组成的邮件发送清单。
    - 由股票名称、股票价格和购买日期组成的股票记录。
    - 由整形零件号码、零件名称、库存量，整形记录员编号组成的库存记录。
    1. 编写一个 C 语言程序，接收用户用小时和分钟输入的时间。使程序计算并显示一分钟后的时间。
    2. 定义一个 5 个结构的数组，用来保存一个码头中已经停靠的小船的数据，小船数据包括姓名、小船行驶证号、小船长度和当前停靠的码头号。在主函数中通过键盘输入靠岸的小船的信息，然后通过 printf 输出到屏幕上。
    3. 使用声明

struct MonthDays

{

char name[10]； int days；

}；

定义 12 个类型为 MonthDays 的结构。命名这个数组为 conver[]， 并用一年中 12 个月份的名称和每个月中的天数初始化这个数组。接收来自一个用户的数字形式的月份，显示这个月份的名称和这个月份的天数。比如，如果输入的是 3，则程序应该显示“March has 31 days”（三月份有 31 天）。

* + 1. 声明一个适合于由一个整形的识别号码、一个姓名（最多由 20 个字符组成）、一个浮点型的工资率和一个浮点型的已工作小时数组成的雇员记录的单一结构类型。编写一个程序，交互地接收下面的数据到一个 6 个结构的数组中。

|  |  |  |  |
| --- | --- | --- | --- |
| 识别号 | 姓名 | 工资率 | 已工作的时间 |
| 3462 | Jones | 4.62 | 40.0 |
| 6793 | Robbins | 5.83 | 38.5 |
| 6985 | Smith | 5.22 | 45.5 |
| 7834 | Swain | 6.89 | 40.4 |
| 8867 | Timmins | 6.43 | 35.5 |
| 9002 | Williams | 4.75 | 42.0 |

输入了数据之后，程序输出每个雇员的识别号、姓名和总工资的工资报表。

* + 1. 声明一个适合于由一个整形汽车车牌号码、这辆汽车已行驶英里的整形值和一个每辆汽车已消耗的燃料加仑数的整形值组成的一个汽车记录的单一结构类型。编写一个程序，交互地接收下面的数据到一个 5 个结构的数组中。

|  |  |  |
| --- | --- | --- |
| 汽车车牌号 | 行驶的英里数 | 消耗的加仑数 |
| 25 | 1450 | 62 |
| 36 | 3240 | 136 |
| 44 | 1792 | 76 |
| 52 | 2360 | 105 |
| 68 | 2114 | 67 |

一旦输入了数据后，程序就创建并输出一个列出每辆汽车车牌号和这辆汽车实际的每百英里消耗的燃料加仑数的报表。

* + 1. 构建一个由整形值标识号和浮点型工资组成的结构体，并通过键盘输入数据，给 10 个结构体元素的数组赋值，然后以工资为标准通过冒泡排序对结构体数组进行从小到大排序，最后输出排序后的结果。
    2. 编写一个名称为 days()的函数，确定作为一个结构传递的任意日期自 1900 年 1 月 1 日起的天数。

并在主函数测试该函数的功能。结构体定义如下： struct Date

{

int year； int month； int day；

}；

在编写 days()函数时，每年按 360 天，每月按 30 天计算。

* + 1. 编写一个名称为 difDays()的函数，计算并返回两个日期之间相差的天数，并在主函数测试该函数的功能。日期结构体定义如下： struct Date

{

int year； int month； int day；

}；

提示：difDays()函数中可以调用第 10 题中的 days()函数两次。

* + 1. 编写一个名称为 larger()的函数，该函数返回传递给它的任意两个日期中靠后的日期。例如，如果传递日期 2001/10/9 和 2001/11/3 给 larger()，则第二个日期应该返回。
    2. 假设已经存在定义： union

{

double rate；

double taxes；

int num；

}flag；

为这个联合编写适当的 printf()函数调用，显示这个联合的各个成员。

* + 1. 定义个名称为 car 的联合变量，包含一个名称为 year 的整数、一个名称为 name 的 10 个字符的数组和一个名称为 model 的 10 个字符的数组。在主函数中初始化该联合体的变量，并输出。
    2. 使用 gets()和 fputs()函数，编写一个程序，从键盘接收几行文本，并将每一行写入一个名称为 text.dat 的文本文件，直到输入一个空行时为止（空行是指没有文本的行，即通过按回车键引起的一个新行）。
    3. 编写一个程序，实现跟 15 题相反的过程，即通过 fgets()读取 text.dat 文件中的所有内容，并通过 puts()显示在屏幕上。
    4. 创建一个包含下列数据的名称为 emloy.dat 的文本文件：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Anthony | A.J. | 10031 | 7.82 | 62/12/18 |
| Burrows | W.K. | 10067 | 9.14 | 63/6/9 |
| Fain | B.D. | 10083 | 8.79 | 59/5/18 |
| Janney | P. | 10095 | 10.57 | 62/9/28 |
| Smith | G.J. | 10105 | 8.50 | 91/12/20 |

编写一个程序，读取出上述文件中的内容到结构体数组中（自己定义对应的结构体类型），然后显示在屏幕上。

* + 1. 编写一个程序，打开一个文本文件，读取出文件的所有内容，并显示带有相关行的文件的内容。

即程序能够在文件的第一行前面输出 1，在第二行的前面输出 2，等等。 17、创建一个包含下列数据的文本文件：

H. Baker 614 Freeman St. Orange NJ

D. Rosso 83 Chambers St. Madison NJ

K. Tims 891 Ridgewood Rd. Millburn NJ

B. Williams 24 Tremont Ave. Brooklyn NY

编写一个程序，将以上数据读取到结构体数组中（自己定义对应的结构体类型），利用下列输出格式显示以上数据：

Name:

Address:

City， State:

* + 1. 用下列包含零件号、初始数量、售出数量和要求的最小库存数的数据创建一个文本文件：

|  |  |  |  |
| --- | --- | --- | --- |
| 零件号 | 初始数量 | 销售数量 | 最小库存数 |
| QA310 | 95 | 47 | 50 |
| CM145 | 320 | 162 | 200 |
| MS514 | 34 | 20 | 25 |
| EN212 | 163 | 150 | 160 |

编写一个程序，读取出以上文件中的数据到结构体数组中（自己定义对应的结构体类型），创建一个基于以上数据的库存报表，并将零件号、当前库存数量以及使库存数量达到最低标准所必须增加的数量显示在屏幕上。

* + 1. 编写一个程序保存下列数据到一个文本文件中。

5 96 87 78 93 21 4 92 82 85 87 6 72 69 85 75 81 73

同时在程序中计算并显示以上数据中每组数的平均数。这些数据在文件中的排列方式是：表示每组数目的数放在该组数的前面。例如，文件中的第一个数 5 表示接下来的 5 个数将被分组在一起，数 4 表示跟随的 4 个数是一组，而 6 表示最后的 6 个数是一组。

* + 1. 编写一个程序，读取和显示一个名称为 text.dat 文本的文件中每隔一个字符的字符。
    2. 编写一个名称为 readBytes()的函数，读取从一个文本文件中的任意位置开始的 n 个字符，并将读取到的字符保存到一个字符数组中（该函数应该接收四个参数：文件指针、第一个被读取字符的偏移量、要被读取的字符数辆和存储读到的内容的字符数组地址）。在主函数中编写程序，测试 readBytes() 函数的功能是否正确。
    3. 编写程序，把数 92.65，88.72，77.46，82.93 作为双精度值写入一个名称 results.bin 的二进制文件中。在写入数据到这个文件之后，程序再从这个文件中读取数据，计算所读取的 4 个数的平均值并显示这个平均值。
    4. 编写一个程序创建一个名称为 points.bin 的二进制文件，并将下列数据写入文件中。

6.3 8.2 18.25 24.32

4.0 4.0 10.0 -5.0

-2.0 5.0 4.0 5.0

同时在该程序中使用一个 for 循环读取 4 个数，并将每条记录中的第一个数和第二个数作为某个点的坐标，第三个数和第四个数作为第二个点的坐标，计算并显示每一对点的斜率和中点。

