



ZUIXIN C YUYAN

最新

语言程序设计教程 习题详解与上机实验

(第2版)

CHENGXU

SHEJI

JIAO CHENG

XITI XIANGJIE YU SHANGJI SHIYAN

刘正林 郭胜 丁玮 陈维

编

著

华中科技大学出版社

- 配套教材的习题详解
- 丰富的范例程序
- 深入浅出、图文并茂地讲述，特别适合独立学院的大专生和本科生

最新 C 语言程序设计教程 习题详解与上机实验 (第 2 版)

刘正林 郭 胜 编著
丁 玮 陈 维

华中科技大学出版社

中国·武汉

图书在版编目(CIP)数据

最新C语言程序设计教程习题详解与上机实验(第2版)/刘正林 郭胜 丁玮 陈维 编著
武汉:华中科技大学出版社,2006年9月

ISBN 7-5609-2680-0

I. 最…

Ⅱ. ①刘… ②郭… ③丁… ④陈…

Ⅲ. C语言-程序设计-高等学校-教学参考资料

Ⅳ. TP312

最新C语言程序设计教程
习题详解与上机实验(第2版)

刘正林 郭胜 编著
丁玮 陈维

责任编辑:王汉江

封面设计:刘卉

责任校对:陈骏

责任监印:张正林

出版发行:华中科技大学出版社

武昌喻家山

邮编:430074

电话:(027)87557437

录排:华中科技大学惠友文印中心

印刷:湖北省通山县九宫印务有限公司

开本:787×1092 1/16

印张:18

字数:406 000

版次:2006年9月第2版

印次:2006年9月第5次印刷

定价:29.80元

ISBN 7-5609-2680-0/TP·463

(本书若有印装质量问题,请向出版社发行部调换)

内 容 简 介

本书是《最新 C 语言程序设计教程（第 2 版）》教材的配套书，不仅给出了教材的各章习题的详细解答，并通过剖析习题内容归纳总结编程要点，书中所给出的每个程序都有详细的注释，便于阅读和理解，并且都在 Visual C++ 6.0 中完成了调试且能成功地运行。其特点是通过实用化例程让读者全面、灵活、系统地掌握面向对象程序设计的技术思路 and 开发方法，并用“深入浅出，通俗易懂”的方式讲解令人眼花缭乱的编程技术。

本书适用于大专院校理、工科各类专业大专生和本科生，特别是独立学院的学生。遵循 ISO/ANSI C++ 标准讲述 C++ 模块化的编程技术，是快速过渡到 C++ 的最佳途径。在内容安排上有深有浅，且侧重应用，适用于各个层次的读者，既适合以前从未接触过 C++ 的初学者，也可以作为具有一定编程能力的读者提高编程能力、快速向 C++ 过渡的参考书。

前 言

“C 语言程序设计”是一门实践性非常强的课程，若要真正掌握编程技术，使编程能力有实质性的提高，必须在认真听课并读懂教材内容的基础上，通过上机实验加强开发软件的基本技能训练。只有勤学苦练才能积累宝贵的编程经验，悟出编程技术的要领，牢固地掌握像 Borland C++和 Visual C++这样的优秀应用程序开发工具，快速冲向计算机应用领域的前沿。

在学习这门课程时，很多学生都提出这样的困惑：课堂和教材的内容基本上都能接受和理解，但真要自己实际动手编程却又感到脑袋一片空白而无从下手；其二，即便学完了本课程的所有内容，对编写实用化的应用程序，总是感到眼花缭乱而无从下手。特别是对独立学院的学生来说，究竟该如何提高自身阅读程序和编写程序的能力，才能达到全国计算机等级考试“C 语言”科目的要求呢？问题的关键在于学习方法：读者在理解教材基本内容的基础上，要注重查找、收集和阅读“经典”的算法程序，并通过上机实验利用调试功能透彻理解这些经典算法程序的编程思路。例如，教材中所讲述的冒泡排序、插入排序和折半查找等算法均是作者精心收集的这类经典算法程序。碰到需要编写实用化的应用程序时，要有目的地通过 Internet 网、计算机杂志和资料等去查找现存的、成熟的相关程序。

