

普通高等教育“十三五”规划教材  
高等教育创新性实验教材

# C 语言程序设计实验实训

金东萍 刘一臻 王彦明 主 编  
畅惠明 李辽辉 副主编

西南交通大学出版社  
· 成 都 ·

## 内容简介

全书包括上机实验、综合实训和等级考试三部分内容。上机实验部分含基本的 C 语言程序设计实验和单元测试,既考虑了读者在学习过程中的上机实验需要,又考虑了程序设计的典型算法应用,有针对性地提高读者的编程能力;综合实训部分含基本实训和项目实训,是为培养读者的 C 语言程序设计应用能力和开发能力而编写的;等级考试部分含全国计算机等级考试(二级)考试大纲、模拟题、参考答案及解析等内容。

本书适合于高等院校理工类各专业学习使用,也适合 C 语言自学者或参加各种 C 语言考试的读者学习使用。

---

### 图书在版编目(CIP)数据

C 语言程序设计实验实训 / 金东萍, 刘一臻, 王彦明  
主编. — 成都: 西南交通大学出版社, 2016.1  
高等教育创新性实验教材  
ISBN 978-7-5643-4340-8

I. ①C… II. ①金… ②刘… ③王… III. ①C 语言—  
程序设计—高等学校—教材 IV. ①TP312

中国版本图书馆 CIP 数据核字(2015)第 243922 号

---

### 高等教育创新性实验教材 C 语言程序设计实验实训

金东萍 刘一臻 王彦明 主编

责任编辑	张华敏
特邀编辑	蒋雨杉 唐建明
封面设计	何东琳设计工作室 西南交通大学出版社
出版发行	(四川省成都市二环路北一段 111 号 西南交通大学创新大厦 21 楼)
发行部电话	028-87600564 028-87600533
邮政编码	610031
网 址	<a href="http://www.xnjdcbs.com">http://www.xnjdcbs.com</a>
印 刷	成都勤德印务有限公司
成品尺寸	185 mm × 260 mm
印 张	14.5
字 数	362 千
版 次	2016 年 1 月第 1 版
印 次	2016 年 1 月第 1 次
书 号	ISBN 978-7-5643-4340-8
定 价	29.50 元

图书如有印装质量问题 本社负责退换  
版权所有 盗版必究 举报电话: 028-87600562

# 前 言

“C 语言程序设计”课程是高等院校理工类非计算机专业学生必修的专业基础课程。要学好 C 语言，需要进行大量的实际操作和实践训练。本书是为配合“C 语言程序设计”课程的学习而编写的实验实训教材，以帮助读者深入理解和掌握 C 语言程序设计所涉及的概念、方法与技巧。在编写本书的过程中，我们以培养读者的实践能力和创新能力为理念，注重基础，突出应用。全书内容共分为三部分：

第一部分 上机实验。含基本的 C 语言程序设计实验和单元测试。实验内容由浅入深，其中有些题目难度高于教科书中的例题，目的是适合不同水平基础的读者。

第二部分 综合实训。含基础实训和项目实训。是为培养读者的 C 程序设计应用能力和开发能力而编写的，同时给出了解题思路、程序分析等，以便于读者理解、思考。

第三部分 等级考试。含全国计算机等级考试（二级）考试大纲、模拟题、参考答案及解析等内容。

本书是一本实用的 C 语言实验实训教材，编者尽量突出 C 语言程序设计课程的特点，力求以知识为主导，由易到难，并结合国家计算机等级考试和计算机专业软件水平考试的内容，让读者通过综合运用所学过的知识，学习编写规模稍大、实用性较强的程序，对于提高读者的实践能力、创新能力和编程兴趣，将会有很好的效果。因此本书不仅适合于高等院校理工类各专业学习使用，也适合 C 语言自学者或参加各种 C 语言考试的读者学习使用。

为了便于读者自测，本书的所有习题均附有参考答案，其中编程题答案不一定是最好的和唯一的，仅仅起到示范或参考作用。书中的所有程序均在 Visual C++6.0 环境下调试通过。

本书由金东萍、刘一臻、王彦明任主编，畅惠明、李辽辉任副主编，刘慧宇任参编。具体分工如下：金东萍编写第 1 章至第 4 章；刘一臻编写第 5 章和第 6 章；金东萍、王彦明、畅惠明、李辽辉和刘慧宇共同编写第 7 章至第 12 章。金东萍负责全书的总体规划设计和统稿工作。

由于作者水平所限，书中缺点和错误在所难免，敬请各位读者谅解，并恳请读者及时提出宝贵意见和建议，我们将不胜感激。

编 者

2016 年 1 月

# 目 录

## 第一部分 上机实验

第 1 章 Visual C++6.0 集成开发环境 .....	2
实验 1 Visual C++6.0 实验环境 .....	2
实验 2 使用 Visual C++6.0 运行 C 语言程序的方法 .....	12
单元测试 .....	19
第 2 章 C 语言的基础知识 .....	20
实验 1 简单程序设计 .....	20
实验 2 阅读程序验证结果 .....	25
单元测试 .....	28
第 3 章 C 语言程序设计基本结构 .....	31
实验 1 顺序结构程序设计 .....	31
实验 2 选择结构程序设计 .....	36
实验 3 循环结构程序设计（一） .....	44
实验 4 循环结构程序设计（二） .....	50
单元测试 .....	55
第 4 章 函 数 .....	75
实验 1 函数（一） .....	75
实验 2 函数（二） .....	78
单元测试 .....	81
第 5 章 数 组 .....	89
实验 1 数组（一） .....	89
实验 2 数组（二） .....	91
实验 3 数组（三） .....	95
单元测试 .....	99
第 6 章 指 针 .....	106
实验 1 指针（一） .....	106
实验 2 指针（二） .....	109
实验 3 指针（三） .....	111
单元测试 .....	116
第 7 章 结构体类型及其他构造类型 .....	123
实验 结构体类型及其他构造类型的应用 .....	123
单元测试 .....	127
第 8 章 文 件 .....	134
实验 文件操作 .....	134

单元测试 .....	140
第 9 章 预处理指令与位运算 .....	144
实验 预处理指令与位运算的应用 .....	144
单元测试 .....	146
第二部分 综合实训	
第 10 章 基础实训 .....	152
实训一 .....	152
实训二 .....	154
实训三 .....	156
实训四 .....	157
实训五 .....	159
实训六 .....	160
实训七 .....	161
实训八 .....	163
第 11 章 项目实训 .....	166
实训一 学生成绩管理程序设计 .....	166
实训二 职工信息管理程序设计 .....	169
实训三 会议预约管理程序设计 .....	173
实训四 电话订餐管理程序设计 .....	177
第三部分 等级考试	
第 12 章 全国计算机二级 C 语言等级考试 .....	183
全国计算机等级考试二级 C 语言程序设计考试大纲 .....	183
全国计算机等级考试二级公共基础知识考试大纲 .....	186
第一套模拟题 .....	187
第二套模拟题 .....	199
第三套模拟题 .....	212
参考文献 .....	226