26、编写一个程序，创建一个 grades.bin 的二进制文件，并将下列 5 行数据写入文件。

90.3 92.7 90.3 99.8

85.3 90.5 87.3 90.8

93.2 88.4 93.8 75.6

82.4 95.6 78.2 90.0

93.5 80.2 92.9 94.4

同时在该程序中读取出每一行的四个数据（分数），计算并显示每一组的平均值（平均分）。

* + 1. 定义一个结构体节点。

struct TeleType

{

char name[30]；

char phoneNum[15]；

struct TeleType \*nextAddr；

}；

在主函数中定义 5 个结构体变量，并连接成链表，记录头节点，并单独设计一个函数来负责输出该简单链表的内容。

* + 1. 利用 malloc 动态分配内存空间方法改造第 27 题，使得可以动态通过键盘输入不同的节点数据，并将这些节点连接成链表，自己规定输入固定的数据后就可以停止链表节点的创建，同样单独设计一个函数来输出链表的内容，另外再设计一个函数来释放链表中动态申请的内存空间，并在主函数退出的时

候调用该函数。

* + 1. 设计一个函数 creatlist，用来建立一个带头节点的单链表，链表的新节点总是插入在链表的末尾。链表的头指针作为函数值返回，链表最后一个节点的 next 域放入 NULL，作为链表结束标志。同时设计一个函数用来输出链表的内容。在主函数中调用这两个函数来测试链表是否正确建立。链表节点结构体的参考定义如下。

struct node

{

char data[64]；

struct node \*next；

}

* + 1. 设计一个函数 first\_insert()，该函数的功能是在已知链表的首节点之前插入一个指定值的节点；另外设计一个函数 reverse\_copy()，该函数的功能是按已知链表复制出一个新链表，但新链表的节点顺序与源链表的节点顺序相反；再设计一个函数 print\_link()，该函数用来输出链表中各表元的值；同时设计函数 free\_link()，该函数用来释放链表全部表元空间。在主函数中合理调用以上设计好的函数来验证这些函数功能的正确性。

1. 实验结果

写实验报告。内容包括：

* + 1. 必做习题的运行结果，源程序；
    2. 选做习题的运行结果，源程序。
    3. 程序调试中出现的错误提示。（英文、中文对照）
    4. 若有没通过的程序，分析原因。

## 实验 **10** 文件

1. 实验目的
   1. 掌握文件、缓冲文件系统和文件结构体指针的概念。 2、掌握文件操作的具体步骤。
   2. 学会使用文件打开、关闭、读、写等文件操作函数。
   3. 学会用缓冲文件系统对文件进行简单的操作。
2. 必做题目
   1. 阅读下面的程序，分析其执行后，文件 filetest.txt 中的内容是什么。

#include <stdio.h> #include <string.h>

void fun(char \*fname，char \*st)

{

FILE \*myf； int i；

myf=fopen(fname，"w")； if (st==NULL)

return；

for(i=0；i<(strlen(st))；i++) fputc(st[i]，myf)；

fclose(myf)；

}

void main()

{

fun("filetest"，"new world")；

fun("filetest"，"hello!")；

}

2、以下程序中用户由键盘输入一个文件名，然后输入一串字符（用#结素输入）存放到此文件中形成文本文件，并将字符的个数写到文件尾部，请填空。

#include <stdio.h> void main()

{

FILE \*fp； char ch，fname[32]； int count=0；

printf("Input the filename:")； scanf("%s"，fname)；

if((fp=fopen( ⑴ ，"w+"))==NULL)

{

printf("Can't open file:%s\n"，fname)；

exit(0)；

}

printf("Enter data:\n")； while((ch=getchar())!='#')

{

fputc(ch，fp)；

count++；

}

fprintf( ⑵ ，"\n%d\n"，count)； fcolse(fp)；

}

1. 从键盘输入一个字符串，将其中的小写字母全部转换成大写字母，然后输出到一个磁盘文件

“test.dat”中保存，再从该文件中读出字符串并显示出来。

1. 有五个学生，每个学生有 3 门课的成绩，从键盘输入数据。（包括学号、姓名、三门课的成绩）、计算出平均成绩，将计算出的平均分存放在磁盘文件“stud.dat”中。
2. 编写程序，生成 10000 个随机整数，并将其保存到磁盘文件 test.txt 中。再从文件 test.txt 中读入这些整数到内存中，对其进行排序，将排序后的结果输出到文件 data.txt 中。

三、选做题目

1、阅读下列程序说明和 C 程序，并填空。

本程序找出正文文件“st.dat”中的所有整数。在该正文文件中各整数间以空格字符、Tab 符（制表符）、

回车符分隔。程序中用数组 b[]存储不同的整数，变量 k 为已存入数组 b 中的不同整数的个数，并假定文件中的不同整数个数不超过 1000 个。程序如下：

#include <stdio.h> #define N 1000 void main()

{

FILE ⑴ ； int b[N]，d，i，k； if ((fpt= ⑵ ==NULL){ printf("Can not open file st.dat.\n")； return； }

k=0；

while(fscanf( ⑶ )==1)

{ b[k]=d；

for(i=0；b[i]!=d；i++)；

if( ⑷ ) k++；

}

(5) ； for(i=0；i<k；i++) printf("%d"，b[i])； printf("\n")；

}

2、阅读下列程序说明和 C 程序，并填空。

本程序从存有产品名称和数量的文件读入数据，按产品的类别符（产品名的第一个英文字母）所属

分段范围，输出各大类产品的数量清单。程序约定产品类别符的大类分段范围如下：

A-E，F-I，J-M，N-Q，R-U，V-Z

程序如下，请将程序补充完整：

#include <stdio.h>

#include <process.h> #include <string.h> char class\_list[]="EIMQUZ"； int q\_list[26]，q； FILE \*fopen()， \*fp；

void main()

{

char file\_name[120]，pname[20]；

int i，sp；

char sch；

printf("Input data file name.")； scanf("%s"，file\_name)；

if((fp=fopen(file\_name，"r"))==NULL)

{

printf("Can not open file %s\n"，file\_name)；

exit⑴；

}

for(i=0；i<26；i++) q\_list[i]=0； fscanf(fp，"%20s%s"，pname，&q)； while(!feof(fp))

{

sp=⑴；

while(class\_list[sp]!='\0')

if((pname[0]>='a'&&pname[0]<='z'&&⑵)||⑶)

break；

else ⑷；

q\_list[（5）]+=q；

fscanf(fp，"%20s%d"，pname，&q)；

}

fclose(fp)；

sch='A'；

for(i=0；i<strlen(class\_list)；i++)

{

if(q\_list[i]>0) printf("%c--%c:%d\n"，sch，class\_list[i]，q\_list[i])； sch=class\_list[i]+1；

