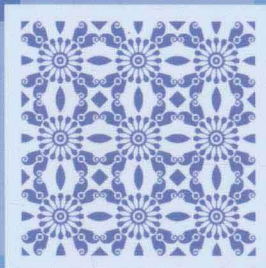


# C语言程序设计 习题解析与上机指导 第2版

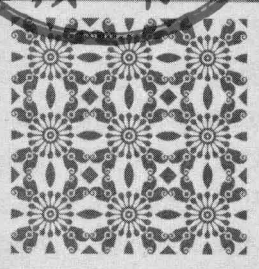
罗晓芳 李慧 孙涛 孙大为 朱鸣华 杨微 编著



高等院校计算机教材系列

# C语言程序设计 习题解析与上机指导 第2版

罗晓芳 李慧 孙涛 孙大为 朱鸣华 杨微 编著



机械工业出版社  
China Machine Press

## 图书在版编目 (CIP) 数据

---

C语言程序设计习题解析与上机指导 / 罗晓芳等编著. —2版. 北京: 机械工业出版社, 2013.12  
(高等院校计算机教材系列)

ISBN 978-7-111-44993-5

I. C… II. 罗… 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 (2013) 第286813号

---

### 版权所有·侵权必究

封底无防伪标均为盗版

本书法律顾问 北京市展达律师事务所

本书配合主教材, 根据教学要求共设置了4部分的内容, 包括: C语言编程环境简介、知识提要及习题解析、上机指导及实验、图形程序使用基础。本书对主教材各章后面练习中的问题进行了详细解析, 同时还汇总了C语言程序设计课程学习中的疑难问题和常见问题的解析。精心设置了9个上机实验, 每个实验项目都给出了实验目的和要求, 并给出了编程的示例和练习题目, 读者可以通过实际训练中的由浅入深式学习, 逐步熟悉编程环境, 掌握程序调试方法, 理解和掌握程序设计的思想、方法和技巧。

本书适合各类高等院校C语言程序设计课程的实验教学, 同时也适合作为学习C语言的指导教材。

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: 刘立卿

三河市杨庄长鸣印刷装订厂印刷

2014年1月第2版第1次印刷

185mm × 260mm · 9.75印张

标准书号: ISBN 978-7-111-44993-5

定 价: 25.00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

投稿热线: (010) 88379604

购书热线: (010) 68326294 88379649 68995259

读者信箱: hzjsj@hzbook.com

## 第2版前言

《C语言程序设计习题解析与上机指导》教材于2009年2月出版以来多次重印，它配合主教材的使用，在教学和实践中起到了较好的辅助作用。

第2版在第1版的基础上对全书进行了修订，删除了Turbo C环境介绍的内容，兼顾Visual C++ 6.0系统环境，规范了程序的书写格式，补充了部分试题，第四部分内容介绍Visual C++ 6.0环境下绘图的基本方法，同时，修正和完善了教材中文字叙述的部分内容。

书中的错误和不足之处，敬请读者指正。

编者

2013年11月

## 第1版前言

C语言程序设计是一门实践性很强的课程，该课程的学习有其自身的特点，学习者必须通过大量的上机编程训练，在实践中逐步掌握程序设计的方法。本书是与朱鸣华、刘旭麟、杨微等编著的《C语言程序设计教程》配套使用的习题解答与上机指导教材。

本书内容从教学实际需要出发，兼顾不同学生的计算机实际水平，配合主教材的内容，对课程中的主要知识点进行归纳和总结，对各章习题中的疑难问题和常见问题进行详细解析。本教材还安排了循序渐进的上机训练，使学生能自己进行有目的的实验训练，通过在实际训练中由浅入深地学习，逐步熟悉编程环境、程序调试方法，更好地理解和掌握程序设计的思想、方法和技巧。