# 第一部分 上机实验

实验是学习 C 语言程序设计十分重要的环节。通过有针对性的上机实验，读者可以进一步熟悉 C 语言的功能，逐步理解和掌握 C 语言程序设计的思想、步骤和方法，通过大量的编程训练，有利于培养读者应用 C 语言程序设计解决实际问题的能力、应用开发的能力和创新能力。

当然，如果你是第一次接触计算机软件编程，不要期望立即就能写出一个实用的程序，也不必惧怕自己不会写程序，编程本来就是一个循序渐进的过程，程序设计不是听会的，也不是看会的，而是自己上机练会的。请根据教师的引导，独立自主、大胆地进行所要求完成的实验工作。动手才能找到感觉，动手才能找到自信，动手才能有成就感。

本部分共 9 章内容，每章包含上机实验和单元测试两部分。既考虑了读者在学习过程中的上机实验需要，又有针对性地提高读者的编程能力。单元测试内容不仅能满足教学需要，而且对于参加各类计算机考试的读者，也是很好的辅导教材。

需要指出的是，为适合不同水平基础的读者，其中有些题目难度高于教科书中的例题。这里提供的解答未必就是最好的，仅仅是提供了一种解题思路，并用规范的书写风格表达出来。为了达到理想的实验效果，读者应该自己先编写程序，然后上机调试，在程序出现问题时，或在自己的程序完成以后，再参考答案作比较，这样做效果会更好。

# 第 1 章 Visual C++6.0 集成开发环境

## 实验 1 Visual C++6.0 实验环境

### 【实验目的】

1. 了解和熟悉 C 语言程序设计开发环境 Visual C++6.0。
2. 了解 C 语言程序设计的一般步骤。

### 【实验内容】

Visual C++6.0, 简称 VC++ 6.0、VC 或者 VC6.0, 是微软推出的一款 C++编译器, 将“高级语言”翻译为“机器语言”的程序。Visual C++是一个功能强大的可视化软件开发工具。自 1993 年 Microsoft 公司推出 Visual C++1.0 后, 随着其新版本的不断问世, Visual C++已成为专业程序员进行软件开发的首选工具。

Visual C++6.0 不仅是一个 C++ 编译器, 而且是一个基于 Windows 操作系统的可视化集成开发环境 (integrated development environment, IDE)。Visual C++6.0 由许多组件组成, 包括编辑器、调试器以及程序向导 AppWizard、类向导 Class Wizard 等开发工具。这些组件通过一个名为 Developer Studio 的组件集成为和谐的开发环境。

Visual C++6.0 除了包含文本编辑器、C/C++混合编译器、连接器和调试器外, 还提供了功能强大的资源编辑器和图形编辑器, 利用“所见即所得”的方式完成程序界面的设计, 大大减轻了程序设计的劳动强度, 提高了程序设计的效率。

Visual C++6.0 的功能强大, 用途广泛, 不仅可以编写普通的应用程序, 还能很好地进行系统软件设计及通信软件的开发。

### 一、安装和启动

安装方法: 首先下载软件安装包, 运行 AUTORUAN.exe, 出现 Visual C++6.0 安装界面, 选择中文版, 如图 1-1 所示。

然后按照安装程序的指导完成安装过程。软件会长时间搜索硬盘, 确定你的配置以及环境, 此时一定不要关闭软件。搜索完成后, 会提示如图 1-2 所示的选择界面, 选择“是”。



图 1-1 Visual C++6.0 安装界面



图 1-2 Visual C++6.0 安装选择界面

继续选择安装类型 *Typica* 和安装路径。最后提示是否注册环境变量，打上对钩，点击“OK”，结束安装。

启动 Visual C++6.0 的方法：点击开始菜单的“所有程序”，单击 Microsoft Visual C++6.0 图标，或者在 Window 桌面上建立一个快捷方式，双击它即可启动。

## 二、创建工程项目

使用 Visual C++ 6.0 编写并处理的任何程序都要创建一个与之相关的工程, Visual C++ 6.0 用工程化的管理方法把一个应用程序中的所有相互关联的一组文件组织成一个有机的整体, 构成一个项目, 也称为工程项目。工程项目以文件夹方式管理所有源文件, 工程项目名作为文件夹名。文件夹中包含源程序文件 (.C)、项目文件 (.DSP)、项目工作区文件 (.DSW)、项目工作区配置文件 (.OPT) 和 Debug (调试) 子文件夹等。

首先要创建一个工程项目(project), 用来存放 C 程序的所有信息。创建一个工程项目的操作步骤如下:

第一步，启动 Visual C++6.0，进入 Visual C++6.0 环境后，在主菜单中选择“文件”菜单，在其下拉菜单中单击“新建”项，弹出一个“新建”对话框，单击上方的选项卡“工程”，从中选择“Win32 Console Application”工程类型，在“工程名称[N]:”一栏中输入工程名，例如 wodec，在“位置[C]:”一栏中输入路径，例如：F:\WODEC\wodec，如图 1-3 所示，继续单击“确定”按钮。



图 1-3 创建工程项目（1）

第二步，弹出对话框“Win32 Console Application-步骤 1 共 1 步”，如图 1-4 所示，选择“一个空工程[E]”按钮，单击“完成”按钮。