本书是《最新 C 语言程序设计教程（第 2 版）》教材的配套书，遵循 ISO/ANSI C++ 标准讲述 C++模块化的编程技术，是快速过渡到 C++的最佳途径。为初学者培养编程能力、加强开发软件的基本技能训练提供必要的辅导材料，对教材中各章习题均进行了详细地解答。不仅给出该题的正确答案，而且还通过剖析习题内容，归纳总结技术要点，使学生知其然且知其所以然，并引导学生举一反三，扎实、灵活和系统地掌握编程要点。书中所给出的每一个程序均都有详细的注释，便于学生阅读和透彻理解编程思路，并且都在 Visual C++ 6.0 集成开发环境中完成了调试且能成功地运行。其特点是通过实用化例程让读者全面、灵活、系统地掌握程序设计的技术思路 and 开发方法，并用“深入浅出，通俗易懂”的方式讲解 C 语言编程技术，在教给学生“面包加糖果”之类知识的同时，更重要的是要让他们掌握“猎枪和钓鱼杆”这类工具的使用方法，学会了“猎枪和钓鱼杆”这类生存工具的使用方法，学生就有了独立探索新知识的本领，这才是事半功倍的学习方法。

本书适用于大专院校理、工科各类专业大专生和本科生，特别是独立学院的学生，在内容安排上有深有浅，且侧重应用。

衷心感谢读者选用本书，欢迎批评指正。

华中科技大学文华学院
刘正林 郭 胜 丁 玮 陈 维
2006 年 8 月

目 录

第 1 章 概论.....	(1)
第 2 章 数据类型、运算符和表达式.....	(12)
第 3 章 语句和流程控制.....	(25)
第 4 章 数组与指针.....	(62)
第 5 章 函数.....	(88)
第 6 章 复合数据类型和函数调用.....	(142)
第 7 章 结构体.....	(193)
第 8 章 标准函数.....	(228)
附录 1 Visual C++ 6.0 常见编译和链接错误.....	(249)
附录 2 Borland C++ 3.1 使用方法.....	(255)
参考文献.....	(282)

第1章 概 论

习 题 1

一、计算题

1. 将如下十进制数转换成 8bits 的无符号二进制数。

说明 4 位二进制数可以用一位十六进制数表示, 因此, 可以说十六进制数是二进制数的简洁记法, 按教材表 1.1 所列的十六进制数和二进制数的对应关系, 则把十进制数转换成二进制数的问题可借用十六进制数简化十翻二运算, 即十翻二运算可以先把十进制数转换成十六进制数, 然后, 从最高位开始 (从左至右) 把每一位十六进制数利用教材表 1.1 的对应关系转换成 4 位二进制, 这样计算不仅大大简化了计算步骤且不容易出错。

(1) 127

$$\begin{aligned} \text{解 } 127 &= 7 \quad \text{F} \quad (\text{H}) \\ &\downarrow \quad \downarrow \quad \text{十六进制数转换成二进制数} \\ &= 0111, 1111 \end{aligned}$$

$7 \times 16 + 15 = 127$
0 1 2 3 4 5 6 7 8 9
A B C D E F

(2) 55

$$\begin{aligned} \text{解 } 55 &= 3 \quad 7 \quad (\text{H}) \\ &\downarrow \quad \downarrow \quad \text{十六进制数转换成二进制数} \\ &= 0011, 0111 \end{aligned}$$

$16 \times 3 + 7 = 55$

(3) 255

$$\begin{aligned} \text{解 } 255 &= \text{F} \quad \text{F} \quad (\text{H}) \\ &\downarrow \quad \downarrow \quad \text{十六进制数转换成二进制数} \\ &= 1111, 1111 \end{aligned}$$

(4) 128

$$\begin{aligned} \text{解 } 128 &= 8 \quad 0 \quad (\text{H}) \\ &\downarrow \quad \downarrow \quad \text{十六进制数转换成二进制数} \\ &= 1000, 0000 \end{aligned}$$

2. 将如下十进制数转换成 4 位十六进制数, 并按教材表 1.1 写成对应的 16bits 的无符号二进制数。

(1) 32767

$$\begin{aligned} \text{解 } 32767 &= 7 \quad \text{F} \quad \text{F} \quad \text{F} \quad (\text{H}) \\ &\downarrow \quad \downarrow \quad \downarrow \quad \downarrow \quad \text{十六进制数转换成二进制数} \\ &= 0111, 1111, 1111, 1111 \end{aligned}$$