全书配合主教材，根据教学要求共设置了四部分内容：第一部分主要介绍计算机程序设计实验的一般方法和步骤、Visual C++ 6.0上机指南、Turbo C 2.0上机指南以及具体调试程序的方法和步骤；第二部分主要对配套教材各章后习题中的问题进行详细解析，同时对C语言程序设计课程学习中的疑难问题和常见问题进行解析；第三部分精心设置9个上机实验，每个实验项目都给出实验目的和要求，并给出编程的示例和练习题，读者可以先阅读、模仿示例练习，然后独立完成上机练习题；第四部分给出几个常用的程序设计实例，对学习内容进行一定的拓展。

本书叙述通俗易懂，逻辑性强，实验内容设置合理，既方便教师教也方便学生学习。考虑到目前“C语言程序设计”课程采用了Visual C++ 6.0测试环境，本书还介绍了Visual C++ 6.0编程环境的使用方法。本书中的实例都在Turbo C 2.0环境下测试通过，对于部分编程示例还给出了N-S流程图表示。本书适合各类高等院校C语言程序设计课程的实验教学，也适合作为学习C语言的指导教材。

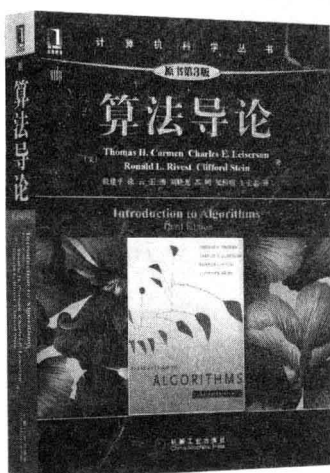
本书由朱鸣华、杨微统稿。其中，第1~5章、实验1和实验2由罗晓芳编写；第6章和实验3由李慧编写；第7章、第11章、实验4、实验7和第四部分由孙涛编写；第8章、第13章、实验5和实验9由孙大为编写；第9章和实验6由朱鸣华编写；第10章、第12章和实验8由杨微编写。

由于编写时间仓促，书中难免有错误和不足之处，敬请读者指正。

编者

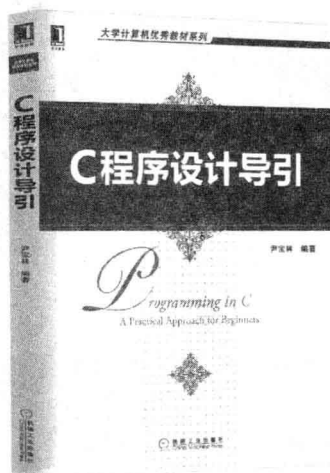
2008年12月

## 推荐阅读



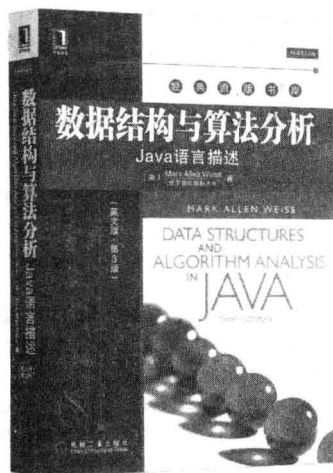
### 算法导论 (原书第3版)

作者: Thomas H.Cormen 等 ISBN: 978-7-111-40701-0 定价: 128.00元



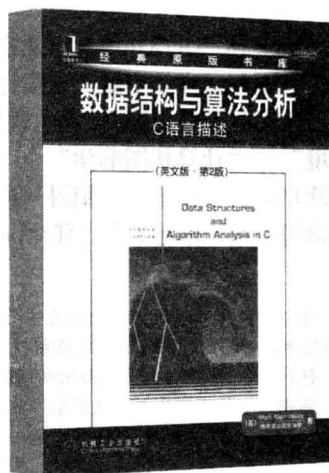
### C程序设计导引

作者: 尹宝林 ISBN: 978-7-111-41891-7 定价: 35.00元



### 数据结构与算法分析 ——Java语言描述 (英文版·第3版)

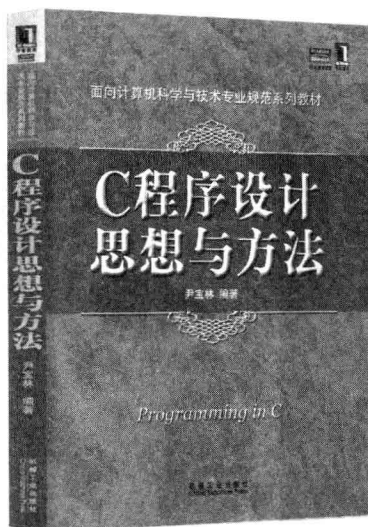
作者: Mark Allen Weiss ISBN: 978-7-111-41236-6 定价: 79.00元