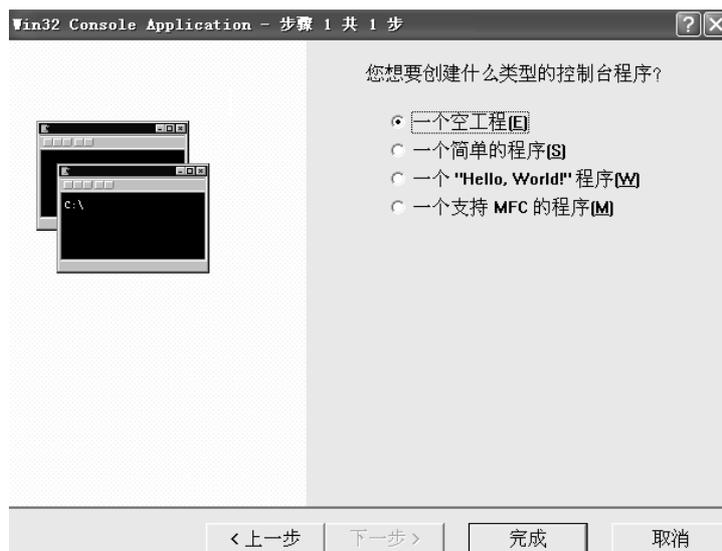


图 1-4 创建工程项目（2）

此对话框主要是询问用户想要创建什么类型的工程，各选项的含义如下：

- 一个空工程：生成一个空的工程，工程内不包括任何文件。
- 一个简单的程序：生成包含一个空的 main ( ) 函数和一个空的头文件的工程。
- 一个“Hello World!”程序：是包含有显示“Hello World!”字符串的输出语句。
- 一个支持 MFC 的程序：可以利用 Visual C++ 6.0 所提供的类库来进行编程。

在弹出的“新建工程信息”对话框中，单击“确定”按钮，至此完成工程项目的创建（文件夹名称 wodec）。

### 三、新建一个 C 源程序

创建工程项目完成之后，再创建 C 源程序文件，继而进行源程序的输入和编辑工作。

**第一步**，创建 C 源程序文件。进入 Visual C++6.0 环境后，在主菜单中选择“文件”菜单，在其下拉菜单中单击“新建”项，弹出一个“新建”对话框，单击上方的选项卡“文件”，在对话框内单击“C++ Source File”项，在“文件名[N]:”栏内输入新添加的源文件名，如 wodec.c（注意扩展名为.c），在“位置[C]:”一栏输入文件路径，如图 1-5 所示，此时新建了一个 C 源程序文件，最后单击“确定”按钮，完成一个 C 源程序文件的创建。

**注意：**输入 C 源程序文件名一定要加上扩展名“.c”，否则系统会为文件添加默认的 C++ 源文件扩展名“.CPP”。

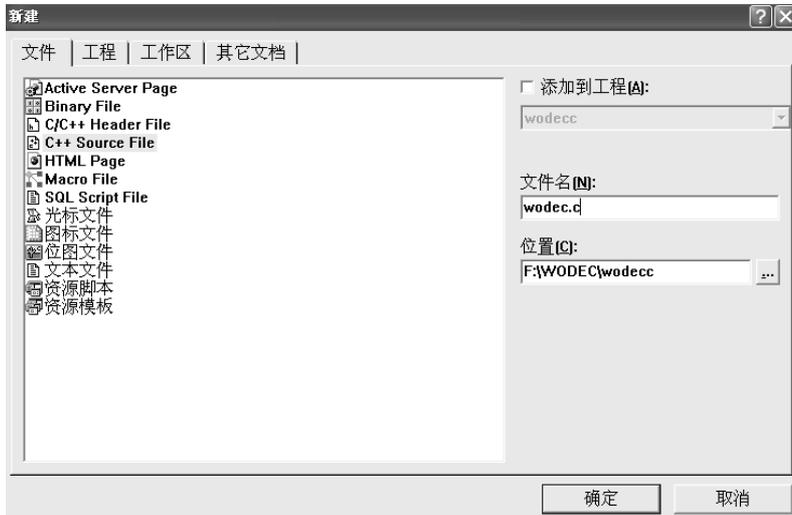


图 1-5 创建 C 源程序文件（1）

**第二步**，如图 1-6 所示，在文件编辑区输入和编辑源程序；或者在左侧的工作区窗口，单击下方的“FileView”选项卡（文件视图显示），单击展开，打开“Source Files”文件夹，单击源程序文件名，进行输入和编辑源程序。

**第三步**，保存文件。在主菜单中选择“文件”菜单，在其下拉菜单中单击“保存”项，保存工作区文件；或者在工具条上单击保存图标进行保存。

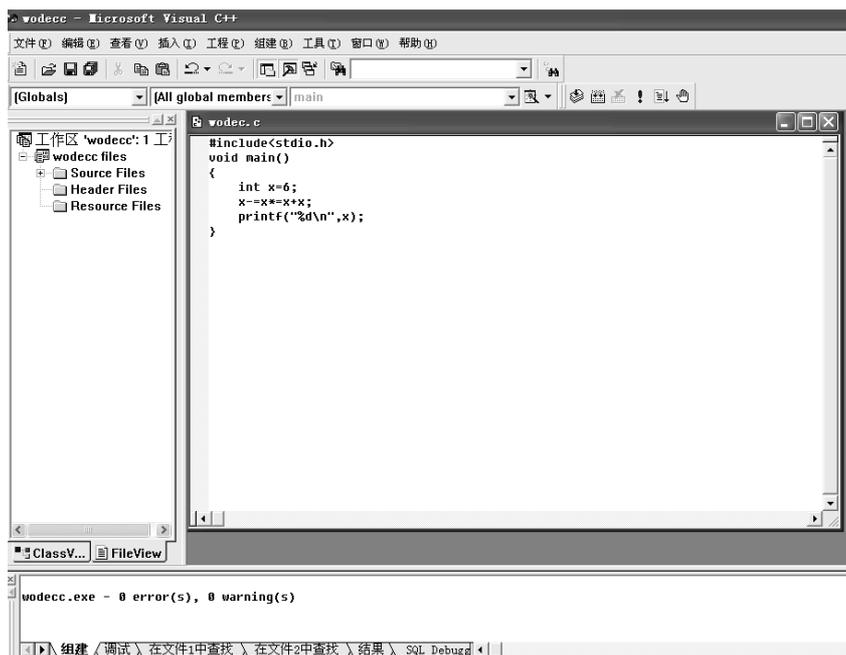


图 1-6 创建 C 源程序文件 (2)

#### 四、打开已存在的程序

首先按前面介绍的方法进入 Visual C++6.0 环境后, 打开已经存在的 C 源程序, 方法如下:

方法一, 选择主菜单“文件”, 单击下拉菜单中的“打开工作空间(W)”, 在弹出的对话框“打开工作区”中选择要打开的工作区文件 wodec.dsw, 单击“打开”按钮。

在左侧的工作区窗口, 单击下方的“FileView”选项卡, 打开“Source Files”文件夹, 双击 wodec.c, 在文件编辑区中会出现 wodec.c 源程序, 此时可以进行编辑和修改, 如图 1-7 所示。