(2) 65535

$$\begin{aligned} \text{解 } 65535 &= \begin{array}{cccc} \text{F} & \text{F} & \text{F} & \text{F(H)} \\ \downarrow & \downarrow & \downarrow & \downarrow \end{array} \text{十六进制数转换成二进制数} \\ &= 1111, 1111, 1111, 1111 \end{aligned}$$

(3) 2554

$$\begin{aligned} \text{解 } 2554 &= \begin{array}{cccc} 0 & 9 & \text{F} & \text{A(H)} \\ \downarrow & \downarrow & \downarrow & \downarrow \end{array} \text{十六进制数转换成二进制数} \\ &= 0000, 1001, 1111, 1010 \end{aligned}$$

(4) 64128

$$\begin{aligned} \text{解 } 64128 &= \begin{array}{cccc} \text{F} & \text{A} & 8 & 0(\text{H}) \\ \downarrow & \downarrow & \downarrow & \downarrow \end{array} \text{十六进制数转换成二进制数} \\ &= 1111, 1010, 1000, 0000 \end{aligned}$$

3. 求如下整数的补码, 分别用 8bits 二进制数和两位十六进制数表示。

说明 二进制运算法则规定: 一个正整数的补码就是它的原码, 一个负整数的补码是把它绝对值 (肯定是一个正整数) 变成二进制数后再变反加 1 即变补。

(1) -128

$$\begin{aligned} \text{解 } (-128)_{\text{补}} &= (-|-128|)_{\text{补}} = (-(128)_{\text{原}})_{\text{补}} \\ &= (1000, 0000)_{\text{补}} = 1000, 0000 \\ &\quad \uparrow \qquad \qquad \qquad \uparrow \\ &\quad \text{8 位二进制最高位 } 2^7 \qquad \text{符号位} \end{aligned}$$

(-128) 的绝对值 128 已不在带符号 8 位二进制数的数值范围 $(-128 \sim 127)$ 内, 因此, 把绝对值 128 变成二进制数, 其最高位 b_7 已不是符号位而是数值 $2^7 = 128$ 。求 128 的补码即将二进制码 $(1000, 0000)$ 变反加 1, 按二进制运算法则所得到的二进制码 $1000, 0000$ 是补码形式, 其最高位 b_7 是符号位, 它为 1 表示结果值是负整数。

(2) -1

$$\begin{aligned} \text{解 } (-1)_{\text{补}} &= (-|-1|)_{\text{补}} = (-(1)_{\text{原}})_{\text{补}} \\ &= (0000, 0001)_{\text{补}} = 1111, 1111 \\ &\quad \qquad \qquad \qquad \uparrow \\ &\quad \qquad \qquad \text{符号位} \end{aligned}$$

(3) -2

$$\begin{aligned} \text{解 } (-2)_{\text{补}} &= (-|-2|)_{\text{补}} = (-(2)_{\text{原}})_{\text{补}} \\ &= (0000, 0010)_{\text{补}} = 1111, 1110 \\ &\quad \qquad \qquad \qquad \uparrow \\ &\quad \qquad \qquad \text{符号位} \end{aligned}$$

(4) -127

$$\begin{aligned} \text{解 } (-127)_{\text{补}} &= (-|-127|)_{\text{补}} = (-(127)_{\text{原}})_{\text{补}} \\ &= (0111, 1111)_{\text{补}} = 1000, 0001 \\ &\quad \qquad \qquad \qquad \uparrow \\ &\quad \qquad \qquad \text{符号位} \end{aligned}$$

4. 用 8bits 补码相加实现如下运算。

(1) $-18 + 6$

解 $-18 + 6 = (-18) + 6 = 1111, 0100 = -12$

计算过程如下。

$$\begin{array}{r} (-18)_{补} = (-(18)_{原})_{补} = (0001, 0010)_{补} = 1110, 1110 \\ +) \quad (6)_{补} = 0000, 0110 \\ \hline 1111, 0100 \end{array}$$

(2) $-55 - 64$

解 $-55 - 64 = (-55) + (-64) = 1000, 1001 = -119$

计算过程如下。

$$\begin{array}{r} (-55)_{补} = (-(55)_{原})_{补} = (0011, 0111)_{补} = 1100, 1001 \\ (-64)_{补} = (-(64)_{原})_{补} = (0100, 0000)_{补} = 1100, 0000 \\ \hline 1000, 1001 \end{array}$$

