

C 程序设计

课程设计

刘振安 刘燕君 单继龙 编著



第2版



机械工业出版社
China Machine Press

高等院校计算机课程设计指导丛书

C程序设计

课程设计

刘振安 刘燕君 单继龙 编著



机械工业出版社
China Machine Press

本书独立于具体的 C 语言教科书,重点放在 C 语言结构化设计的基本特征上,涵盖 C 语言的重要基础知识。书中通过详细的实例,循序渐进地启发学生完成课程设计,培养学生实际分析问题、编程和动手的能力。本课程设计在注重基础知识训练的同时,也注重技能训练。

本书适合作为高等院校相关专业的 C 语言课程设计指导用书。

版权所有,侵权必究

本书法律顾问 北京市展达律师事务所

图书在版编目(CIP)数据

C 程序设计课程设计 第 2 版 / 刘振安,刘燕君,单继龙编著. —北京:机械工业出版社, 2009.11

(高等院校计算机课程设计指导丛书)

ISBN 978-7-111-28541-0

I. C… II. ①刘… ②刘… ③单… III. C 语言—程序设计—高等学校—教学参考资料 IV. TP312

中国版本图书馆 CIP 数据核字(2009)第 187325 号

机械工业出版社(北京市西城区百万庄大街 22 号 邮政编码 100037)

责任编辑:迟振春

三河市明辉印装有限公司印刷

2010 年 1 月第 2 版第 1 次印刷

184mm×260mm • 12 印张

标准书号:ISBN 978-7-111-28541-0

定价:24.00 元

凡购本书,如有缺页、倒页、脱页,由本社发行部调换

客服热线:(010) 88378991; 88361066

购书热线:(010) 68326294; 88376949; 68995259

投稿热线:(010) 88379604

读者信箱:hzjsj@hzbook.com

高等院校 计算机课程设计指导丛书

专家指导委员会

(以姓氏拼音为序)

陈向群	(北京大学)
陈 鸣	(解放军理工大学)
戴 葵	(国防科技大学)
何钦铭	(浙江大学)
廖明宏	(哈尔滨工业大学)
林 闯	(清华大学)
刘振安	(中国科技大学)
马殿富	(北京航空航天大学)
齐 勇	(西安交通大学)
宋方敏	(南京大学)
汤 庸	(中山大学)
王立福	(北京大学)
吴功宜	(南开大学)
赵一鸣	(复旦大学)

联络人 温莉芳

前 言

语言课程应注重边学边练，但由于课堂教学和实验的深度和广度有限，练习的深度也受到一定限制。为了弥补这一点，特设计了本课程设计。它比教学实验复杂一些，涉及的深度广些并更加接近实用。目的是通过课程设计的综合训练，培养学生实际分析问题、编程和动手的能力，最终目标是想通过这种形式，帮助学生系统掌握该门课程的主要内容，更好地完成教学任务。

本课程设计的主要特点如下：

(1) 独立于具体的 C 语言教科书，重点放在 C 语言结构化设计的基本特征上，以“不变”应“万变”，涵盖 C 语言的重要基础知识。

(2) 结合实际应用的要求，使课程设计既覆盖知识点，又接近工程实际需要。通过激发学习兴趣，调动学生主动学习的积极性，并引导他们根据实际编程要求，训练自己实际分析问题的能力 & 编程能力，并养成良好的编程习惯。

(3) 通过详细的实例，循序渐进地启发学生完成设计。课程设计将要求、算法和源程序分开，为学生创造独立思考的条件。学生在充分理解要求和算法的前提下，完全可以不按书中提供的参考程序，而设计自己的应用程序。

(4) 课程设计分为基本部分与技能提高训练，以满足不同学校和不同学生的要求。

(5) 提供一个综合课程设计，这个课程设计的重点放在强化 C 语言结构化设计的知识和进一步锻炼学生的动手能力上。

另外，在实际编程中，为了提高编程质量，对空行、空格和注释均有要求。本书也尽可能地根据实际编程要求给出空行、空格和注释，有时因为标题和页码等实际原因，也会适当减少空行、空格和注释，但希望学生在书写代码时，还是严格要求处理，以便建立良好的编程风格。