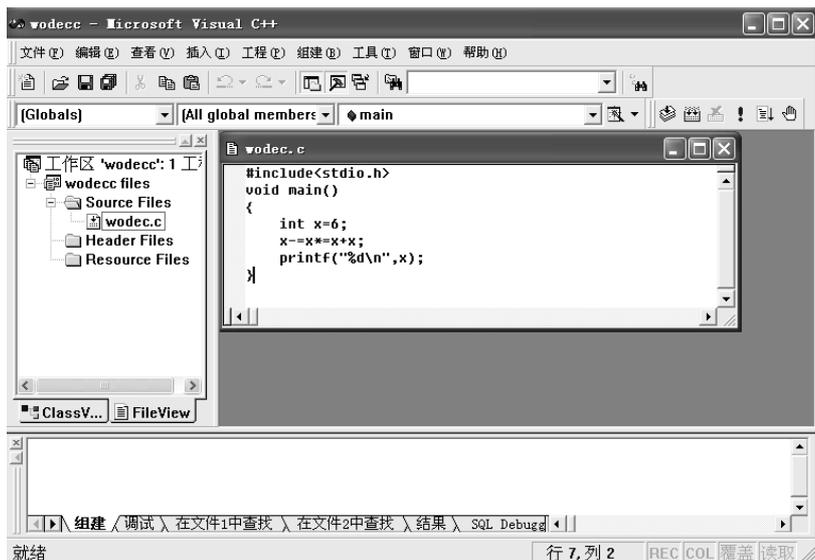


图 1-7 打开 wodec.c 源程序

方法二，选择主菜单“文件”，单击下拉菜单中的“打开”，在“打开”对话框中单击要打开的文件 wodec.c，单击“打开”按钮，在文件编辑区会出现 wodec.c 源程序，此时可以进行编辑和修改。

方法三，按路径寻找已经存在的程序。以查找 wodec.c 程序为例，双击“我的电脑”，双击 F 盘，打开 wodec 文件夹，双击 wodec.c 文件，则进入了 Visual C++6.0 环境，而且程序已经显示在文件编辑区了，此时可以进行编辑和修改。

注意：编辑和修改后的文件要进行保存。

## 五、程序的编译、连接、运行和退出

在对 C 源程序进行编辑和修改后，若要运行该程序，还要再对该源程序进行编译和连接。步骤和方法如下：

**第一步，编译。**C 源程序以文本形式存储，它不能由计算机直接执行，计算机只能执行机器语言编写的程序，因此要把源程序文件翻译成特定机器语言描述的目标程序文件，这个过程就是编译。编译是将源程序代码翻译成目标文件（扩展名为.obj），在这个过程中可能发现源程序的语法、代码等错误，需要进行修改。

方法：选择主菜单“组建”，单击下拉菜单中的“编译”命令，或单击工具条上的图标。在输出（Output）窗口中将显示编译过程中检查出的错误或警告信息，在错误信息处单击鼠标右键或双击鼠标左键，可以使输入焦点跳转到引起错误的源代码处的大致位置，此时可进行修改。如图 1-8 所示，输出窗口中提示“F:\wodec\wodecc\wodec.c(7) : error C2143: syntax error : missing ';' before '}'”，提示在“}”之前缺少分号，同时在程序窗口中已经标注出了出错语句的大致位置，此时可修改源程序，再进行编译，直到没有错误为止。当没有错误与警告出现时，输出窗口所显示的最后一行应该是“hello.obj — 0 error(s), 0 warning(s)”。

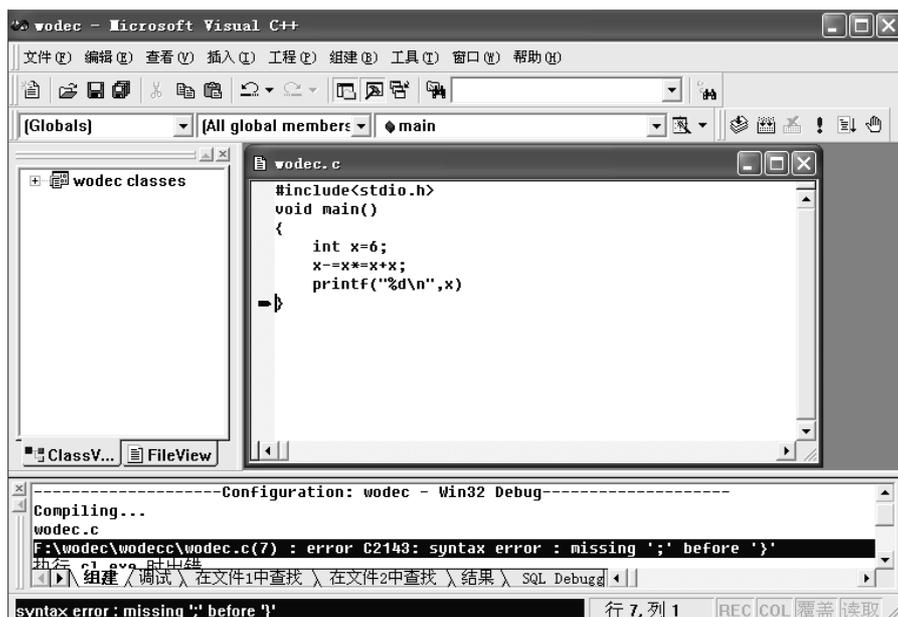


图 1-8 在编译过程中显示错误提示信息

**第二步，连接。**源程序经编译后生成的目标文件（.obj）还不能在计算机上直接执行。因为C语言源程序一般都由若干个独立的程序模块组成，这些模块往往分别进行编辑和编译，编译后生成的是一个相对独立的目标程序模块。利用连接程序按一定的方式将它们连接、装配成一个整体后，才生成了可执行的目标码文件。可执行文件的扩展名为.exe。

方法：选择主菜单“组建”，单击下拉菜单中的“组建”命令，或单击工具条上的图标，完成连接操作。

程序连接完成后生成的目标文件（.obj）、可执行文件（.exe）存放在当前工程项目所在文件夹的“Debug”子文件夹中。

另外，选择主菜单“组建”中的“全部重建”命令，允许用户编译所有的源文件，而不管它们何时曾经被修改过。选择主菜单“组建”中的“批组建”命令，能单步重新建立多个工程文件，并允许用户指定要建立的项目类型。

**第三步，运行程序。**当通过了编译、连接之后，就可以运行该程序了。运行程序时，程序的每条语句都会被执行。如果程序向操作人员请求数据，这时程序将暂停运行，等待用户输入数据。程序的运行结果将在终端上显示。

方法：选择主菜单“组建”，单击下拉菜单中的“执行”命令，或单击工具条上的图标，执行程序，便会弹出一个新的窗口，即运行窗口。按照程序的要求正确输入后，程序即可执行，运行窗口将显示程序运行的结果。按任意键返回编辑状态。