} }

四、实验结果

写实验报告。内容包括：

1. 必做习题的运行结果，源程序；
2. 选做习题的运行结果，源程序。
3. 程序调试中出现的错误提示。（英文、中文对照）
4. 若有没通过的程序，分析原因。

## 实验 **11** 综合实验

1. 实验目的
   1. 巩固掌握 C 语言的相关知识点。
   2. 能够融会贯通、灵活运用 C 语言处理实际问题。
2. 必做题目
   1. 解决硬币体系问题。从键盘输入硬币体系中不同面值的 N 和各种面值大小，输入要买商品的价格 P，输出缩用硬币的最优方案，似的硬币总面值等于 P 且用的硬币数目最小。
   2. 某个化学实验室可以用 3 套不同的仪器中任意一套去完成。在做完一个实验之后，如果下次仍用原用的那套仪器，则必须对仪器的某些部分进行清洗，这要花费一段时间。如果下次换用另一套仪器，则要吧原仪器从辅助装置上拆卸下来再装上换用的仪器，这也要花费一段时间。假定依次实验的时间比任一套仪器的清洗时间都长，那么一套仪器换下来后可以在实验过程中清洗，在下一次实验时再使用，相当于节省了清洗时间。设当 i≠j 时，tij 表示仪器 I 换成仪器 j 时所需的时间；当 i=j 时，tij 表示仪器 i 清洗所需的时间。tij 如下表示。

i j

1

2

3

1

10

9

14

2

9

12

10

3

6

5

8

现在要做 5 次实验，应如何安排使用仪器的顺序，使得在第一次开始实验之后，到最后一个实验完成之间，花费在仪器清洗和仪器更换上的总时间最少。

1. 选做题目
   1. 口令程序设计和大量数据的分析。

该题目包括口令程序的设计和大量数据的分析两个部分。

其中，实施口令程序按如下步骤进行。

* 1. 一般口令程序：程序中预设一个口令，然后由键盘输一个口令，如果键盘上输入的口令与程序中预设的口令一致，则通过口令程序进入下一个程序模块。
  2. 关闭口令屏幕回显：在上一步中，用户输入的口令是有屏幕回显的，即从屏幕上可以看到用户输入的口令，关闭口令屏幕回显，就是使用户输入的口令在屏幕看不到。
  3. 限制口令输入次数：在口令输入中可以限制输入的次数，如 3 次，3 次都不正确则退出程序。
  4. 按不同方式设计口令：如按时间的日期设口令，在时间未到指定日期没有口令，当指定的日期一到启动口令程序，需要校验口令。

思考题：

* 1. 能否设计出一个像 Windows 界面的口令程序？
  2. 能否设计出一个能随时修改口令的口令程序？

大量数据分析：数据分析分好多种，如排序、筛选出符合条件的数等，数据量大必须把数据存入磁盘，在数据分析中，按如下步骤：

⑴在磁盘预存不超过 200 个 4 位数

⑵用 C 打开此文件并把文件中的数据读入一维数组。

⑶数据分析：比如分析数据中千位与个位相等的四位数、百位加个位等于千位加十位的四位数（显示在屏幕）。

思考题：

* 1. 求出一篇英文文章中单词最长的单词。
  2. 求出不多于 200 个整型数中，超过平均数的那些数中的素数。

本题总的步骤：

⑴在磁盘存放不超过 200 个 4 位数

⑵在程序中预设 4 个口令，这四个口令作为进行不同数据分析的依据，由键盘输入一个口令，只要与其中一个口令相符（注意关闭屏幕回显，限制口令输入不⑶数据分析：第一口令用于统计并输出 4 位数中 个位＋千位＝百位＋十位 的数第二口令用于统计并输出 4 位数中 个位＜十位＜百位＜千位 的数 第三口令用于统计并输出 4 位数中全部素数，并按由小到大输出这些素数第四口令用于统计并输出 4 位数中百位、千位是偶数，十位、个位是奇数的数

* 1. 用 C 语言编写音乐软件。选择自已喜欢的一首歌曲作为程序的曲谱文件，并对该曲谱文件进行数字化，然后编写出播放该歌曲的音乐程序。

⑴音频的实现程序中要用到几个函数：

sound()函数的用法为：void sound(unsigned frequency)；功能是以打开 PC 扬声器发出指定频率的声响，直到用函数 nosound()去关闭它。参数 frequency 是一个无符号整数，代表的是发声的频率，也就是音高。一般音乐简谱上的音高与频率的对应关系如表一所示。

nosound()函数的用法为：void nosound(void)；功能是关闭 PC 扬声器。它是一个无参函数，关闭用 sound()函数打开的 PC 扬声器，禁止扬声器发声。

delay()函数的用法为：void delay(unsigned milliseconds)；功能是将程序的执行暂停一段时间。该函数的参数也是一个无符号类型的参数，代表是廷时的长度，其单位是毫秒。

sleep()函数的用法为：unsigned sleep(unsigned seconds)；功能是让程序序执行挂起一段时间，它与 delay()函数功能有点类似。它有一个无符号类型的参数，其单位是秒，例如 delay⑴就是让程序挂起一秒种，即在这一秒钟里面，程序暂停执行。

表一 音符的频率

|  |  |  |  |
| --- | --- | --- | --- |
| 音 符 | 低 音 | 中 音 | 高 音 |
| 1 | 131 | 262 | 523 |
| 2 | 147 | 294 | 587 |
| 3 | 165 | 330 | 659 |
| 4 | 175 | 350 | 698 |
| 5 | 196 | 392 | 783 |
| 6 | 220 | 440 | 898 |
| 7 | 247 | 494 | 987 |

⑵简谱文件的数字化实现

当拿到一首歌的简谱后，可以将简谱中的音高与音长分别用一个数字信息，用不同的数字来代表不同的音高与音长。

音高的对应关系如下：

1. 代表不发声也就是表示休止符
2. 02 03 04 05 06 07 代表低音的七个音节

11 12 13 14 15 16 17 代表中音的七个音节 21 22 23 24 25 26 27 代表高音的七个音节

对于音长的处理我们可以把最短的音长用 1 代表，其余的音加倍处理。经过设定这样的对应关系后我们就可以对音乐简谱进行数字化处理了。我们用两个数据来表示一个音，第一个数据用来表示音高，第二个数据用来表示音长，中间空格或[tab]或回车分隔，例如对于简谱中某个节为：3 1 7． 1，按照上面的对应关系，我们就可以在数据文件中用以下数据表示：13 1 11 1 07 2 11 2。按照这种方法把一首歌的简谱全部经过这样的处理后就得到了该歌曲数字化的数据文件，这就完成了音乐简谱的数字化处理。考虑到音长的具体值，我们可以在数据文件中对音长进行具体确定，即数据文件中第一行数据用一个数字表示最短音长的廷时长度，第二行开始为简谱的数字化数据。

四、实验结果

写实验报告。内容包括：

1. 必做习题的运行结果，源程序；
2. 选做习题的运行结果，源程序。
3. 程序调试中出现的错误提示。（英文、中文对照）
4. 若有没通过的程序，分析原因。

**4** 课程设计

《程序设计基础课程设计》是《程序设计基础》课程的延续，属于综合实践环节。《程序设计基础课程设计》是一门独立实验课程，有 1 个学分，要求学生在课内至少完成 16 学时的实验。

## **4.1** 课程设计的目的

1. 综合应用课程的知识点：灵活运用结构化程序设计方法；熟练运用 C 程序设计技术；熟练使用程序设计语言中的各种数据类型；
2. 培养学生分析问题的能力：
3. 培养学生解决问题的能力：对问题提出自己的解决方案；学会寻找解决问题的多种途径；
4. 培养认真、求实、严谨的工作作风，为后续课程打好基础；
5. 发掘、发现各类学生的特点，为不同类别、层次的学生指明今后努力的方向。

## **4.2** 课程设计流程

1. 课程设计共包括课内 16 个学时，具体时间、地点由任课教师进行安排。

将课内 16 个学时分为 4 个阶段：

* 1. 自由分组，每组同学不超过 2 人，每组设置组长 1 名，组长负责组织项目实施。
  2. 选题：根据附录的题目进行选题。各组组长将所选题目报本班学习委员，由学习委员将选题表报指导老师。
  3. 开始课程设计。
  4. 提交文档，答辩。

1. 学生完成课程设计后，由评阅教师进行统一检查，检查形式如下：
   1. 分组答辩。
   2. 检查文档。
   3. 检查运行结果。

## **4.3** 考核办法

课程设计总成绩由下面 4 部分组成

1. 源程序及运行结果：学生完成程序后，由教师检查运行结果正确或部分正确。
2. 相关文档：代码、课程设计文档（纸质，格式附后）。
3. 答辩：课程设计结束后分组答辩，由指导教师根据学生完成的题目情况进行提问，根据学生口头回答的情况予以评分，该分数计入课程设计总成绩。