全书共分 13 章。第 0 章是概述；第 1 章是设计菜单；第 2 章是计算机解题基础；第 3 章是使用数组和指针统计成绩；第 4 章是典型算法和简单的文件编程结构；第 5 章是多文件编程；第 6 章是函数设计；第 7 章是用结构数组设计职工文件；第 8 章是设计出圈游戏；第 9 章是设计链表；第 10 章是使用循环链表求解约瑟夫环游戏；第 11 章是链表文件信息管理；第 12 章是综合课程设计。

本书第 1 版曾被全国许多高等院校采用，有的学校还将其用作毕业设计的参考资料，第 2 版的编写也得到他们的支持和帮助，在此表示感谢，希望今后继续不吝赐教。

刘振安

中国科学技术大学

2009 年 10 月 1 日

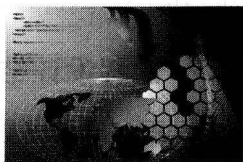
目 录

专家指导委员会	2.4 评价标准	24
前言	2.5 参考题目	24
第 0 章 概述	第 3 章 使用数组和指针统计成绩 ..	25
0.1 课程设计目标	3.1 设计题目	25
0.2 课程设计结构	3.2 设计要求	25
0.3 评价标准	3.3 参考程序	26
第 1 章 设计菜单	3.3.1 使用数组实现	26
1.1 设计一个菜单程序	3.3.2 使用分配的内存实现	28
1.1.1 设计要求	3.3.3 将数组内容全部存入内存	30
1.1.2 设计实例	3.4 评价标准	34
1.1.3 程序清单	第 4 章 典型算法和简单的文件	
1.2 增加菜单项的处理功能	编程结构	35
1.2.1 菜单项设计要求	4.1 迭代算法求 $\sin x$ 的值	35
1.2.2 建立控制台文件	4.2 递推算法	36
1.2.3 源程序清单	4.3 递归算法	38
1.2.4 运行结果	4.3.1 递归与递推的比较	38
1.3 评价标准	4.3.2 递归程序的单文件结构	39
1.4 练习使用编程环境	4.3.3 一个头文件和一个源文件的	
1.4.1 Visual C++ 6.0 简介	程序结构	40
1.4.2 VC 主窗口和工具栏	4.3.4 图解递归执行过程	41
1.4.3 如何建立控制台应用程序	4.4 查找算法	42
1.4.4 一个简单的示例程序	4.4.1 线性查找	42
第 2 章 计算机解题基础	4.4.2 二分查找	43
2.1 寻找成绩最佳者	4.5 冒泡排序	44
2.1.1 逻辑思维的计算机表示	4.5.1 图解排序过程	44
2.1.2 使用枚举法解题的思路	4.5.2 算法分析	45
2.2 寻找肇事车辆	4.5.3 算法设计	45
2.2.1 计算机解题的一般步骤	4.5.4 参考程序	45
2.2.2 课程设计内容	4.6 鸡兔同笼	46
2.3 计算机解题小结	4.7 求解百鸡问题	47
	4.8 评价标准	49