### 数据结构与算法分析 ——C语言描述(英文版·第2版)

作者: Mark Allen Weiss ISBN: 978-7-111-31280-2 定价: 45.00元

# 推荐阅读



## C程序设计思想与方法

作者：尹宝林 ISBN: 978-7-111-25495-9 定价：36.00元

### 作者简介

**尹宝林** 1973年毕业于北京航空学院计算机专业，1984年获英国爱丁堡大学博士学位。现任北京航空航天大学计算机系教授、博士生导师。从事计算机专业教学和科研工作多年，曾主讲“C语言程序设计”、“高级语言程序设计”、“UNIX程序设计环境”、“计算机图形学”、“图像处理”等课程，其中“高级语言程序设计”被评为北京市精品课程。主编过《离散数学》等教材。参加全国信息学奥林匹克（NOI）活动的组织与指导工作，任全国信息学奥林匹克科学委员会副主席。

本书围绕程序设计的思想和方法组织教学内容，把C语言作为讲解程序设计的工具，把教学的重点从C语言本身转移到程序设计的思路和方法以及使用C语言编程上来，把讲授语言的概念转变为讲授语言的运用，把对概念的讲授由抽象的分析和讲解转变为通过实例和经验的学习和运用。

本书作者根据学生对新知识学习、认知的规律，从C语言和C程序的基本要素以及程序设计的基本方法开始，循序渐进地引入对程序设计专业化的要求和相关的知识；从增强学生的感性认识入手，通过多角度的对例题的分析，示范对关键知识和技术的运用，通过对关键内容在不同层次上的适当重复，深化学生对概念的理解和掌握。本书精选的例题和练习题，在功能、知识和技术上覆盖了从语言入门到实用工具多个层面，有助于学生掌握专业化的编程技能，养成专业化的编程习惯。

本书作者专门设计和实现了基于Web的C程序练习/评测系统，为学生提供具有及时监控和反馈环节的实践环境，为本书的练习题提供联网的实时评测。

# 目 录

第2版前言

第1版前言

## 第一部分 C语言编程环境简介

### 第1章 计算机程序设计实验的一般步骤 .....1

- 1.1 计算机程序设计实验的目的 .....1
- 1.2 计算机程序设计实验的准备 .....1
- 1.3 计算机程序设计实验的步骤 .....1

### 第2章 Visual C++ 6.0上机指南 .....3

- 2.1 Visual C++ 6.0概述 .....3
- 2.2 使用Visual C++ 6.0建立C语言应用程序 .....3
- 2.3 快速建立简单的C程序 .....10
- 2.4 Visual C++ 6.0的C语言应用程序文件结构概述 .....10

## 第二部分 知识提要及习题解析

### 第3章 数据类型、运算符与表达式 .....13

- 3.1 本章知识重点 .....13
- 3.2 习题解析 .....14
  - 3.2.1 计算题 .....14
  - 3.2.2 选择题 .....15

### 第4章 数据的输入和输出 .....18

- 4.1 本章知识重点 .....18
- 4.2 习题解析 .....18
  - 4.2.1 选择题 .....18
  - 4.2.2 完善题 .....19
  - 4.2.3 阅读程序题 .....20

### 第5章 选择结构 .....21

- 5.1 本章知识重点 .....21
- 5.2 习题解析 .....22
  - 5.2.1 判断题 .....22

5.2.2 选择题 .....23

5.2.3 完善题 .....26

5.2.4 阅读程序题 .....28

### 第6章 循环结构 .....30

- 6.1 本章知识重点 .....30
- 6.2 习题解析 .....31
  - 6.2.1 判断题 .....31
  - 6.2.2 选择题 .....31
  - 6.2.3 完善题 .....33
  - 6.2.4 阅读程序题 .....36