对于比较简单的程序，可以直接选择运行命令，编译、连接和运行一次完成。

如果完成了一个程序的运行，不再对它进行其他处理，可以选择主菜单“文件”，单击下拉菜单中的“关闭工作空间”命令，以结束对该程序的操作。

**第四步，退出 Visual C++6.0 环境。**单击主菜单“文件”的下拉菜单“退出”命令，如图 1-9 所示，可退出 Visual C++6.0 环境。

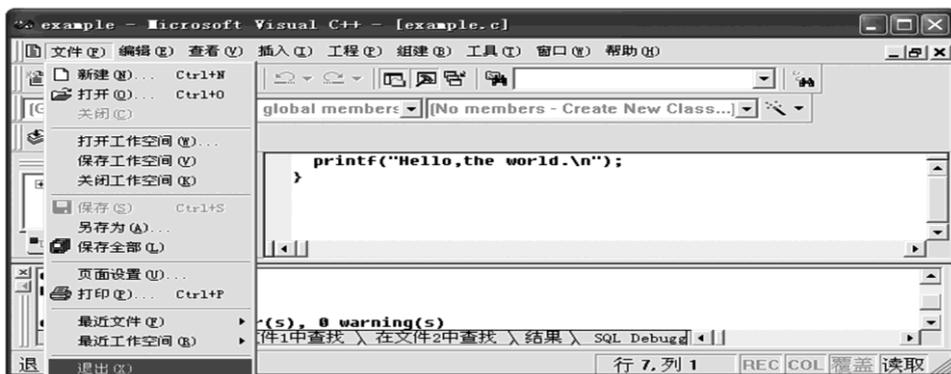


图 1-9 退出 Visual C++6.0 环境

## 六、建立多个 C 源程序

**方法一，**在一个工程项目中建立多个 C 源程序文件。选择主菜单“文件”，单击下拉菜单中的“打开工作空间”命令，在“打开工作区”对话框内找到并单击要打开的工作区文件“wodec.dsw”，单击“打开”按钮；再选择主菜单“工程”中的“增加到工程→新建”命令，弹出“新建”对话框，单击选项卡“文件”，选择“C++SourceFile”项，并在“文件名[N]”

中输入“文件名.c”，单击“确定”按钮，至此在一个工程项目中又建立了一个新的 C 源程序文件。以此方法，可以在一个工程项目中建立多个 C 源程序文件。

在左侧的工作区窗口，单击下方的“FileView”选项卡，可以看见新增加的源程序。注意，在一个工程项目中，同时只能运行一个 C 源程序，当运行一个程序时，需要将其其他的源程序删除，否则连接程序会出现错误提示。删除方法：在“FileView”显示窗口中单击要删除的源程序，按“Delete”键即可（不是真删除，仍然保存在工作区中）。

**方法二**，在工程项目中添加已经存在的 C 源程序文件。选择主菜单“文件”，单击下拉菜单中的“打开工作空间”命令，在“打开工作区”对话框内找到并单击要打开的工作区文件“wodec.dsw”，单击“打开”按钮。

选择主菜单“工程”，单击下拉菜单中的“增加到工程→文件”命令，弹出“插入文件到工程”对话框，找到并单击已经存在的 C 源程序文件，单击“确定”按钮完成添加。

## 七、调试程序

在编写较长的程序时，能够一次成功而不含有任何错误绝非易事，对于程序中的错误，系统提供了易用且有效的调试手段。调试是一个程序员最基本的技能，不会调试的程序员就意味着即使学会了一门语言，也不能编制出好的软件。

### （一）调试程序环境介绍

#### 1. 进入调试程序环境

选择主菜单“组建”，单击下拉菜单中的“开始调试”命令，选择下一级提供的调试命令，或者在菜单区空白处单击鼠标右键，在弹出的菜单中选中“调试”项。激活调试工具条，选择需要的调试命令，系统将会进入调试程序界面；同时提供多种窗口监视程序运行，通过单击“调试（Debug）”工具条上的按钮，可以打开/关闭这些窗口，参考图 1-10 所示的调试程序界面。



图 1-10 调试程序界面

## 2. Watch (观察) 窗口

单击调试 (Debug) 工具条上的 Watch 按钮, 就出现一个 Watch 窗口。

系统支持查看程序运行到当前指令语句时变量、表达式和内存的值。所有这些观察都必须在断点中断的情况下进行。

观看变量的值最简单, 当断点到达时, 把光标移动到这个变量上, 停留一会就可以看到变量的值。

还可以采用系统提供一种被称为 Watch 的机制来观看变量和表达式的值。在断点中断状态下, 在变量上单击右键, 选择 Quick Watch, 就弹出一个对话框, 显示这个变量的值。

在该窗口中输入变量或者表达式, 就可以观察变量或者表达式的值。注意 “++” 和 “--” 运算符绝对禁止用于这个表达式中, 因为这个运算符将修改变量的值, 导致程序的逻辑被破坏。

## 3. Variables (变量) 窗口

单击调试 (Debug) 工具条上的 “Variables” 按钮弹出 Variables 窗口, 显示所有当前执行上下文中可见的变量的值。特别是当前指令语句涉及的变量, 以红色显示。

## 4. Memory (内存)

由于指针指向的数组, Watch 窗口只能显示第一个元素的值。为了显示数组的后续内容, 或者要显示一片内存的内容, 可以使用 memory 功能。单击调试 (Debug) 工具条上的 “memory” 按钮, 就弹出一个对话框, 在其中输入地址, 就可以显示该地址指向的内存的内容。

## 5. Registers(寄存器)

单击调试 (Debug) 工具条上的 “Registers” 按钮弹出一个对话框, 显示当前的所有寄存器的值。

## 6. Call Stack (调用堆栈)

调用堆栈反映了当前断点处函数是被哪些函数按照什么顺序调用的。单击调试 (Debug) 工具条上的 “Call stack” 显示 Call Stack 对话框。在 Call Stack 对话框中显示了一个调用系列, 最上面的是当前函数, 往下依次是调用函数的上级函数。单击这些函数名可以跳到对应的函数中去。

### (二) 单步执行调试程序

系统提供了多种单步执行调试程序的方法: 可以通过单击调试 (Debug) 工具条上的按钮或按快捷键的方式选择多种单步执行命令。

① 单步跟踪进入子函数 (Step Into, F11), 每按一次 F11 键或按 , 程序执行一条无法再进行分解的程序行, 如果涉及子函数, 则进入子函数内部。