4.9 参考题目	49	8.1.1 设计思想	77
第 5 章 多文件编程	51	8.1.2 参考程序	78
5.1 设计要求	51	8.2 使用二维字符串数组设计出圈程序	78
5.1.1 功能设计要求	51	8.2.1 设计思想	78
5.1.2 具体实现要求	51	8.2.2 参考程序	79
5.2 算法分析	51	8.3 使用分配内存设计出圈程序	80
5.3 参考程序	52	8.3.1 设计思想	80
5.4 组成一个工程	55	8.3.2 参考程序	81
5.5 评价标准	56	8.4 使用结构设计出圈程序	82
第 6 章 函数设计	57	8.4.1 设计思想	82
6.1 设计要求	57	8.4.2 参考程序	83
6.1.1 功能设计要求	57	8.5 生死游戏	85
6.1.2 参考数据及其操作示范	57	8.5.1 直接按出圈方法计算	85
6.2 函数设计注意事项	58	8.5.2 输出没有标记的生存者	88
6.2.1 函数参数传递方式	58	8.6 评价标准	90
6.2.2 函数类型和返回值	59	第 9 章 设计链表	91
6.2.3 结构与函数	59	9.1 设计要求	91
6.3 算法分析	59	9.1.1 功能设计要求	91
6.4 参考程序	61	9.1.2 总体设计	91
6.5 评价标准	63	9.1.3 函数设计	92
第 7 章 用结构数组设计职工		9.2 算法分析	93
文件	65	9.3 参考程序	94
7.1 设计要求	65	9.4 测试程序	98
7.2 算法分析	65	9.5 评价标准	101
7.3 参考程序	66	第 10 章 使用循环链表求解	
7.4 测试程序	71	约瑟夫环游戏	102
7.4.1 测试菜单和读写空文件	71	10.1 简单的循环链表求解	
7.4.2 测试生成和显示职工信息文件	72	约瑟夫环游戏	102
7.4.3 测试生成和显示职工简明		10.1.1 简单的循环链表设计思想	102
信息文件	73	10.1.2 函数实现	103
7.4.4 测试删除操作	73	10.1.3 参考程序	104
7.4.5 建立符合要求的文件	74	10.2 能输出姓名的循环链表求解	
7.5 评价标准	75	约瑟夫环游戏	107
第 8 章 设计出圈游戏	77	10.2.1 设计思想	107
8.1 使用数组设计出圈程序	77	10.2.2 参考程序	107

10.3. 使用动态内存的循环链表求解	12.2.3 确认测试	144
约瑟夫环游戏	12.3 程序的测试与调试	145
10.3.1 设计思想	12.4 测试用例设计技术	147
10.3.2 参考程序	12.4.1 逻辑覆盖法	147
10.4 优化循环链表求解约瑟夫环游戏 ...	12.4.2 等价划分法	150
10.4.1 设计思想	12.4.3 边值分析法	151
10.4.2 参考程序	12.4.4 因果图法	151
10.4.3 扩充设计	12.4.5 错误猜测法	152
10.5 评价标准	12.5 设计学生成绩管理程序	152
第 11 章 链表文件信息管理	12.5.1 功能设计要求	152
11.1 设计要求	12.5.2 总体设计	154
11.1.1 功能设计要求	12.5.3 函数设计	155
11.1.2 总体设计	12.6 参考程序	157
11.2 record 程序的模块设计	12.7 测试示例	169
11.3 record 程序的测试	12.7.1 菜单项及空表和空文件测试 ...	169
11.4 评价标准	12.7.2 测试建表	170
第 12 章 综合课程设计	12.7.3 测试读取文件	171
12.1 实用结构化程序设计基础	12.8 评价标准	173
12.1.1 模块化程序设计	12.9 增加功能实例	174
12.1.2 分块开发	12.9.1 增加功能	174
12.1.3 工程文件	12.9.2 增加的函数和文件	174
12.2 软件测试	12.9.3 修改程序清单说明	175
12.2.1 模块测试	参考文献	183
12.2.2 组装测试		

第 0 章 概 述



由于各校的情况不一，为了便于根据本校的特点和教学计划选择相应的课程设计内容，本章简要介绍本书课程设计题目及其预期目标。

0.1 课程设计目标

一般来讲，课程设计比教学实验复杂一些，涉及的深度广些并更加接近实用。目的是通过课程设计的综合训练，培养学生实际分析问题、编程和动手的能力，最终目标是想通过这种形式，帮助学生系统掌握该门课程的主要内容，更好地完成教学任务。

本课程设计具有如下特点：

(1) 独立于具体的 C 语言教科书，重点放在 C 语言结构化设计的基本特征上，以“不变”应“万变”，涵盖 C 语言的重要基础知识。

(2) 结合实际应用的要求，使课程设计既覆盖知识点，又接近工程实际需要。通过激发学习兴趣，调动学生主动学习的积极性，并引导他们根据实际编程要求，训练自己实际分析问题的能力 & 编程能力，并养成良好的编程习惯。