(3) $2 - 128$

解 $2 - 128 = 2 + (-128) = 1000, 0010 = -126$

计算过程如下。

$$\begin{array}{r} (2)_{补} = 0000, 0010 \\ (-128)_{补} = (-(128)_{原})_{补} = (1000, 0000)_{补} = 1000, 0000 \\ \hline 1000, 0010 \end{array}$$

(4) $-120 + 7$

解 $-120 + 7 = (-120) + 7 = 1000, 1111 = -113$

计算过程如下。

$$\begin{array}{r} (-120)_{补} = (-(120)_{原})_{补} = (0111, 1000)_{补} = 1000, 1000 \\ +) \quad (7)_{补} = 0000, 0111 \\ \hline 1000, 1111 \end{array}$$

二、选择题

1. 下列各种高级语言中, (D) 是面向对象的程序设计语言。

[1] A) BASIC B) PASCAL C) C++ D) Java

解 只有 Java 才是完全面向对象的程序设计语言。C++是既可以做面向过程的模块化程序设计, 又支持面向对象程序设计的混合型语言。美国 Microsoft 公司把模块化、可视化和面向对象程序设计等现代软件新技术融入到 BASIC 中, 将它改造成 Visual BASIC, 还有美国 Borland 公司将面向对象思想引入到 PASCAL, 推出了面向对象开发软件包

Delphi PASCAL, 但严格来讲, 它们都不是面向对象的程序设计语言。

2. 关于 C++ 和 C 语言的关系描述中, (D) 是错误的。

- [2] A) C 语言是 C++ 的一个子集 B) C 语言和 C++ 是兼容的
C) C++ 对 C 语言进行了一些改进 D) C 语言和 C++ 都是面向对象的

解 参见教材的图 1.9, C 语言蕴涵在 C++ 中, 是它的一个子集, 即它的模块化语言基础的内核所在, 换句话说, C++ 是 C 语言的超集。在符合最新 ANSI C++ 标准的前提下, C 语言和 C++ 是兼容的, 具体来说, 在 Visual C++ 集成开发环境下, 在以 “.cpp” 为扩展名的源文件中, 既可以用模块化的 C 语言 (不符合 ISO/ANSI C++ 标准的 C 语言老版本除外) 格式编程, 也可以采用面向对象风格的 C++ 格式编程。C++ 还对 C 语言的一些不够完善的地方进行了修改, 例如, 引入了 “引用变量” 用以取代指针变量。

3. C 语言的源程序是由 (D) 组成的。

- [3] A) 若干个过程 B) 若干个子程序
C) 一个主程序和几个子程序 D) 若干个函数

4. 下列说法中 (B) 是正确的。

- [4] A) C 语言程序总是从第 1 个定义的函数开始执行
B) C 语言程序总是从 main() 主函数开始执行
C) C 语言源程序中 main() 主函数必须放在程序的最开头
D) C 语言源程序中, 一个函数体内只允许使用一对大括号 “{ }”

5. 下列说法中 (D) 是错误的。

- [5] A) 一个 C 语言源程序可以由多个源文件组成
B) 一个 C 语言源程序必须包含一个且只能包含一个 main() 主函数
C) C 语言源程序的基本组成单位是函数
D) 在 C 语言源程序中, 注释说明只能出现在一条语句的后面

6. 下列说法中 (D) 是错误的。

- [6] A) 分号是 C 语言语句的必要组成部分
B) C 语言源程序不使用行号
C) 函数是 C 语言源程序的基本组成单位
D) C 语言源程序中, 主函数的名字不一定用 main

7.

```
#define     PI     3.141593
#include    <stdio.h>

void    main( void )
```

```
{ printf("PI = %f\n", PI); }
```

该程序的输出结果为 (B)。

- [7] A) 3.141593 = 3.141593 B) PI = 3.141593
C) 3.141593 = PI D) 程序有错误, 无输出结果

三、问答题

1. 请列举 C 语言的主要特点。

答 C 语言的主要特点如下。

(1) 介于高级语言 (PASCAL) 和汇编语言之间, 兼有两者的优点。既适合于开发系统软件, 又适合于编写应用程序, 虽然 C 语言最初设计时, 是作为一种面向系统软件 (OS 和语言处理系统) 的开发语言, 即用来代替汇编语言的, 但是由于它强大的生命力, 以致足以取代汇编语言来编写各种系统软件和应用软件。在事务处理、科学计算、工业控制和数据库技术等方面都得到了广泛应用。即便进入到以计算机网络为核心的信息时代, C 语言仍然是作为通用的汇编语言使用, 由于它的开放性和兼容性, 可做到与硬件平台无关。