**[**注意**]**：如果课程设计分组进行，则要求同组学生必须明确分工，分别完成课题的一部分任务。答辩时，学生只针对自己完成的工作进行答辩。如果工作量不够，则认为该学生总成绩为不及格。

1. 考勤情况：课程设计期间，如果学生无故缺课，则总成绩为不及格。

成绩的评定将综合上述几个方面，综合考虑各人完成工作的难度、完成的工作量、答辩情况、考勤情况，进行成绩评定。

最终成绩等级包括：优，良，中，及格，不及格。

**[**注意**]**：

* 不及格的同学将不能得到学分，需要重修。
* 以小组为单位进行申优。
* 如未能完成任何题目，该小组以不及格计算。
* 若两组源程序代码相同，两组均以不及格计算。

## **4.4** 应提交的资料

课程设计结束时，应提交相关资料存档。具体提交时间由任课教师安排。

应提交的资料包括：

1. 每个同学提交打印的课程设计报告一份（文档要求附后）。
2. 各班以班为单位刻录光盘：包括每个同学的源程序代码、课程设计报告的电子文档。

附录 **A C** 语言常用调试技巧

## **A.1 C** 语言在编程中的特点

由于 C 语言具有功能强、使用灵活方便的特点，使 C 语言得到了广泛的应用。C 语言是一种表达式语言，利用标准库函数和自己设计的函数可以完成许多功能。如果善于利用已有的函数，可以使程序设计易于模块化。一个有丰富经验的 C 语言程序设计人员可以编写出能解决相当复杂的问题、运行效率较高、占用内存少的高质量的程序。但是，C 语言的“灵活”使得人们又难以驾驭它，尤其是初学者，出了错误但找不出错误在哪里。

C 语言的编译对语法的检查不如其它的高级语言严格，所以，它需要由程序设计人员设法保证程序的正确性。调试一个 C 语言程序要比调试一个 PASCAL 或 FORTRAN 等其它高级语言程序显得困难一些。这就更需要学习者不断积累经验，不断提高程序设计和程序调试的水平。

**A.2 C** 语言编译时的常见错误

## **A.2.1** 源程序错误信息的分类

C 程序的编译程序检查源程序错误信息分为三类：

⑴灾难性错误；⑵一般性错误；⑶警告。

1．灾难性错误出现得不多，它通常是指内部的编译错误。当一个灾难性错误发生时。编译系统立即系统停止编译，必须采取适当的措施并重新启动编译系统。

2．一般错误是指程序的语法错误，例如磁盘、存储器的存取访问错误，命令行出错等，编译程序将完成现阶段的编译，然后停止。编译程序在每个阶段，如预处理、语法分析、程序优化和代码生成等，尽可能多地找出源程序中的错误。

3．警告并不影响编译的进行，警告指出一些值得怀疑的情况，而这些情况的本身作为程序的一部分又是合理的，一旦源程序中使用了与机器有关的结构，编译程序也将产生警告信息。

编译程序首先输出三种信息，然后输出源文件名及编译程序发现出错的行号，最后输出出错信息的内容。

## **A.2.2 C** 程序的常见错误

1.说明变量容易出现的错误

1. 忘记定义变量 例如：

#include<stdio.h>

void main()

｛x=5； y=10； printf(“%d\n”，x+y)；｝

* 1. 语言要求程序中所用到的每一个变量都必须先进定义。上面的程序中对变量 x 和 y 就没有进行定义。

应该在程序的开始部分进行定义：“int x， y”，这是初学者最容易出现的一个错误。

1. 标识符的大小写字母混用
   1. 语言对标识符的大小写一般是非常敏感的，除非将其对应的开关设置为不敏感。例如：#include<stdio.h>

void main()

｛int a， b， c； a=5；b=6； C=A+B； printf(“%d\n”，C)； ｝

C 语言的编译程序把 A 和 a， B 和 b， C 和 c，分别当做不同变量。它会提示出 A、B、C 变量是未定义的变量。

1. 字符和字符串的使用混淆
   1. 语言规定字符是只具有一个字符的常量或变量。而字符串不管是常量还是变量，它起码要有两个以上的字符。

例如： char sex；

sex=〃f〃；

……………

这里，字符变量sex系统只为其分配一个字节的内存空间，无法使用两个字节的空间，应将sex=〃f〃；

改变 set=’f’。

1. 指针型函数和指向函数的指针之间的混淆
   1. 语言中的指针型函数是指该函数的返回值是地址量；而指向函数的指针是指本指针只能接受函数名对它的赋值，通过对指向函数的指针的内容运算可以将程序的控制流程切换到这个函数。

例如： int (\*ptr1)()；

int \*ptr2()；

其前者说明 ptr1 是一个指向函数的指针，而后者说明 ptr2 是一个指针型函数。

2．使用运算符容易出现的错误

1. 误把“=”当做“==”运算符

在许多高级语言中，都把“=”既作为赋值运算，又把它作为运算符中的“等于”；BASIC 语言和 PASCAL 语言都是将它们混用的。

例如：

if (a=h) printf (“a 等于 b!”)；

C 语言程序规定：“=”是赋值运算符，“==”才是关系运算符中的“等于”，在 C 语言编译程序中，将（a=b）当做赋值表达式进行处理，它将 b 的值赋值给 a，然后判断 a 的值是否为 0，如果 a 的值是非零，则输出 “a 等于 b！”这个字符串；否则，将执行本语句的后继语句。而不是在 a 等于 b 时，就输出“a 等于 b!”这个字符串，否则就执行其后继语句。

1. 使用增 1 和减 1 运算符容易出现的错误
   1. 例如：

#include <stdio.h> void main( )

{

int \*ptr，a[]=｛1，3，5，7，9｝； ptr=a；

printf(＂%d＂，\*ptr++)；

}

在这个程序中，有人认为其输出是 a 数组的第一个元素 a[1]的值：3。其实是由于\*ptr++表达式中的增 1 运算是后置加 1，所以要先输出 ptr 指针所指向的内容：a[0]的内容，即是 1，然后再调整指针，使指针 ptr 指向 a[1]。如果是\*（++ptr），则先使指针 ptr 指向数组 a[1]，然中输出其值。

* 1. 例如：

#include <stdio.h> void main( )

{ int a=2； printf(＂%d:%d:%d\n＂，a，++a，a--)；

}

上面程序的结果很容易判定为“2：3：3”；其正确的结果是“2：2：2”。其原因是：在执行 printf（）

语句时，C 语言的编译程序将参数自右至左依次压入栈中。即是先压入 a—参数的值，再是++a 参数的值，最后是 a 参数的值。出栈时，弹出的顺序是 a，++a，a--；其结果是 2：2：2。

1. a>>2 操作并不能改变 a 的值例如：

#include <stdio.h> void main()

{ unsigned char a； a=0x10； while(a) printf(＂%0x 右移两位的值是%0x\n＂，a，a>>2)；

}

上面的程序是一个死循环。其原因是循环体内 a 的值并发生变化。请注意；象 a>>2 这样操作并不

会使操作数 a 的值发生变化。在作 a=a>>2 操作时，变量 a 的值才能发生变化。

1. 错把&运算符当做&&运算符例如：

#include <stdio.h> void main()

{

int a=5，b=7，c=9，d=11；

if ((a>b)&(c>d)) printf(＂a>b 并且 c>d!\n＂)；

else printf(＂a 不大于 b 而且 c 也不大于 d!\n＂)；

}

本例中，将位逻辑运算符&当做了逻辑运算符&&。此种错误是 C 语言初学者最容易犯的错误。

3．使用 I/O 函数容易出现的错误

⑴ 输入出数据的类型与所用的格式控制符不一致

例如：

#include <stdio.h> void main()

{ int a； float b； a=3； b=4.5； printf(＂%f %d\n＂，a，b)；

}

上述程序在编译时并不给出错误信息，但是运行的结果确是有点令人不可理解：

0.000000 16403

它们在存储时是按照赋值转换的规则进行转换的，而输出时是将数据在存储单元中的形式按输出格

式控制符的要求将其输出。

1. 忘记使用地址运算符 例如：

scanf (＂%d %d＂，x， y)；

这是许多 C 语言初学者容易犯的错误，更是其它语言中的习惯所造成的习惯性错误。C 语言要求使用某些变量时，应该指明其地址标记。其方法是将上述语句改为： scanf (＂%d %d＂， &x， &y)；