(3) 结合实例阐述了计算机解题方法及典型算法。

(4) 通过详细的实例，循序渐进地启发学生完成设计。课程设计将要求、算法和源程序分开，为学生创造独立思考的条件。学生在充分理解要求和算法的前提下，完全可以不按书中提供的参考程序，而设计自己的应用程序。

(5) 有些课程设计提出了一些改进措施或要求，可以供有兴趣的学生扩充自己的设计。

(6) 课程设计基本按教学顺序设计，尽量避免涉及后续章节的知识。后续的课程设计尽量引用前面的课程设计内容，以便增加印象并加深理解。

(7) 对于同一类型的实验，提供不同的实现方法，方便选择。

(8) 课程设计分为基本部分与技能提高训练，以满足不同学校和不同学生的要求。

(9) 提供一个综合课程设计，这个课程设计的重点放在强化 C 语言结构化设计的知识和进一步锻炼学生的动手能力上。

(10) 这些课程设计的思想和方法还可以作为学生做毕业论文时的参考资料。

另外，在实际编程中，为了提高编程质量，对空行、空格和注释均有要求。本书也尽可能地根据实际编程要求给出空行、空格和注释，有时因为标题和页码等实际原因，也会适当减少空行、空格和注释，但希望学生在书写代码时，还是严格按照要求处理，以便建立良好的编程风格。

《C 程序设计课程设计》自 2004 年出版以来，虽然多次重印，但该书总体内容偏深。在听取了許多读者的意见之后，又编写了一本更接近大多数院校教学需要的课程设计——《C 程序设计课程设计 第 2 版》。本课程设计偏重基础训练，通过课程设计的题目明确训练的内容，训练由易而难，逐步深入。有些题目（例如出圈游戏）始终贯穿其中，通过不同的方法

对其求解，这样可以使读者体会结构化编程的风格和特点。

0.2 课程设计结构

C程序的组成是函数调用，所以函数是重点。函数设计中的难点是选择函数类型及其参数传递方式。数据类型及程序控制方式是C语言的基础；数组、指针和结构的使用技术是编程的核心技术。学生学习时，常常避开多文件编程和文件的使用，但这些都是程序设计人员必备的知识，因此课程设计特意加强这方面的训练。

本课程设计共选择12个设计题目，不再使用一个独立的例子涵盖许多知识点，而是按层次逐步深入。与第1版相比，程序大大简化。前8个设计都很简单，主要是引入结构化思维方法。全书多次使用不同方法实现出圈游戏，就是为了通过不同实现方式，使学生加深对编程的理解。

这些实验题目及其简要说明如下：

1. 设计菜单（第1章）

菜单是应用程序的界面，控制语句用来改变程序执行的顺序，是实现结构化程序设计的基础。本章的任务是设计一个比较实用的菜单，通过菜单项进行选择，每个菜单项又实现不同的功能。这些模块主要涉及C语言的数据类型及控制语句的使用，是程序设计的基础。

目前C语言都是使用与其相兼容的C++编译器，为了方便程序的调试，本章还简要介绍最流行的Visual C++ 6.0的使用方法。

2. 计算机解题基础（第2章）

计算机强大的逻辑分析功能是由人通过程序赋给它的，一些逻辑问题必须转换成计算机能够看得懂的数学表达式和一定的程序指令。本课程设计主要练习如何将人对问题的思考转换为计算机能解的数学表达式，并结合课程设计讨论使用计算机解题的一般步骤。

3. 使用数组和指针统计成绩（第3章）

数组和指针是C语言的重要概念，数组与指针之间的关系又是容易混淆的地方。另外，字符串数组的用法与普通的数值型数组的用法又有所不同，它们与指针之间的关系也有其特殊性，由于它们的抽象性，给学习也带来一定的困难。

该章的实验就是围绕以上问题展开的，同时有意拓宽字符串数组的用途，引入字符串数组及数组指针。为了适应不同院校的需要，将它们分为几个不同的设计要求，以便根据本校的教学大纲灵活取舍。