(2) 引用模块化程序结构, 便于软件工程化。现在的计算机软件系统, 即便是应用软件, 常常不可能是一个人的劳动结晶, 而是一班人、一个梯队、多少“人年”完成的。因此, 软件必须走“工程化”的道路, 即采用工程的概念、原理、技术和方法来开发和维护软件。模块化程序设计仍然是当前软件工程最基本、最普遍采用的设计方法, 自顶向下划分模块, 直到最底层的每个模块都是能完成单一独立的功能为止。C 和 C++ 是以函数模块为单位来思考问题的, 每个模块有特定的目的和功能, 一个 C 和 C++ 程序只不过是把这些模块装配起来, 以实现编程者所要求的全部任务。

(3) 语言简洁, 且表达能力强, 使用灵活, 易于学习和应用。

(4) C 和 C++ 的整个源程序是一些函数的集合体, 便于分块编译。软件工程所采用的方法是将一个复杂的问题分解成许多简单的子问题, 然后再解决各个子问题。它的基本思想是把软件设计成由相对独立、功能单一的若干个模块组成, 又称模块化程序设计。而 C 和 C++ 的源程序可以划分成一个以上的源文件, 在编译该程序时, 可以按一个个源文件为单位对它们单独地进行编译, 然后通过系统所配有的链接程序 (LINK) 将它们连接、装配在一起, 成为一个可执行文件, 分块编译把整块程序分割成若干个相对独立的功能模块, 自顶向下得到最底层一系列模块后, 即可分发给软件开发小组的各成员在多台计算机上独立地、并行地同时进行开发各个模块, 最后再汇总进行整体调试, 这种由众人同时进行集体性开发的软件工程技术方法加快了软件开发速度, 大幅度地缩短了开发周期。

(5) 配备有“预处理操作命令” (Preprocessor Directives), 增强了用户所编程序的灵活性和适应性。

2. C++ 和 C 语言的关系如何? 为什么说 C++ 是 C 语言的一个超集?

答 C 语言蕴涵在 C++ 中是它的一个子集, 即它的模块化语言基础的内核所在, 换句话说, C++ 是 C 语言的超集, 它保留了 C 的所有组成部分并与其完全兼容, 而只是增加了

面向对象的编程机制，可以说它是 C 语言的面向对象扩展，既可以做传统的结构化程序设计，又能进行面向对象程序设计，是面向对象程序设计的第 1 个大众化版本，是当前学习面向对象程序设计方法的首选语言。在系统软件的开发研究上，C++ 的运行效率与 C 相比毫不逊色，在大型应用软件开发上，以 Windows 开发环境为操作系统平台的 C++ 标准类库和组件正在迅猛发展，C++ 即将取代 C 已是不可抗拒的事实，它涉及计算机研究和应用的各个领域。

3. 何谓“分块编译”和“一个编译单位”？

答 C 和 C++ 的源程序可以划分成一个以上的源文件，在编译该程序时，可以按一个个源文件为单位对它们单独地进行编译，这些单独进行编译的每个源文件称为“一个编译单位”。然后通过系统所配有的链接程序（LINK）将它们连接、装配在一起，成为一个可执行文件，即“文件名.EXE”。这样的编译过程称为“分块编译”。

4. 何谓“标准函数库”和“头文件”？

答 C 语言与其他高级语言不同，本身不具备输入/输出（I/O, Input/Output）操作语句，这些功能是由执行输入/输出操作的标准函数来实现的。由 C 语言系统所提供的这些标准函数的集合体称为“标准函数库”。标准函数库不仅配备了各种 I/O 操作的标准函数，而且还备有各种常用的数学函数、字符和字符串处理函数、内存管理和进程管理函数、屏幕绘图函数、时间和日期函数以及与系统有关的函数等。这些标准函数的说明以及它们所使用的符号常量、宏指令的定义等语句都包含在扩展名为“.h”的相应文件中，这些扩展名为“.h”的文件，称为头文件。因此在使用某标准函数时，必须用#include 语句嵌入相应的头文件。例如，在源程序中若使用了 printf()、scanf() 等标准函数，则必须要写上：

```
#include <stdio.h>
```