### 第7章 数组 .....39

- 7.1 本章知识重点 .....39
- 7.2 习题解析 .....41
  - 7.2.1 判断题 .....41
  - 7.2.2 选择题 .....42
  - 7.2.3 完善题 .....45
  - 7.2.4 阅读程序题 .....48

### 第8章 函数 .....52

- 8.1 本章知识重点 .....52
- 8.2 习题解析 .....53
  - 8.2.1 判断题 .....53
  - 8.2.2 选择题 .....55
  - 8.2.3 完善题 .....58
  - 8.2.4 阅读程序题 .....60

### 第9章 编译预处理 .....65

- 9.1 本章知识重点 .....65
- 9.2 习题解析 .....66
  - 9.2.1 判断题 .....66
  - 9.2.2 选择题 .....66
  - 9.2.3 完善题 .....67
  - 9.2.4 阅读程序题 .....68





# 第一部分 C语言编程环境简介

## 第1章 计算机程序设计实验的一般步骤

### 1.1 计算机程序设计实验的目的

C语言程序设计是一门实践性很强的课程，该课程的学习有其自身的特点，学习者必须通过大量的编程训练，在实践中掌握程序设计语言，培养程序设计的基本能力，并逐步理解和掌握程序设计的思想和方法。具体地说，通过上机实践，应该达到以下几点要求：

1) 使学习者能很好地掌握一种程序设计开发环境的基本操作方法（例如Visual C++ 6.0），掌握应用程序开发的一般步骤。

2) 在程序设计和程序调试的过程中，学习者可以进一步理解教材中各章节的主要知识点，特别是一些语法规则的理解和运用，程序设计中的常用算法和构造及应用，也就是所谓“在编程中学习编程”。

3) 通过上机实践，提高程序分析、程序设计和程序调试的能力。程序调试是一个程序员最基本的技能，不会调试程序的程序员就意味着他即使会一门语言，也不能编制出任何好的软件。程序员通过不断地积累经验，摸索各种比较常用的技巧，可以提高编程的效率和所编程序代码的质量。

下面介绍计算机程序设计实验的准备工作 and 一般步骤。

### 1.2 计算机程序设计实验的准备

上机前需要做好如下准备工作，以提高上机编程的效率：

- 1) 在计算机上安装一种程序设计开发工具，并学会基本的操作方法。
- 2) 复习与本次实验相关的教学内容和主要知识点。
- 3) 准备好编程题程序流程图和全部源程序代码，并且先进行人工检查。
- 4) 对程序中有疑问的地方做出标记，充分估计程序运行中可能出现的问题，以便在程序调试过程中给予关注。
- 5) 准备好运行和调试程序所需的数据。

### 1.3 计算机程序设计实验的步骤

计算机程序设计实验一般有如下几个步骤：

- 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出错原因，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。
- 7) 实验后，应提交实验报告，主要内容应包括程序清单、调试数据和运行结果，还应该包括对运行结果的分析 and 评价等内容。

## 第2章 Visual C++ 6.0上机指南

### 2.1 Visual C++ 6.0概述

Visual C++ 6.0是微软公司推出的目前使用极为广泛的基于Windows平台的可视化集成开发环境。它和Visual Basic、Visual Foxpro、Visual J++等其他软件构成了Visual Studio (又名Developer Studio) 程序设计软件包。Visual Studio是一个通用的应用程序集成开发环境, 包含了文本编辑器、资源编辑器、工程编译工具、增量连接器、源代码浏览器、集成调试工具以及一套联机文档。使用Visual Studio, 可以完成创建、调试、修改应用程序等各种操作。

Visual C++ 6.0提供面向对象技术的支持, 它能够帮助使用MFC库的用户自动生成一个具有图形界面的应用程序框架。用户只需在该框架的适当部分添加、扩充代码就可以得到一个满意的应用程序。