1. 输入数据与要求不符

使用 scanf ()语句时，应该特别注意输入数据与语句要求数据的一致性。

例如：

scanf (＂%d， %d＂， &x， &y )；

如果您输入数据时： 6 7〈CR〉，这是错误的；您应该回答： 6，7〈CR〉。

4．使用函数容易出现的错误

⑴ 未对被调用的函数进行必要的说明例如：

# include〈stdio.h〉 void main()

{

float x， y， z ；

x=3.5； y=-7.5； z=max (z，y )； printf (＂较大的数是 %f\n＂，z)；

｝

float max (float x， float y)

{ return (x>y ? x : y )； }

这个程序看起来并没有错误，但在编译时将给出出错信息。其原因是：max（）函数是个浮点型函数，并且是在 main（）函数之后定义，而调用本函数时并未对其进行说明，所以出现错误。其改正的方法有两种：

其一是：在函数调用之前用“float max ( )”进行说明，其位置最好是在主函数的变量定义和说明部分进行。

其二是：将 max（）函数的定义移动到 main（）函数的前边。

1. 将函数的形式参数及其局部变量一起说明例如：

min（）int x， y， z ；

｛z=x<y ? x:y； return (z)； ｝

这里的错误是将函数的形式参数和函数的局部变量混为一谈了。函数的形式参数具有局部的存储属性，也需要对其进行说明，其位置应该在函数名和函数的主体之间进行。而函数内部的局部变量应该在函数体的首部，变量定义和变量函数的说明部分进行定义，本例应该将函数的局部变量 z，单独在函数体内进行定义。

1. 认为函数的形式参数可以影响调用函数的实在参数例如：

#include <stdio. h> void main ()

{

int x， y； x=5； y=9；swap(x， y)；

printf(＂%d，%d\n＂，x ，y)；

}

swap (int x， int y)

{

int t ；

t=x； x=y； y=t；

｝

其本意是想通过调用 swap（）函数来使得 main（）函数中变量 x 和 y 的值能得到交换，但结果是未能达到预期的效果，其原因是：这种传值方法的函数调用，实在参数和形式参数是分别不同的独立单元，每一组单元的操作并不能影响另一组单元。要想达到上述目的，只有采用传址方式，即：

# include <stdio. h> void main()

{

int x， y ；

x=5； y=9； swap(&x，\&y)；

} swap ( int \*x， int \*y)

{ int t ；

t=\* x ； \*x=y； \*y=t；

}

⑷ 实在参数和形式参数类型的不一致

例如：

#include <stdio. h> void main ( )

{

int a=4， b=9， c； c=fun (a， b)；

……

}

fun (float x， float y)

｛… …｝

上面程序的实在参数 a 和 b 是整型变量，而形式参数 x 和 y 却是浮点型变量。C 语言要求实在参数

和形式参数的数据类型必须一致。

1. 函数参数的求值顺序造成的差异例如：

printf (＂%d， %d， %d \n＂， i， ++i， ++i )；

如果变量 i 在此前的值是 3，人们一般认为其输出 3，4，5。其实并不一定。在有些计算机系统中输出确是：5，5，4。其原因是有些系统采用的是自右至左的顺序求函数参数的值，即输出为 5，5，4。而有的系统则是从左至右求函数参数的值，故其输出是 3，4，5。

1. 用动态的地址作为函数的返值例如：

#include <stdio.h>

char \*strcut(char \*s，int m，int n)；

void main()

{ static char s[]="Good Morning! "； char \*ptr； ptr=strcut(s，3，4) ；

printf(＂%s\n＂，ptr)；

}

char \*strcut (char \*s， int m， int n)

{ char substr[20]；int i； for(i=o；i<n；i++) substr [i]=s[m+i-1]； substr [i]= ′\0′； return (substr)；

｝

函数 strcut（）是个指针型函数，它的返值是个地址量——自动型字符数组 substr[ ]的地址。由于字符数组 substr[ ]的地址在函数 strcut（）函数返回到主调函数时，其地址空间已被释放了，所以指针 ptr 所指的地址就是一个不定的，对应字节符串的输出也就是莫名其妙的，严重时可能使系统瘫痪。其解决的方法是，把字符数组定义为静态型： static char substr [20]；

⑺ 对指向函数的指针赋值有错误例如：

#include <stdio.h> char pl()； void main()

{ char (\*sl)()； char ch； sl=p1()； ch=(\*sl)(＂abcde＂)；

printf(＂%c\n＂，ch)；

}

char pl (char \*s2)

{ return (s2 [1])； }

本例中的指针 sl 是一个指向函数的指针，它所接收的是某函数的地址。而函数的地址就是函数名的本身，本例中就是 pl，而不是 pl（）。在此例中的“sl=pl（）；”是将函数 pl（）的返值赋值给一个指函数的指针 sl，这是绝对不允许的。

⑻ 函数的返值与期望的不一致例如：

# include<stdio. h > void main ( )

{ float x， y； scanf (＂%f%f＂，&x， &y)； printf (＂%d\n＂，addup(x，y))；

｝

float addup(float x，float y)

{ return(x+y)； }

本例中的主函数期望从函数 addup（）的返值得到一个整型数，而函数 addup（）返回的确是一个浮

点数。

5．使用数组容易出现的错误

1. 引用数组元素使用了圆括号

例如：

……

int i， a (10 )；

for (i=0； i <10； i++) printf(＂%d＂，a(i))；请记住：C 语言的数组的定义和数组元素的引用都必须使用方括号。

1. 引用数组元素越界例如：

#include<stdio.h> void main()

{

static int a[10]=｛10，9，8，7，6，5，4，3，2，1｝； int i； for (i=0；i<10；i++)

printf (＂%d＂，a[i])；

}

上面程序的错误在于：C 语言规定，数组的下标是从零开始到 N-1；而本例在定义时定义了 a[0]~a[9] 十个元素，引用时却使用了 a[0]~a[10]共十一个元素。学过 BASIC 语言的人最容易犯这类错误，其原因是 BASIC 语言数组的定义是不包括第 0 个元素在内的元素的个数。

1. 对二维或多维数组定义和引用的错误例如：

void main ()

｛ int a[5，9]；

……

printf(＂%d＂，a[3，5])；

……

｝

这种错误也是学过 BASIC 语言的人最容易犯的错误。记住：C 语言中的数组的维数是由方括号对指

定的，即：a[5][9]，而不是使用逗号分隔符来指明数组的维数。

1. 数组名只不过代表数组的首地址例如：

#include <stdio.h> void main()

{

int a[]=｛5，4，3，2，1｝；

printf(＂%d%d%d%d\n＂，a)；

}

我们需要记住，数组名它是地址常量，它仅代表被分配的数组空间的首地址，并不能代表数组的全

体元素。

1. 向地址常量——数组名赋值例如：

#include <stdio.h> void main()

{ char str[20]；

str=＂Turbo C＂；

printf(＂%s\n＂，str)；

}

上面的错误是将数组和指针的特性混淆造成的。数组名是地址常量，指针是地址变量。数组在定义的同时可以对其初始化，有的 C 语言还规定其数组应该是表态或外部数组；而指针可以在其定义的后面再对其赋值，而且是将字符串的首地址赋给指针变量。

1. 数组初始化越界例如：

#include <stdio.h>

void main()

｛

char str [6]="Out of!"；

printf(＂%s\n＂，str)；

｝

上面例子有两处错误：其一是对自动型字符数组进行初始化操作，但新版本的 Turbo C 是允许对自动数组进行初始化；其二是定义数组时所开辟的空间不足，就是给数组 str[ ]所开辟的存储空间只有六个字节，而对其初始化操作有七个字符，系统将使“f !”丢失。解决这种错误的好办法是使用不定长数组进行初始化：static char str[ ] ="Out of!"；。