② 单步跟踪跳过子函数 (Step Over, F10), 每按一次 F10 键, 程序执行一行; Watch 窗口可以显示变量名及其当前值, 在单步执行的过程中, 可以在 Watch 窗口中加入所需观察的变量, 辅助加以进行监视, 随时了解变量当前的情况, 如果涉及子函数, 不进入子函数内部。

③ 单步跟踪跳出子函数 (Step Out, Shift+F11), 按键后, 程序运行至当前函数的末尾, 然后从当前子函数跳到上一级主调函数。

④ 运行到当前光标处，当按下 CTRL+F10 后，程序运行至当前光标处所在的语句。  
表 1-1 所示为常用调试命令一览表。

表 1-1 常用调试命令一览表

菜单命令	工具条按钮	快捷键	说明
Go		F5	继续运行，直到断点处中断
Step Over		F10	单步，如果涉及子函数，不进入子函数内部
Step Into		F11	单步，如果涉及子函数，进入子函数内部
Run to Cursor		CTRL+F10	运行到当前光标处
Step Out		Shift+F11	运行至当前函数的末尾，跳到上一级主调函数
		F9	设置/取消断点
Stop Debugging		Shift+F5	结束程序调试，返回程序编辑环境

### (三) 设置断点调试程序

为了方便对较大规模程序进行跟踪，断点是最常用的技巧。断点是调试器设置的一个代码位置。当程序运行到断点时，程序中断执行，回到调试器。调试时，只有设置了断点并使程序回到调试器，才能对程序进行在线调试。图 1-11 所示为设置断点调试程序。

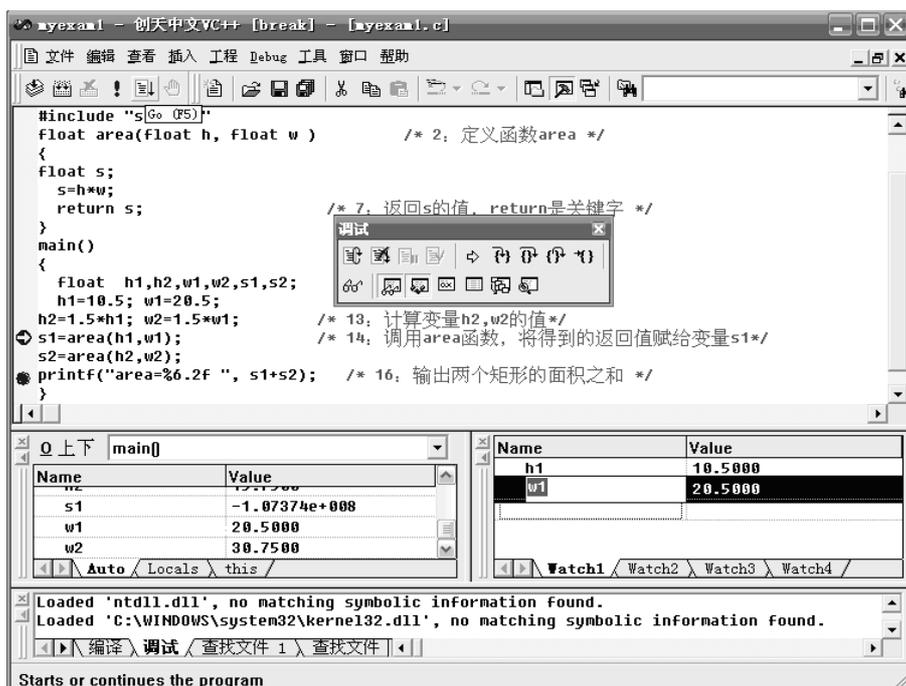


图 1-11 设置断点调试程序

#### 1. 设置断点的方法

可以通过下述方法设置一个断点。首先把光标移动到需要设置断点的代码行上，然后按

F9 快捷键或者单击“编译”工具条上的按钮，断点处所在的程序行的左侧会出现一个红色圆点，参考图 1-11 和表 1-1。

还可以选择主菜单“编辑”，单击下拉菜单中的“断点”命令，弹出“Breakpoints”对话框，打开后点击“分隔符在:”编辑框右侧的箭头，选择合适的位置信息。一般情况下，直接选择 line xxx。

## 2. 程序运行到断点

选择主菜单“组建”中的“开始调试”命令的下一级的“Go”调试命令，或者单击“编译 (Compile)”工具条上的按钮，程序执行到第一个断点处将暂停执行，该断点处所在的程序行的左侧红色圆点上添加了一个黄色箭头，此时，用户可方便地进行变量观察。继续执行该命令，程序运行到下一个相邻的断点。

## 3. 取消断点

只需在代码处再次按 F9 或者单击“编译”工具条上的按钮。

### (四) 结束程序调试，返回程序编辑环境

选择主菜单“调试”中的“Stop Debugging”命令，或者单击“调试”工具条上的按钮，或者单击 Shift+F5 键，可结束程序调试，返回程序编辑环境。

# 实验 2 使用 Visual C++6.0 运行 C 语言程序的方法

## 【实验目的】

1. 进一步了解 C 语言源程序的特点。
2. 进一步熟悉 C 语言编程环境 Visual C++6.0。
3. 掌握编辑、编译、连接和运行一个 C 语言程序的方法。

## 【实验内容】

C 语言程序设计的一般步骤：

- ① 运行程序设计开发工具，进入程序设计开发环境。
- ② 新建一个文件，输入准备好的程序。
- ③ 不要立即进行编译和连接，应该首先仔细检查刚刚输入的程序，如有错误及时改正，保存文件后再进行编译和连接。
- ④ 如果在编译和连接的过程中发现错误，根据系统的提示找出出错语句的位置和原因，改正后再进行编译和连接。直到成功为止。
- ⑤ 运行程序，如果运行结果不正确，修改程序中的内容，直到结果正确为止。
- ⑥ 保存源程序和相关资源。

## 一、编译、连接、运行程序(1)

实验步骤如下：

### 步骤一：建立工程项目

① 进入 Visual C++ 6.0 主窗口，选择主菜单中“文件”→“新建”命令，弹出“新建”对话框，在弹出的对话框中选择“工程”选项卡，从中选择“Win32 Console Application”选项（工程类型中名为 Win32 Console Application 的选项，称为控制台应用程序，它是用来编写和运行 C 语言程序方法中最简单的一种，其入口函数是 main（））。