5. 请说明#define 和#include 两个预处理命令的主要作用？

答 (1) #define 语句又称宏定义语句，其格式为：

```
#define 字符串 1    字符串 2
```

它是把字符串 1 定义成字符串 2，在预处理操作时，系统把程序中出现的字符串 1 一律用字符串 2 替换。其主要作用如下。

① 将一个常量值用编程者所启用的名字（通常用英文大写字母组成的字符串）来代替，该常量称为“符号常量”。例如：

```
#define NULL    0
#define EOF    -1
#define PI    3.1415926
```

② 定义宏指令（简称“宏”）和带参数的宏指令。例如：

```
#define MAX(x, y) ((x) > (y)) ? (x) : (y))
```

求 x 和 y 中较大的一个，若程序中写有：

```
int a = 6, b = -2, c;
```

```
c = MAX(a, b);
```

在编译前的预处理操作时, 不仅把字符串 1 “MAX(x, y)” 替换成字符串 2 “((x) > (y)) ? (x) : (y)”, 且还把实参 a 和 b 分别赋给形参 x 和 y, 则上面的赋值语句变成:

```
c = ((a) > (b)) ? (a) : (b);
```

赋值表达式的右值为变量 a 和 b 中的较大者 a, 其值为 6 赋给变量 c。

(2) #include 语句是指一个源文件可以把另一个源文件的全部程序代码内容都包含进来, 通常是将一个已开发完成的源文件包含到编程者正在开发的源文件之中。其格式为:

```
#include <文件名> (1)
```

```
#include "文件名" (2)
```

特别在使用某标准函数时, 必须用#include 语句嵌入相应的头文件, 这时应使用格式(1)。例如,

在源程序中若使用了 strcmp()、strcpy() 等标准函数, 则必须要写上:

```
#include <string.h>
```

在源程序中若使用了 sin()、cos()、tan()、asin()、acos()、exp()、log() 等初等数学函数, 则必须要写上:

```
#include <math.h>
```

如果编程者自行编写的一个头文件如 menu.h, 则通常采用格式(2), 即:

```
#include "menu.h"
```

6. 请说明如何编译、链接和运行单文件和多文件源程序?

答 (1) 用教材例 1.1 源程序来说明单文件的编译、链接和运行过程。若该程序已经显示在 Visual C++ 集成开发环境的编辑窗口上, 如图 1.1 所示, 用鼠标沿着 “Build