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, 大大减轻程序设计的劳动强度, 提高程序设计的效率。

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。

### 2.2 使用Visual C++ 6.0建立C语言应用程序

利用Visual C++ 6.0提供的一种控制台操作方式, 可以建立C语言应用程序, Win32控制台程序 (Win32 Console Application) 是一类Windows程序, 它不使用复杂的图形用户界面, 程序与用户交互是通过一个标准的正文窗口, 下面我们将对使用Visual C++ 6.0编写简单的C语言应用程序做一个初步的介绍。

#### 1. 安装和启动

运行Visual Studio软件中的setup.exe程序, 选择安装Visual C++ 6.0, 然后按照安装程序的指导完成安装过程。

安装完成后, 在开始菜单的程序选单中有Microsoft Visual Studio 6.0图标, 选择其中的Microsoft Visual C++ 6.0即可运行 (也可在Window桌面上建立一个快捷方式, 以后双击即可运行)。

#### 2. 创建工程项目

用Visual C++ 6.0系统建立C语言应用程序, 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:

1) 进入Visual C++ 6.0环境后, 选择主菜单“文件”中的“新建”命令, 在弹出的对话框中单击上方的选项卡“工程”, 选择“Win32 Console Application”工程类型, 在“工程”一栏中填写工程名, 例如“Myexam1”, 在“位置”一栏中填写工程路径 (目录), 例如“D:\MyProject\Myexam1\”, 见图2-1, 然后单击“确定”按钮继续。

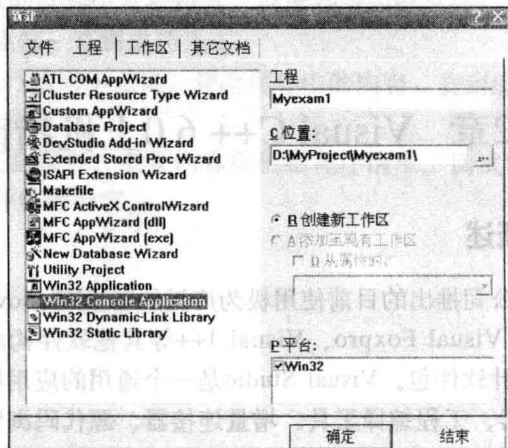


图2-1 创建工程项目

2) 屏幕上出现如图2-2所示的“Win32 Console Application - Step 1 of 1”对话框后, 选择“An empty project”项, 然后单击“完成”按钮继续。

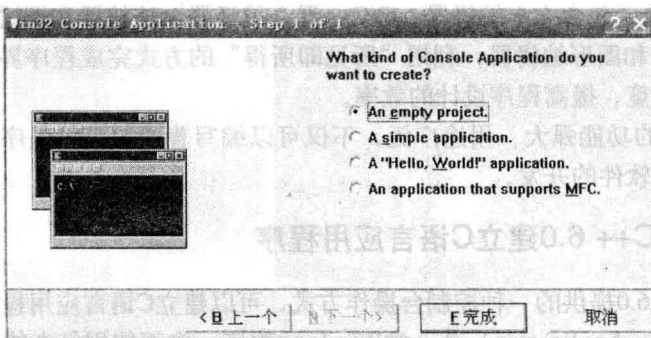


图2-2 “Win32 Console Application - Step 1 of 1”对话框

出现如图2-3所示的“新建工程信息”对话框后, 单击“确定”按钮完成工程创建。创建的工作区文件为“myexam1.dsw”, 工程项目文件为“myexam1.dsp”。

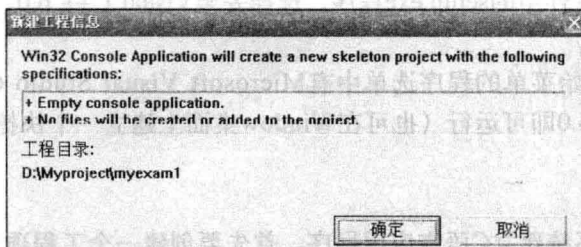


图2-3 “新建工程信息”对话框

### 3. 新建C源程序文件

有两种激活命令的方法:

- 1) 选择主菜单“工程”中的“添加工程→新建”命令;
- 2) 选择主菜单“文件”中的“新建”命令。

屏幕上出现“新建”对话框后, 选择“文件”选项卡, 选定“C++ Source File”项, 在

“文件”一栏中填入新添加的源文件名，如“myexam1.c”，在“目录”一栏中指定文件路径，例如“D:\Myproject\myexam1”，单击“确定”按钮完成C源程序的系统新建操作，如图2-4所示。

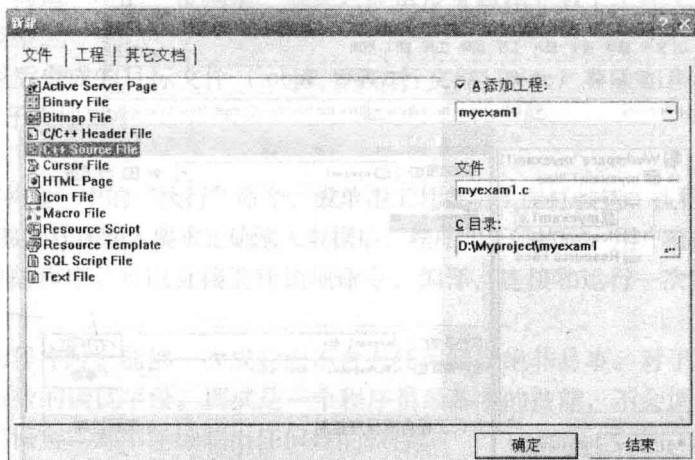


图2-4 加入新的C源程序文件

在文件编辑区输入源程序，然后保存工作区文件，如图2-5所示。

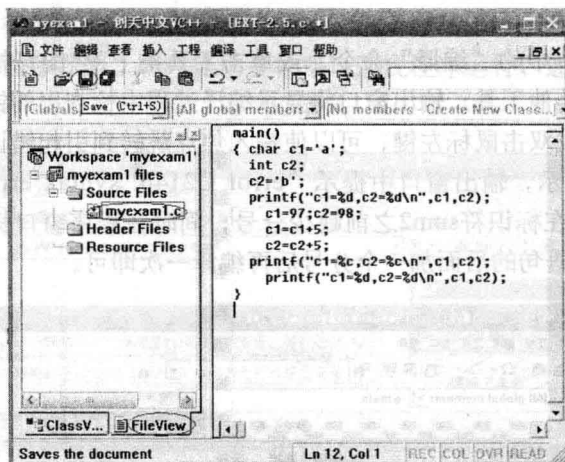


图2-5 建立C源程序

**注意：**填入C源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”。

#### 4. 打开已存在的工程项目，编辑工程项目内的C源程序

进入Visual C++ 6.0环境后，选择主菜单“打开工作区”命令，在“Open Workspace”对话框内找到并选择要打开的工作区文件“myexam1.dsw”，单击“确定”按钮，打开工作区。

在左侧的工作区窗口，单击下方的“FileView”选项卡，选择文件视图显示，打开“Source Files”文件夹，再打开要编辑的C源程序进行编辑和修改，如图2-6所示。

#### 5. 在工程项目中添加已经存在的C源程序文件

选择主菜单“打开工作区”命令，在“Open Workspace”对话框内找到并选择要打开的工作区文件“myexam.dsw”，单击“确定”按钮打开工作区。

将已经存在的C源程序文件添加工程到当前打开的工程区文件中，选择主菜单“工程”中

的“添加工程→文件”选项，在“Insert File into Project”对话框内找到已经存在的C源程序文件，单击“确定”按钮完成添加。



图2-6 打开myexam1.c源程序

## 6. 编译、连接和运行

### (1) 编译



选择主菜单“构建”中的“编译”命令，或单击工具条上图标按钮，系统只编译当前文件而不调用连接器或其他工具。输出窗口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大致位置以进行修改。如图2-7所示，输出窗口中提示“error C2146: syntax error : missing ';' before identifier 'sum2'”，提示在标识符sum2之前缺少分号，同时在程序窗口标注出出错语句的大致位置。在“sum1=b-a”语句的后面加一个分号后再编译一次即可。