6．使用指针容易出现的错误

⑴ 不同类型的指针混用例如：

#include <stdio.h> void main()

{

float a=3.1， \*ptr； int i=3，\*ptr1 ； ptr=&a； ptrl=&i； ptr=ptr1；

printrf(＂%d，%d\n，\*ptr1，\*ptr)；

}

这个例子的错误在于使指向浮点数的指针也指向一个整型数。解决的方法是采用强制转换的方法： ptr=(float \*)ptrl；

1. 混淆了数组名和指针的区别例如：

# include<stdio.h>

void main()

{

int a[10]，i；

for (i=0；<10；i++)

scanf (＂%d＂，a++)；

……

}

上面的例子是把数组名当成指针来使用了。由于数组名是地址常量，它不能做增 1 运算；应该直接使用数组元素的引用的方法——a[i]。

1. 使用指向不定的指针例如：

#include <stdio.h>

void main()

{

char \*ptr； scanf (＂%s＂，ptr) ； printf (＂%s＂，ptr)；

}

语句 char \*ptr；只定义 ptr 是一个指向字符的指针，并没有给指针赋予一个指向确定的地址空间，使

用指向不确定的指针，在编译时会出现：

NULL pointer assignment 警告信息。但是，一般使用也不会使系统瘫痪，但不提倡这样使用。

1. 用自动型的变量去初始化一个静态型的指针例如：

#include <stdio.h> void main()

{

int s=100； static int \*ptr=&s；

printf (＂%d＂，\*ptr)；

}

上面例子的错误在于用一个自动型的变量的地址去初始化一个静态型的指针。虽然本例中不会造成严重的错误，但是在非主函数中这样使用时，函数结束了，自动变量的地址空间已经释放了，而静态型的指针却还指向一个早已不知是做什么用的单元，这是绝对不允许使用的。

1. 给指针赋值的数据类型不匹配例如：

#include <stdio.h> #include <alloc.h> void main()

{ char \*ptr； ptr=malloc(10)；

gets(ptr)； printf(＂%s\n＂，ptr)； free(ptr)；

}

本例中的 ptr=malloc(10)语句在编译时并不会出现错误，但是在基本概念上不清楚。函数 malloc(10)

调用返回的是无符号的整数，而指向字符的指针 ptr 与被赋值的类型不匹配，可以把语句 ptr=malloc(10)；改为： ptr=(char \*)malloc(10)；。

这样做虽然解决了类型匹配的问题，但是没有对函数 melloc()是否已分配到了足够的内存空间进行检查，一个较为完整的程序应该是：

#include <stdio.h> #include <alloc.h> void main()

{

char \*ptr=NULL； ptr=(char \*)malloc(10)； if(ptr==NULL){

printf (＂\7 内存空间不够用！\n＂)； exit ⑴：

｝

gets(ptr)； printf (＂%s\n＂，ptr)； free（ptr）；

}

⑹ 错误地理解两个指针相减的含义例如：

#include <stdio.h> #include <alloc.h> void main()

{

int i，\*ptrl， \*ptr2； if((ptrl=(int )malloc(10 \*sizeof(int)))==NULL)

{printf (＂\7 申请分配内存不成功！\n＂)；exit ⑴；} ptr2=ptr1；

for (i=0；i<10； i++)\*ptr1++=i；

printf (＂两个指针之间的元素个数是：%d\n＂，

(--ptr1-ptr2+1)/sizeof (int))；

}

上面程序的目的是求两个指针之间的元素的个数，但是却错误理解了指针相减的含义。两个指针相减：ptr1-ptr2 之差就是这两个指针指向地址之间的数据的个数，并不是地址的差值。其 printf()语句应改为：

printf（＂两个指针之间的元素个数是：%d\n＂，ptr1-ptr2+1）；。

7．其它常见的错误

1. 数值超过了数据可能表示的范围 例如：

int number； number=32769； print (＂%d＂，number)；

在一般的微型计算机上使用的 C 编译程序，对一个整型数据规定为两个字节，那么其数的表示范围就是从：-32768 到 32767；所以变量 number 所赋的值超过了数据的表示范围了。

如果将变量 number 定义为：long int number；，还必须将 printf()语句中的输出格式控制符改为长整型的，即：

printf(＂%1d)，number；才会不出错误。

1. 语句后面忘记加分号

C 语言规定，语句是以分号作为结束符或分隔符。而某一个语句遗漏了分号，在编译时指出错误的

地址往往是在其后面。如果编译时指出的错误行没有错误，应该检查一下其前面的语句是否遗漏了分号。

学过 PASCAL 语言的人往往在复合语句的最后一个语句不写分号。

例如：｛ t=a； a=b； b=t ｝

在 PASCAL 语言中，分号是两个语句之间的分隔，而不是某个语句的必要的组成部分；在 C 语言中，分号是语句的必要的组成部分，没有分号就不是一个语句。

1. 应该使用复合语句的地方而将其大括号对遗漏例如：其一是：

sum=0； i=1； while(i<100) sum+=i； i++；

本程序段的本意是实现 1+2+3+……+100。但是由于应该使用复合语句的地方遗漏了大括号对，它会不终止地循环下去。本题的改进方法有两种：其一是：

sum=0； i=1； whlie(i<=100)｛sum+=i； i++； ｝其二是：

sum=0； i=1； while(i<=100) sum+=i++；

1. 在不需要分号的地方加了分号

例如：

for(i=0； i<=100； i++)； scanf (＂%d＂a[i])；

在本例程序段中，其本意是用 scanf()语句输入 100 个数据给数组 a[i]的各个元素，由于在 for()语句

的后面多加了一个分号“；”，使得本程序段完不成上述的功能。

1. 括号不配对例如：

while((ch=getchar()!=#) putchar(ch)；

当一个语句有多层括号时，录入程序时往往会有遗漏的时候，这时你应该仔细地检查程序，以便于改正。

1. 混淆结构体类型和结构体变量例如：

struct student

｛ long num； char name[20]； char sex； int age； ｝ student.num=123456； strcpy(student.name， ＂Li lei＂)； student. sex='f '； student. age=20；

上面的错误在于：把一个说明结构体的类型和结构体变量的定义混为一谈。结构体类型是一个空洞的模板，它不被分配内存空间。而定义结构体变量才有确定的内存空间，才可以对其进行赋值。改正的方法是定义一个结构体变量，然后再对其进行赋值。

1. 在 switch()语句的成分子句中漏写了 break 语句

例如：

switch(score)

{ case 5:printf(＂成绩优秀！！！＂)； case 4:printf(＂成绩良好！！＂)； case 3:printf(＂成绩及格！＂)； case 2:printf(＂成绩不及格。＂)； default :printf(＂数据输入有错误＂)

}；

上述程序段在某个学生的成绩是优秀时，却会输出：

“成绩优秀！！！成绩良好！！成绩及格！成绩不及格。数据输入有误”，其原因是在其成分子句的结束部分漏掉了 break 语句造成的。其改正的程序段如下： switch(score)

{ case 5:printf(＂成绩优秀！！！＂)；break； case 4:printf(＂成绩良好！！＂)；break； case 3:printf(＂成绩及格！＂)；break； case 2:printf(＂成绩不及格。＂)；break；

default :printf(＂数据输入有错误＂)；

}

这种错误也是学过 PASCAL 语言的人最容易犯的错误。

⑻ 文件操作的不一致文件操作的第一步是要先打开一个文件，而打开文件时有一个打开模式，初学者往往是在此操作常

有不一致的毛病。

例如：

if((fpr=fopen(＂test＂， ＂r＂)==NULL)