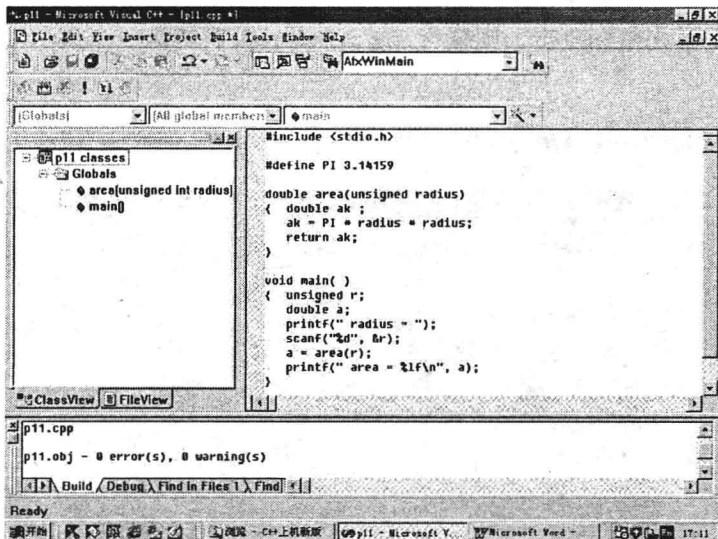


图 1.1 单个文件源程序的编译、链接和运行

（主菜单）/Compile p11.cpp（子菜单）Ctrl+F7（热键）”的操作路径点击，这时编译系统对当前源程序 p11.cpp 进行编译，并把编译过程中所发现的编译错误（Compiler Error）显示在屏幕下方的 Build 窗口上，并指出该错误所在的行号、编译错误码和错误性质的简洁提示，本系统的编译错误码采用“C××××”的格式表示，其中“×”代表一位十进制数码。编程者可根据这些错误信息来修改源程序，若还需进一步了解错误信息和改错的对策，可沿着“Help（主菜单）/Contents（子菜单）或 Search（子菜单）”操作路径用鼠标点击，启动 MSDN（Microsoft Developer Network）库，则弹出“MSDN Library Visual Studio 6.0”窗口，简称 MSDN 窗口，在该窗口的“Multiple Page”控件上点击“搜索”页面，在该页面的“输入要查找的单词(w):”框内键入编译错误码，例如 C2660(CR)。往往一个错误会引起多行错误信息，因此一般是针对第 1 行错误来修改源程序再重新编译，如果原来第 1 行错误消除了，再针对新的第 1 行错误来修改，再重新编译，直到没有编译错误为止。如果没有编译错误，则在 Build 窗口上，将出现如下信息：

-----Configuration: p11 - Win32 Debug-----

Compiling...

p11.cpp

p11.obj - 0 error(s), 0 warning(s)

这说明编译成功，并生成可重定位文件 p11.obj。接着进行链接操作，操作方法是沿着“Build（主菜单）/Build p11.cpp（子菜单）或 F7（热键）”的操作路径用鼠标点击，链接过程中的错误处理与编译时类同，仅连接错误码采用“LNK××××”的格式表示，例如，LNK2660。链接成功后，在 Build 窗口上将出现如下信息，则说明已生成了可执行文件 p11.exe。运行可执行文件的方法是沿“Build（主菜单）/Execute p11.exe（子菜单）”的操作路径用鼠标点击，则执行 p11.exe，并将弹出一个显示输出结果的窗口，如图 1.2 所示。

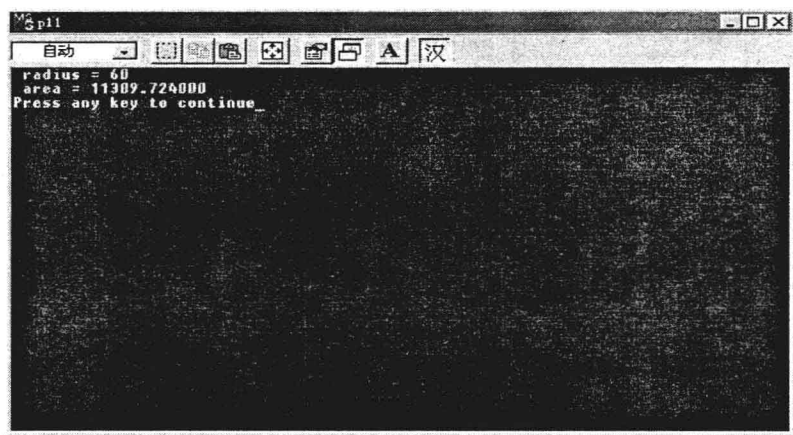


图 1.2 输出结果窗口

当执行到“printf(" radius = ");”语句时,则该窗口显示出“ radius = ”提示信息,操作者接着键入一个半径值为60,通过“scanf("%d",&r);”语句由变量r接收,随后以r为实参调用area()函数,将其返回值存放在double型变量a中,最后执行“printf(" area = %Lf\n", a);”语句,在输出结果窗口上显示出计算结果值“ area = 11309.724000”。

(2) 对多文件源程序应采用建立工程项目的方法,例如有如下3个源文件:

```
#include "Area.h" //Circle.cpp 计算圆面积函数 circle( ) 的定义
const double pi = 3.1415926;
double circle(double radius)
{ return pi * radius * radius; }
```

```
#include "Area.h" //Rect.cpp 计算矩形面积函数 rect( ) 的定义
double rect(double width, double length)
{ return width * length; }
```

```
#include <stdio.h> //CalculateArea.cpp 调用不同功能函数计算面积
#include "Area.h"
void main( )
{ double w , l;
  printf("Please enter a width : ");
  scanf("%lf", &w);
  printf("Please enter a length : ");
  scanf("%lf", &l);
  printf("Area of the rectangle is %lf\n", rect(w, l));

  double r;
  printf("Please enter a radius : ");
  scanf("%lf", &r);
  printf("Area of the circle is %lf\n", circle(r));
}
```