② 在“工程名称”文本框中输入工程名（如 proj1），接着在“位置”文本框中输入存放工程相关文件的目录（如 d:\chengxu），也可通过单击“…”按钮选择并指定这一文件夾位置，此时 Visual C++ 6.0 会自动在“位置”文本框中用该工程名 proj1 建立一个同名子目录，随后的工程文件及其他相关文件都将存放在这个目录中。图 1-12 所示为“工程”选项卡对话框。



图 1-12 “工程”选项卡对话框

在图 1-12 所示界面中选中“创建新的工作空间”单选按钮，单击“确定”按钮，弹出如图 1-13 所示的项目类型对话框。为了编写和运行一个 C 语言程序，可选中“一个空工程”单选按钮，单击“完成”按钮。

### 步骤二：建立 C 语言源程序文件

① 选择主菜单“文件”菜单中的“新建”命令，在弹出的对话框中选择“文件”选项卡，在其选项卡中选择“C++ Source File”选项，选中右边的“添加到工程”复选框，在其下方的“文件名”文本框中输入文件名（如 hello.c，注意此处扩展名.c 不能省略），单击“确定”按钮，进入输入源程序的编辑窗口。图 1-14 所示为“新建”对话框。



图 1-13 项目类型对话框

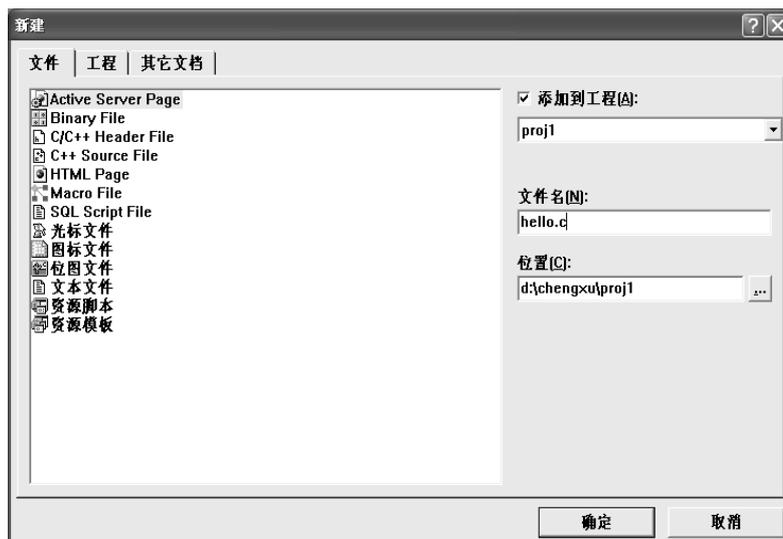


图 1-14 “新建”对话框

② 进入源程序的编辑窗口，输入 C 源程序代码如下：

```
#include<stdio.h>
void main()
{
    printf("Hello World! \n");
}
```

单击 File View 标签，可以看到 Source File 文件夹下的文件 hello.c，图 1-15 所示为编辑源程序窗口。

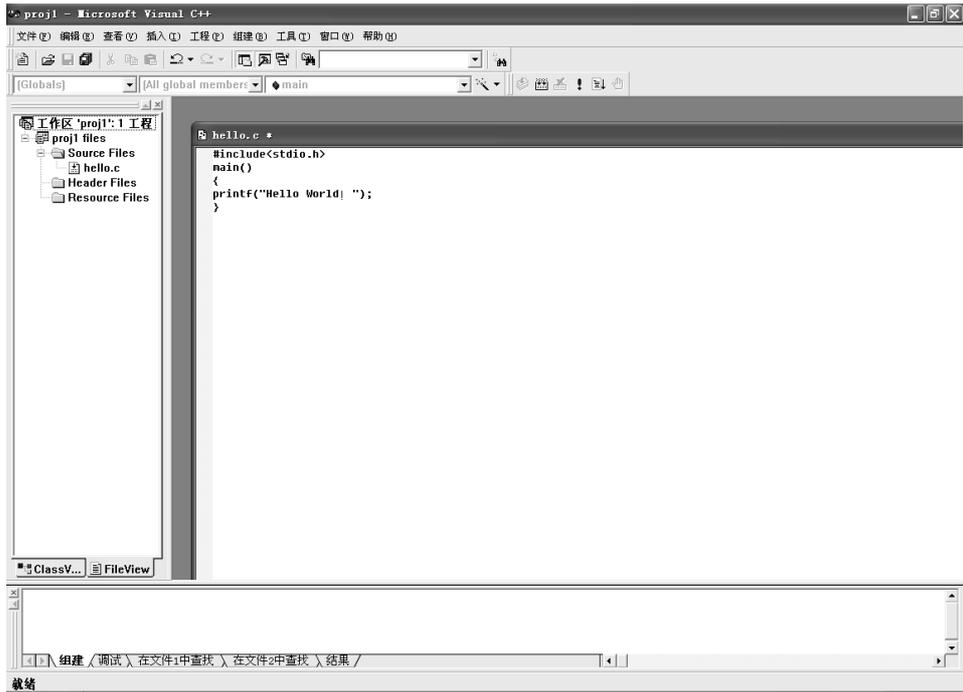


图 1-15 编辑源程序窗口

### 步骤三：编译程序

选择主菜单“组建”中的“编译”命令对程序进行编译（也可通过单击窗口上方工具栏 ，从中选择“编译”按钮  来完成），若编译中发现错误（error）或警告（warning），在输出窗口中将显示它们所在的行及具体的出错或警告信息，双击这些信息，在编辑区的左侧将出现一个“⇔”提示出错行，如图 1-16 所示的编译窗口。