{ printf(＂\7 文件 test 打不开!=\n＂)； exit⑴；}

ch=fgetc(fpr)； while(ch!=’#’) ｛ch+=4； fputc(ch， fpr)； ch=fgetc(fpr)；

}

在上述程序段中，打开文件是以“r”方式——只读方式打开的，而在文件操作却既要进行读操作又要写操作，显然是不允许的。也就是你应该注意打开模式与文件操作一致。除此之外，有的人在程序中打开文件而不注意随时关闭文件，这样做有两个不良的结果，表现为：

⑴随时打开文件而不注意关闭文件有时会造成文件的不够用。虽然 C 系统可以提供 20 个文件供你使用，除了五个系统标准文件之外，你可以使用的只有十五个，多次打开而不随时关闭暂时不使用的文件就会造成文件不够用。

⑵有时系统会自动地关闭一些文件，但可能会造成数据的丢失。因此，必须注意随时将暂时不用的文件关闭。

初学者容易犯错误，大多数是对 C 语言的语法不熟悉造成的。使用 C 语言编程多了，熟练了，出现的错误也就会明显地减少。

程序出现错误可以分为两种情况：

1. 语法错误造成的。这些错误违背了 C 语言的语法规定，编译程序一般能够给出“出错信息”，并指出错误的所在行，只要细心地检查，可以很快地在其指定的行或在该行之前找到错误，并进行改正。
2. 非语法错误造成的。这类错误并无明显地违背 C 语言的语法规则，但是程序的执行结果与预期的结果完全不同，这类错误往往是程序的设计人员给予计算机的指令与编程者的愿意不符，而计算机检查不出语法，错误只能根据“死板”的规定走执行。这类错误是最伤脑筋的，需要采取多种方法才能排除。

附录 **B** 课程设计报告文档格式



重 庆 理 工 大 学课程设计

课程 程序设计基础

题目 综合程序设计

## 院系名称 计算机科学与工程学院

班 级 学生姓名 学号 指导教师 评阅教师

时 间

1. 问题描述

编写程序，实现小学生四则运算练习项目。

1. 需求分析

### **2.1** 功能需求

1. 用户可以从菜单中选择某种运算进行练习。具体包括：加法，减法，乘法，除法；
2. 用户可以指定每次练习的题目数量、设置练习的总分；
3. 每小题练习后给出结果正确与否的提示；一次练习结束后，给出用户所得的总分；
4. 用户选择退出时，可退出系统。

### **2.2** 性能需求

1. 系统设计合理，兼顾系统运行速度和系统资源消耗两方面的需求。
2. 系统运行稳定，具有健壮性：对用户的非法操作能给予相关处理或提示；避免随意终止、退出程序。
3. 由于用户是小学生，系统界面应简单、美观。
4. 系统操作简便，具有良好的交互性（有准确的提示性信息）。

**3** 系统设计

# **3.1** 系统功能模块图

系统功能模块图如图1所示：

小学生数学题系统

选

择

模

块

加

法

模

块

减

法

模

块

乘

法

模

块

除

法

模

块

菜

单

模

块

退

出

模

块

打

印

分

数

模

块

图1 系统功能模块图

### **3.2** 系统功能设计

各模块的功能及输入输出如下。

⑴【菜单模块】

【模块名】：showMenu

【功能】：程序启动时，出现菜单界面。该界面将各功能操作显示成列表，供用户选择。

【输入】：无 【返回值】：无

⑵【选择模块】

【模块名】：getChoice

【功能】：用户输入功能模块的序号，根据用户输入的选项，调用相应的功能模块。

【输入】：无 【返回值】：无

⑶【加法模块】

【模块名】：doAdd

【功能】：用户输入题目数、总分数；随机生成试题。根据用户的答题情况给出用户每题的答题情况，以及总分。

【输入】：无 【返回值】：答题总分

【模块内部逻辑】：随机产生两个整数；

1. 显示题目
2. 接收用户输入的答案
3. 判断正确与否，给出提示信息，并统计总分；

⑷【减法模块】

【模块名】：doSub

【功能】：用户输入题目数、总分数；随机生成试题。根据用户的答题情况给出用户每题的答题情况，以及总分。

【输入】：无 【返回值】：答题总分

【模块内部逻辑】：同⑶

【关键点】：被减数应大于减数

⑸【乘法模块】

【模块名】：doMul

【功能】：用户输入题目数、总分数；随机生成试题。根据用户的答题情况给出用户每题的答题情况，以及总分。

【输入】：无 【返回值】：答题总分

【模块内部逻辑】：同⑶

⑹【除法模块】

【模块名】：doDiv

【功能】：用户输入题目数、总分数；随机生成试题。根据用户的答题情况给出用户每题的答题情况，以及总分。

【输入】：无 【返回值】：答题总分

【模块内部逻辑】：同⑶

【关键点】： 分母不能为0；被除数应该是除数的倍数。

⑺【打印分数模块】

【模块名】：prnScore()