图2-7 编译、连接和运行.c源程序

### (2) 构建


选择主菜单“构建”中的“构建”命令，或单击工具条上图标按钮，对最后修改过的源文件进行编译和连接。

选择主菜单“构建”中的“重建全部”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“构建”中的“批构建”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。

程序构建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。

### (3) 运行

选择主菜单“构建”中的“执行”命令，或单击工具条上图标按钮，执行程序，将会出现一个新的用户窗口，按照程序输入要求正确输入数据后，程序即正确执行，用户窗口显示运行的结果。

对于比较简单的程序，可以直接选择该项命令，编译、连接和运行一次完成。

### 7. 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，却不能编制出任何好的软件。

#### (1) 调试程序环境介绍

##### 1) 进入调试程序环境

选择主菜单“构建”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”工具条上的图标按钮，可以打开/关闭这些窗口，参考图2-8。

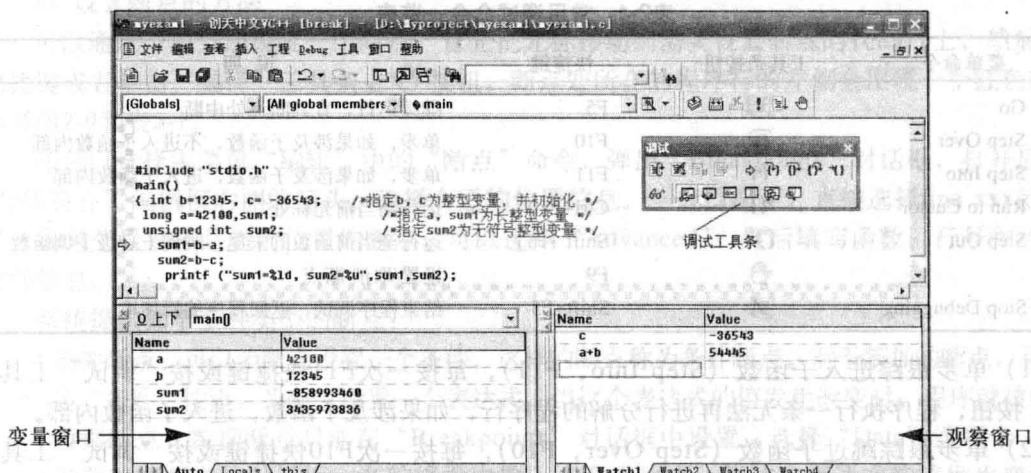



图2-8 调试程序界面

##### 2) “Watch”（观察）窗口

单击“调试”工具条上按钮，下方出现一个观察窗口。

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是在断点中断的情况下进行。

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


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：表达式不能有



副作用，例如“++”和“--”运算符绝对禁止用于表达式中，因为这两个运算符将修改变量的值，导致程序的逻辑被破坏。

还可以采用系统提供了一种称为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”命令，弹出一个对话框，显示这个变量的值。

### 3) “Variables”（变量）窗口

单击“调试”工具条上按钮弹出变量窗口，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）

对于指针指向的数组，“Watch”窗口只能显示数组第一个元素的值。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”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”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”工具条上的“Call stack”按钮，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。


## (2) 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”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。表2-1中列出了常用的调试命令。


表2-1 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

1) 单步跟踪进入子函数（Step Into, F11），每按一次F11快捷键或按“调试”工具条上按钮，程序执行一条无法再进行分解的程序行，如果涉及子函数，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快捷键或按“调试”工具条上按钮，程序执行一行；观察窗口可以显示变量名及其当前值，在单步执行的过程中，可以在观察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Shift+F11组合键或按“调试”工具条上按钮后，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（Run to Cursor, Ctrl+F10），当按下Ctrl+F10组合键或按“调试”工具条上按钮后，程序运行至当前光标处所在的语句。

## (3) 设置断点调试程序

对于较大规模程序的跟踪，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