4. 典型算法和简单的文件编程结构（第4章）

本课程设计重在熟悉计算机解题的典型算法。递推和递归都是计算机数值计算中的重要算法，本章将用它们解决同一个问题，以便演示递推和递归的区别。

前几章课程设计都采用典型的单文件编程结构，该章将引入一个头文件和一个源文件的编程结构。

5. 多文件编程（第5章）

实际的应用程序总是使用一个工程文件并含有多个文件，为了给编制实用程序打下基础，该章的实验练习使用多文件编程原理以及工程文件的管理方法。

6. 函数设计（第6章）

本课程设计要求设计一个公司职工的数据结构，并使用结构指针数组存储员工信息，统计公司员工工资总额和平均工资，目的是深入了解结构指针数组的使用方法以及如何设计函数的类型及其参数传递方式。

7. 用结构数组设计职工文件（第7章）

本课程设计的目的是学习使用结构数组建立职工档案信息文件并掌握文件读写的基本功能。为了节省篇幅，设计不要求使用多文件编程，但要求使用头文件和菜单，并组成一个工程文件。

鉴于软件测试是软件工程的一个重要环节，所以该章要求对编写的程序进行测试。

8. 设计出圈游戏（第8章）

该章将给出4种设计出圈游戏的程序，它们分别使用数组和结构完成，目的是让读者熟悉数组、二维字符串数组、动态内存和结构数组的使用方法。

9. 设计链表（第9章）

本课程设计的目的是学习建立链表，使用链表存储结构信息，增加链表结点及删除链表结点等基本操作。实际设计时，可以增加数据信息，加入检索等功能。

10. 使用循环链表求解约瑟夫游戏（第10章）

该章将使用循环链表，再次设计出圈游戏（求解约瑟夫环游戏）程序。该章给出4种不同的设计方案，目的是让读者接触简单的循环链表概念并进一步熟悉结构化程序设计思想。

11. 链表文件信息管理（第11章）

本课程设计是在第9章的基础上增加文件管理，目的是学习建立链表，使用链表存储结构信息，增加链表结点及删除链表结点等基本操作。实际设计时，可以增加数据信息及检索等功能。本课程设计为选做内容，可以根据实际教学情况决定。

12. 综合课程设计（第12章）

该章总结头文件、多个C语言文件及工程文件的编制方法，并介绍程序测试的知识，以方便学习和查阅。本课程设计将演示如何运用这些知识设计一个实用的小型学生成绩管理程序，培养实际应用能力。建议将本课程设计列为必做内容，但也可以根据实际教学情况决定。本课程设计比上一版本有所简化，但给出一个增加功能的实例，以便为学生扩展功能提供思路。

0.3 评价标准

因为已经有参考程序，所以一般情况下学生都能完成预定设计。如果学生只是按照程序去做，其分数只能在85分以下。为了证明学生已经掌握了设计所涵盖的知识点，应该向学生提一些问题，例如，程序如何实现及其原理等。由于各校情况不一，所以没有在各章的“评价标准”中规定必须提问题才算完成设计任务。

另外，程序的可读性均包含在正确性中记分，不再单列考核标准。

一般遵循如下规律评价：

(1) 严格控制90分，其标准是有创意。

(2) 85分以上，必须全部正确，并有一定改进或者能正确回答设计中的问题。

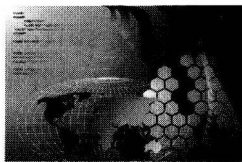
(3) 有少许失误, 可给 75 ~ 83 分。

(4) 错误不多, 给 60 ~ 73 分。

(5) 没有完成规定的要求, 不予及格。

因为课程设计主要是为了锻炼学生, 所以应该鼓励他们不要将此作为负担, 而是提高钻研问题的兴趣, 大胆放手去做。鉴于以上目的, 应该鼓励为主, 避免不及格现象。另外, 评分标准也可以只设“通过”、“没通过”和“优秀”3 档, 以激发学生的学习兴趣。