【功能】：输出一次测试的总分。

【输入】：答题总分 【返回值】：无

⑻【退出模块】

【模块名】：sysExit

【功能】：退出时，给出“再见”的提示信息。

【输入】：无 【返回值】：无

### **3.3** 接口及流程设计

根据各模块的功能，确定各模块及接口设计如下： void showMenu()；

int gerChoice()； int doAdd()； int doSub()； int doMul()； int doDiv()；

void prnScore(int score)；

void sysExit()；

系统整体工作流程如图 2 所示：

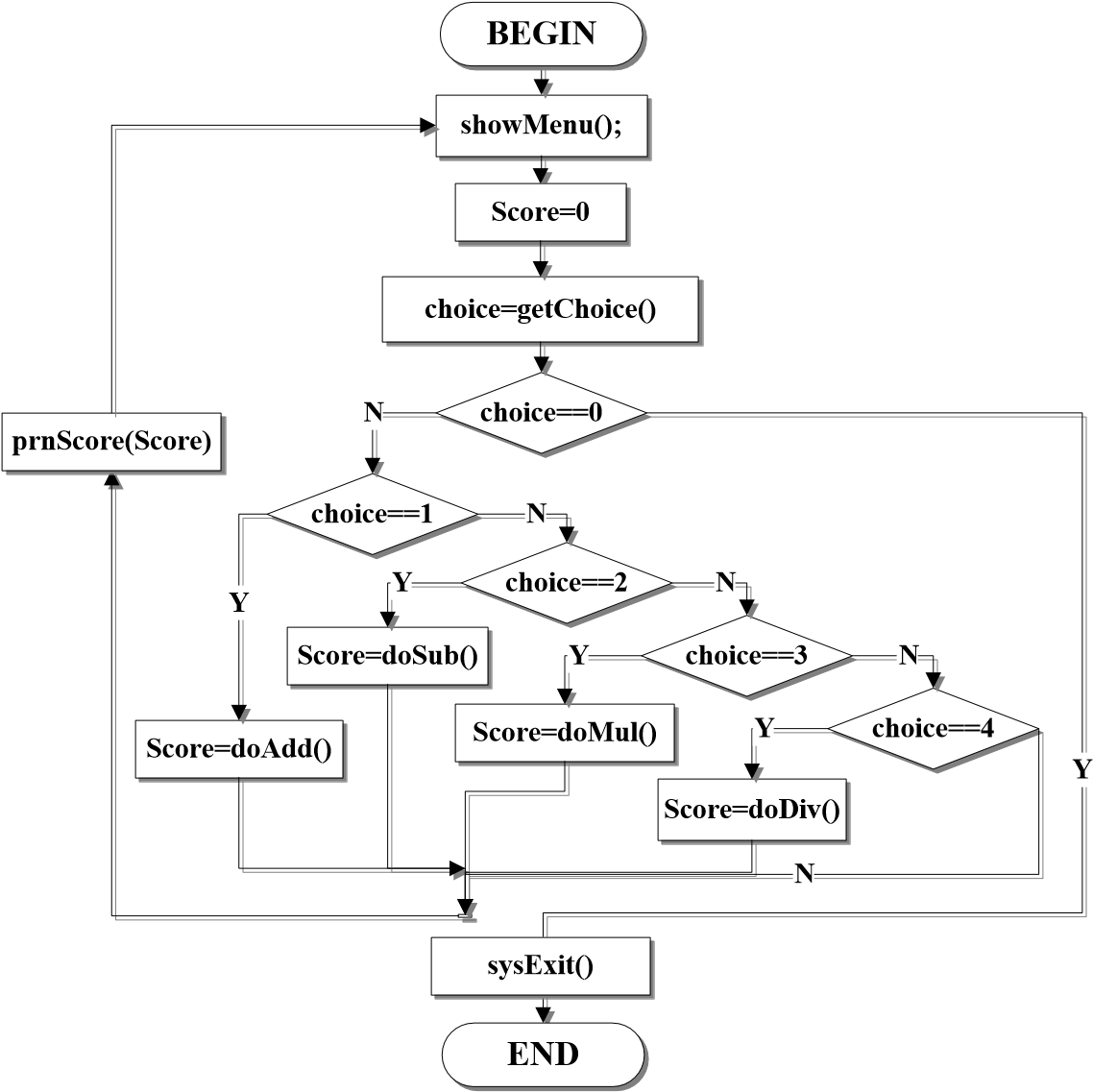


图2 系统整体工作流程图

1. 系统测试

**4.1** 菜单模块

系统菜单模块的运行结果如图 3 所示：

### **4.2** 选择模块

【输入数据】1

【结果】运行结果如图 4 所示，正常进入到【加法】计算模块

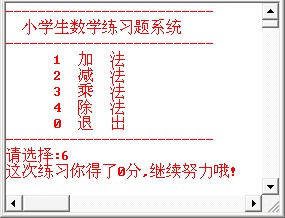
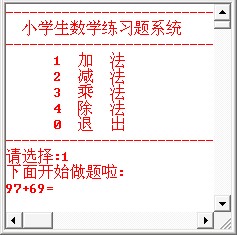
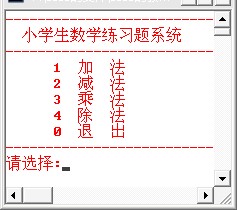
【分析】选择模块运行正常

【输入数据】6（非法数据）

【结果】运行结果如图 5 所示，提示分数为 0 分

【分析】选择模块对非法数据的判定不完整

图3 系统主菜单 图4 输入合法数据的运行结果 图5输入非法数据的运行结果



### **4.3** 计算及分数输出模块

【加法模块】运行结果如图 6 所示

【分析】加法模块运行正常，实现了分数统计。分数输出模块正常运行。

【除法模块】运行结果如图 7 所示

【分析】除法模块运行正常，实现了分数统计。

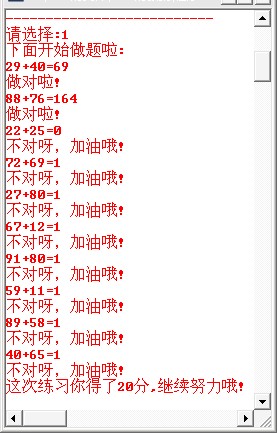
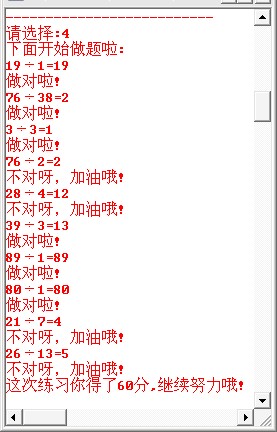
 

图6 加法计算的运行结果 图7除法计算的运行结果

### **4.4** 退出模块

【退出模块】运行结果如图 8 所示

【分析】退出模块正常运行。

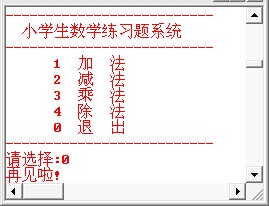


图8 计算及分数输出模块

**5** 总结

### **5.1** 工作总结

本次课程设计进行了小学生数学题系统的设计及实现工作。

首先通过对系统功能的分析，进行了系统各模块的划分；

接着对模块的功能、模块之间的接口，以及系统工作流程进行了详细设计；然后采用 C 语言进行程序编码，实现了小学生数学题系统的全部功能；最后通过选择有针对性的测试数据，对系统进行了充分的测试。测试结果表明，系统的主要功能模块运行正常。实现了系统预期的目标。

**5.2** 心得体会

### 附录源程序代码

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

//小学生数学题管理系统v1.0 作者:cqutLL

//2014年1月1日

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

#include <stdio.h>

#include <stdlib.h>

#include <time.h>

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

#define numQuestion 10

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* void showMain()； int getChoice()； int doAdd()； int doSub()； int doMul()； int doDiv()； void prnScore(int)； void sysExit()；

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

//退出模块

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* void sysExit()

{

printf("再见啦!\n")；

}

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

//主菜单模块

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* void showMain()

{

printf("--------------------------\n")；

printf(" 小学生数学练习题系统\n")；

printf("--------------------------\n")； printf(" 1 加 法\n")； printf(" 2 减 法\n")； printf(" 3 乘 法\n")； printf(" 4 除 法\n")； printf(" 0 退 出\n")； printf("--------------------------\n")； printf("请选择:")；

}

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

//选择模块

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* int getChoice()

{

int select； scanf("%d"，&select)；

return select；

}

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

//加法模块

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* int doAdd()

{

int x，y，result，score=0； printf("下面开始做题啦:\n")； srand((unsigned)time(NULL))；

int k=0；

while(k<numQuestion)

{ x=rand()%100； y=rand()%100； printf("%d+%d="，x，y)； scanf("%d"，&result)； if(result==x+y)

{

printf("做对啦!\n")；

score+=10；

}

else

{

printf("不对呀，加油哦!\n")；

}

k++；

}

return(score)；

}

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

//减法模块

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* int doSub()

{

int x，y，result，score=0； printf("下面开始做题啦:\n")； srand((unsigned)time(NULL))； int k=0；

while(k<numQuestion)

{

x=rand()%100； y=rand()%100； if(x>y)

{

printf("%d-%d="，x，y)；

}

else

{

printf("%d-%d="，y，x)；

}

scanf("%d"，&result)； if(result==x-y)

{

printf("做对啦!\n")； score+=10；

}

else

{

printf("不对呀，加油哦!\n")；

}

k++；

}

return(score)；

}

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

//乘法模块

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* int doMul()

{

int x，y，result，score=0； printf("下面开始做题啦:\n")； srand((unsigned)time(NULL))； int k=0；

while(k<numQuestion)

{

x=rand()%100； y=rand()%100； printf("%d×%d="，x，y)； scanf("%d"，&result)；

if(result==x\*y)

{

printf("做对啦!\n")； score+=10；

} else {

printf("不对呀，加油哦!\n")；

} k++； } return(score)；

}

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

//除法模块

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* int doDiv()

{

int x，y，result，score=0； printf("下面开始做题啦:\n")； srand((unsigned)time(NULL))； int k=0；

while(k<numQuestion)

{

x=rand()%100； y=rand()%100； if(x<y)

{

continue；

}

else if(y==0)

{

continue；

}

else if(x%y!=0)

{

continue；

}

printf("%d÷%d="，x，y)； scanf("%d"，&result)； if(result==x/y)

{

printf("做对啦!\n")； score+=10；

}

else

{

printf("不对呀，加油哦!\n")；

}

k++；

}

return(score)；

}

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

//打印分数模块

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* void prnScore(int score)

{

printf("这次练习你得了%d分，"，score)；

if(score>=80)

{

printf("非常不错哦!\n\n")；

}

else

{

printf("继续努力哦!\n\n")；

}

}

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

//主函数

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* void main()

{ int choice=0； int score=0； while⑴

{ showMain()； score=0； choice=getChoice()； switch(choice) { case 1: score=doAdd()；break； case 2: score=doSub()；break； case 3: score=doMul()；break； case 4: score=doDiv()；break；

}

if(choice==0)

{

break；

}

prnScore(score)；

}

sysExit()；

}