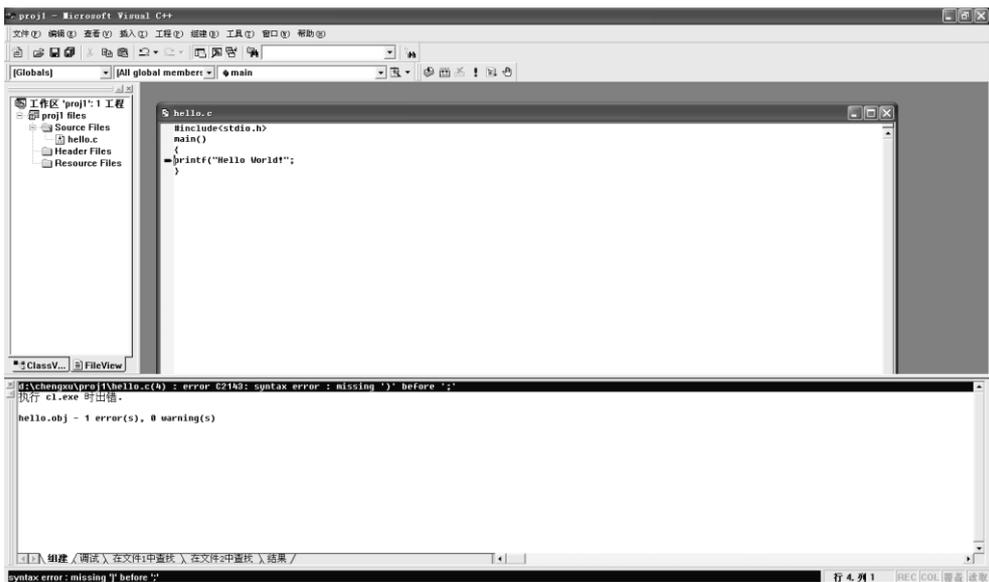


图 1-16 编译窗口

#### 步骤四：连接

选择主菜单“组建”中的“组建”命令来连接生成可执行程序（也可通过单击窗口上方工具栏，从中选择“组建”按钮来完成）。连接成功后，输出窗口显示“proj1.exe — 0 error(s), 0 warning(s)”。

#### 步骤五：运行程序

选择主菜单“组建”中的“执行”命令（也可通过单击窗口上方工具栏中选择“执行”按钮来完成）。运行后将出现一个显示结果界面。其中的提示信息“press any key to continue”是由系统产生的。

至此已经生成并运行了一个完整的程序，完成了一个完整的编程任务。

#### 实验思考：

- ① 将程序中的“\n”去掉，观察程序的运行结果，并和源程序的运行结果对照比较。
- ② `stdio.h` 是头文件，C 提供了多个头文件，`#include<stdio.h>`是预处理命令，将各头文件用`#include`命令包含在程序的首部就可以直接使用了，它不是 C 语句，其后不能加“;”。

#### 注意：

要想编写第二个 C 程序，必须关闭前一个程序的工作空间，方法是：选择主菜单“文件”中的“关闭工作空间”命令，然后通过新的编译连接，产生第二个程序的工作空间，否则将一直运行前一个程序。

## 二、编译、连接、运行程序(2)

实验步骤如下：

步骤一：输入下列程序，编译、连接并运行程序。

步骤二：观察程序的输出结果。

步骤三：对程序做适当改变，再观察显示的内容。

```
#include<stdio.h>      //预处理命令
void main()            //主函数定义
{
    printf("    输入学号： \n");
    printf("===== \n");
    printf("    输入姓名： \n");
    printf("===== \n");
    printf("    输入您的成绩： \n");
}
```

运行结果见图 1-17。

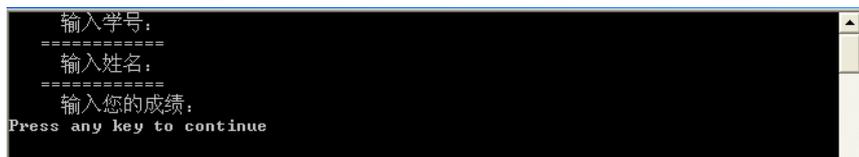


图 1-17 运行结果

### 三、编译、连接、运行程序(3)

实验步骤如下：

步骤一：输入以下程序，编译、连接并运行程序。

步骤二：观察所显示的内容。

步骤三：对程序做适当改变，再观察显示的内容。

```
#include<stdio.h>

void main( )
{ printf("..... \n");
  printf("   ===电话订餐信息处理系统=== \n");
  printf(" \n");
  printf("  1---用户身份验证 \n");
  printf(" \n");
  printf("  2---查询 \n");
  printf(" \n");
  printf("  3---修改 \n");
  printf(" \n");
  printf("  4---显示 \n");
  printf(" \n");
  printf("  5---退出 \n");
  printf("..... \n");
}
```

运行结果见图 1-18。



```
.....
===电话订餐信息处理系统===

1---用户身份验证

2---查询

3---修改

4---显示

5---退出
.....
Press any key to continue_
```

图 1-18 运行结果

## 四、编译、连接、运行程序(4)

实验步骤如下：

步骤一：输入以下程序，编译、连接并运行程序。

步骤二：观察输出结果。

步骤三：对程序做适当改变，再观察显示的内容。

```
#include<stdio.h>

void main( )
{
    int a=3,b=4;           //定义整型变量 a, b, 值分别是 3 和 4
    printf("%d,%d\n",a,b); //按要求格式输出数据
    printf("a=%d,b=%d\n",a,b); //按要求格式输出数据
}
```

运行结果见图 1-19。

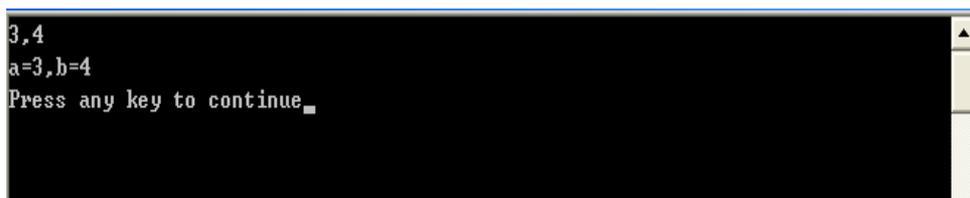


图 1-19 运行结果

**实验思考：**

① 删除“`,b=4`”，再编译程序，阅读错误提示，并根据错误提示进行修改。作为程序设计者，应该具备根据错误提示或程序运行结果，查找并修正程序中错误的能力。

② 删除“`=4`”，编译并运行程序，观察程序的运行结果，并思考其原因。

③ `printf( )` 函数所要显示的信息不仅可以是一个字符串，而且还可以是变量的值或数据。更有趣的是，它可以按照用户定义的某种格式输出。

④ 读者自己编写一个程序，要求可以在屏幕上分行显示出自己的姓名、学号、所在院系、所学课程和 E-mail 地址，发挥想象，使显示形式美观而不单调。

⑤ 读者自己编写程序，输出一个由“\*”组成的菱形。

**注意：**

在进行程序设计时，应养成良好的程序设计风格：

① 一般一个语句占一行。

② 在程序中添加适当的注释。

③ 按照缩进格式书写程序。

## 单元测试

在 Visual C++6.0 开发环境中，编辑并运行以下代码，查看结果。

```
#include<stdio.h>
main()
{
    printf("*****\n");
    printf("C Program!\n");
    printf("*****\n");
}
```

参考答案：

运行结果见图 1-20。



```
*****
      C Program!
*****
Press any key to continue_
```

图 1-20 单元测试运行结果