第 1 章 设计菜单



菜单是应用程序的界面，控制语句用来改变程序执行的顺序，是实现结构化程序设计的基础。本章的任务是设计一个比较实用的菜单，通过菜单选择不同的功能。

本章的设计分为两步：首先设计一个含有多个菜单项的菜单演示程序，然后再为这些菜单项配上相应的功能。

目前 C 语言都是使用与其相兼容的 C++ 编译器，为了方便程序的调试，本章还简要介绍最流行的 Visual C++ 6.0 的使用方法。

1.1 设计一个菜单程序

1.1.1 设计要求

1. 菜单内容

程序运行后，给出 3 个菜单项的内容和输入提示。

```
1. FindNum
2. Diamond
3. Goodbye!
Input 1-3:
```

2. 设计要求

使用数字 1 ~ 3 来选择菜单项，其他输入则不起作用。下面是测试运行的例子：

```
1. FindNum
2. Diamond
3. Goodbye!
Input 1-3: 1
FindNum
1. FindNum
2. Diamond
3. Goodbye!
Input 1-3: 8
2
Diamond
1. FindNum
2. Diamond
3. Goodbye!
Input 1-3: a
1
FindNum
1. FindNum
2. Diamond
3. Goodbye!
Input 1-3: 3
Goodbye! •
```

1.1.2 设计实例

首先编写一个菜单程序，输入 1 ~ 3 以进入相应的选择项。从程序测试结果可知，当选

择相应的选择项时，其输出信息分别为：FindNum、Diamond 和 Goodbye!。

1. 使用 switch 语句实现功能选择

假设输入选择用变量 cn 存储，则可以使用如下结构实现：

```
switch(cn) {
    case 1:
        printf("FindNum\n");
        break;
    case 2:
        printf("Diamond \n");
        break;
    case 3:
        printf("Goodbye!\n");
        break;
}
```

2. 得到 cn 的合理值

可以设计一个函数来输出提示信息并处理输入，这个函数应该返回一个数值 cn，以便供 switch 语句使用。

假设函数名为 menu_select，设计的参考程序如下：

```
int menu_select()
{
    char s;
    int cn;

    printf("1. FindNum\n");
    printf("2. Diamond \n");
    printf("3. Goodbye!\n");
    printf("Input 1-3:");

    do {
        s=getchar();
        cn=(int)s-48;        // 得到数字 1~3
    } while(cn<0||cn>3);

    return cn;
}
```

语句“cn=(int)s-48;”是为了使 switch 中的 case 语句对应数字 1~3。如果这里不进行转换，while 语句应使用 s 进行判别，即

```
while (s<49||s>51);
```

这时的 case 语句则对应 49~51。

3. 实现循环选择

实际使用时，只有选择 3，程序才能结束运行，这就要使用循环控制。

这里使用 for 循环语句来实现菜单的循环选择，为了结束程序的运行，可用“exit(0);”语句代替“case:3”中的“break;”语句。

4. 评论

这个设计是以学过的知识为基础，所以在纠错上有一定的局限性。使用菜单的设计需要用到数组的知识。也可以不将键入的字符转为数字，而直接使用字符进行判别。

1.1.3 程序清单

```

#include<stdio.h>           // 包含系统头文件
#include<stdlib.h>
#include<ctype.h>

int menu_select();          // 声明菜单选择函数原型
void main()
{
    for (; ;) {
        switch (menu_select()) {
            case 1:
                printf("FindNum\n");
                break;
            case 2:
                printf("Diamond\n");
                break;

            case 3:
                printf("Goodbye!\n");
                exit(0);
        }
    }
}
// 菜单选择操作
int menu_select()
{
    char s;
    int cn;

    printf("1. FindNum\n");
    printf("2. Diamond \n");
    printf("3. Goodbye!\n");
    printf("Input 1-3:");
    do {
        s=getchar();
        cn=(int)s-48;
    } while(cn<0||cn>3);

    return cn;
}

```

1.2 增加菜单项的处理功能

本节的设计是以上一节设计的菜单为基础，增加菜单项的处理功能。

1.2.1 菜单项设计要求

这其实是两道编程题，目的是锻炼学生使用控制语句的能力。

1. FindNum

一只老鼠咬坏了账本，公式中的符号□代表被老鼠咬掉的地方。要想恢复下面的等式，应在□中填上哪个相同的数字？

$$3 \square \times 6237 = \square 3 \times 3564$$

利用计算机的计算速度快的特点，把所有可能的数都试一下，从中找出符合条件的数。这就是所谓的枚举法。

只有 0～9 这十个数字，显然已经排除了为 0 的可能性，将 1～9 这九种可能性拿来试验，即可找到合适的数字。由此可见，这道题就是寻找满足下式的 i 值：

$$(30+i)*6237=(10*i+3)*3564$$

2. Diamond

本题要求编制打印以下图案的程序，注意题目还要求在左边留出一定的空格。

```

*
***
*****
*****
*****
***
*

```

通过观察图像的组成特点，可以把它们分成两部分：上面4行和下面3行。上面行按递增计数，下面行则按递减计数。假设第一个“*”距左边15个空格，使用字符“1”模拟的显示图形如下：

```

111111111111111*
11111111111111***
11111111111111*****
11111111111111*****
11111111111111*****
11111111111111***
11111111111111*

```

使用算法描述如下：

```

BEGIN
  for(i=1; i<=4; i++)
    for(j=1; j<=16-i; j++)
      print space;
    endforj
    for(k=1; k<=(2*i-1); k++)
      print *
    endfork
    CR
  endfori
  for(i=1; i<=3; i++)
    for(j=1; j<=i+12; j++)
      print space
    endforj
    for(k=1; k<=(7-2*i); k++)
      print *
    endfork
    CR
  endfori
END

```

// 从第1行到第4行
 // 第i行右移的次数
 // 打印空格
 // 结束j循环
 // 输出*的个数
 // 打印"*"号
 // 结束k循环
 // 换行
 // 结束i循环
 // 从倒数第3行到倒数第1行
 // 第i行右移的次数
 // 打印空格
 // 结束j循环
 // 输出*的个数
 // 打印"*"号
 // 结束k循环
 // 换行
 // 结束i循环

3. Goodbye!

这里除了输出一个简单的信息之外，还必须结束程序运行。使用语句“exit(0);”即可正确地结束程序运行。

1.2.2 建立控制台文件

要先建立控制台文件，才能编写源程序。

图1-1给出结果图，制作方法参考1.4节。建立了c文件后，就可以在其上输入源程序。

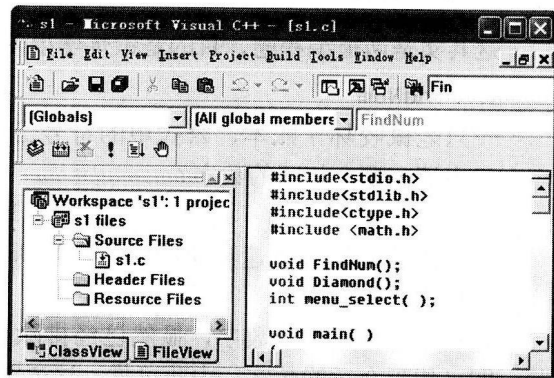


图 1-1 控制台文件

1.2.3 源程序清单

```
#include<stdio.h>
#include<stdlib.h>
#include<ctype.h>
#include <math.h>

void FindNum();
void Diamond();
int menu_select();

void main()
{
    for(;;) {
        switch (menu_select()) {
            case 1:
                FindNum();
                break;
            case 2:
                Diamond();
                break;
            case 3:
                printf("Goodbye!\n");
                exit(0);
        }
    }
}

// 菜单选择操作
int menu_select()
{
    char s;
    int cn;

    printf("1. FindNum\n");
    printf("2. Diamond\n");
    printf("3. Goodbye!\n");
    printf("Input 1-3: ");
    do {
        s=getchar();
        cn=(int)s-48;
    } while(cn<0||cn>3);

    return cn;
}

// 求正确的数字完成等式
// 答案: nun=6
void FindNum()
{
    int i;

    printf("要使下面的等式成立, 应在□中填上哪个相同的数字? \n");
    printf("      3 □ ×6237= □ 3×3564\n");
    printf("求解结果为: \n");
    for(i=1; i<10; i++)
        if(((30+i)*6237==(10*i+3)*3564)
        {
            printf("number=%d\n", i);
            break;
        }
}
```