并自行定义一个头文件Area.h包含各源文件内所定义的外部函数的声明语句,即:

```
// Area.h
double circle(double radius);
double rect(double width, double length);
```

然后,凡是在要用到这些外部函数的源文件开头处写上“#include "Area.h"”语句(源文件中画有虚线的语句),就可以调用这些函数。

建立工程项目的操作方法是:按照“File(主菜单)/New(子菜单)”的操作路径进入到New对话框,在New对话框内,再选择多页面控件中的Project页面,接着选择“Win32

Console Application”子菜单项，这时项目（Project）的目标平台选择框内将会出现“Win32”；然后，在“Project name:”框内输入工程文件名为 CalculateArea，在“Location:”框内会自动出现指定路径名为“d:\test\CalculateArea”，建立工程文件所需要的所有源文件（例如，Area.cpp、Circle.cpp、Rect.cpp 和 Area.h 等）应该放在该路径名下。若输入的路径名为“d:\test”，则在“Location:”框内将会出现“d:\test\CalculateArea”。也可用鼠标点击“Location:”框右边的浏览键，在“Choose Directory”对话框内选取所需要的路径；再用鼠标点击 New 对话框内的 OK 键，则将弹出“Win32 Console Application - Step 1 of 1”选择框，可选择“An empty project”项后，点击 finish（完成）键，则弹出“New Project Information”窗口，其内将出现如下信息。

+ Empty console application.

+ No files will be created or added to the project.

用鼠标点击该窗口内的 OK 键，则建立了名字为 CalculateArea 的工程文件项目。接着向工程文件项目中添加源文件，按照“Project（主菜单）/Add To Project（1 级子菜单）/New（2 级子菜单）”的操作路径进入到“New”对话框，在其内再选择多页面控件中的 Files 页面，接着选择 C++ Source File 项，并在右边 File 框内填上源文件名 Area（默认扩展名为“.cpp”，可以写也可以不写；“.c”的扩展名必须写），随后按回车键或用鼠标点击 OK 按钮，即将新建的源文件 Area.cpp 添加到 CalculateArea 的工程项目中，接着进入到编辑窗口，操作者用键盘敲入该文件的内容并保存，才完成了添加一个源文件的操作。用同样的方法把 Circle.cpp、Rect.cpp 和 Area.h 等文件添加到 CalculateArea 的工程项目中，而在添加头文件 Area.h 时，应该选择 New 对话框 Files 页面中的 C/C++ Header File 项，则默认扩展名即为“.h”。

如果要添加的文件已经存在，则按照“Project（主菜单）/Add To Project（1 级子菜单）/Files（2 级子菜单）”的操作路径进入到“Insert Files into Project”对话框，如图 1.3 所示，其下方的“insert into:”应出现工程项目名 CalculateArea。



图 1.3 将已存在的文件 Area.cpp 添加到 CalculateArea 中

在该对话框中用鼠标点击的方法从指定的目录（如图 1.3 中的“d:\test\CalculateArea”）下选取所要添加到该工程文件项目的文件（如图 1.3 中的 Area.cpp，添加文件

可以是任何盘任意目录下已存在的源文件 `filename.c`、`filename.cpp` 或目标文件 `filename.obj` 等，为了便于管理和移植，通常应把它们移到工程项目的目录路径下，如“`d:\test\CalculateArea`”，选中后双击该文件名，则将所选中的文件添加到项目 `CalculateArea` 中。

编译、链接和运行工程文件项目时，按照“`Build(主菜单)/Rebuild All(子菜单)`”的操作路径用鼠标点击，则对项目中所包含的所有文件进行编译、链接。若有错误，将显示在错误信息窗口上，便于编程者修改程序时阅读；当没有错误信息时，便生成了一个可执行文件 `CalculateArea.exe`，其文件名就是项目名。运行工程文件的方法与单个源程序完全相同，即按“`Build(主菜单)/Execute p11.exe(子菜单)`”的操作路径用鼠标点击，则执行 `CalculateArea.exe`，其输出结果为：

```
Please enter a width : 64.0 (CR)      (其中“(CR)”表示按回车键，下同)
Please enter a length : 32.0 (CR)
Area of the rectangle is 2048.000000
Please enter a radius : 72.0 (CR)
Area of the circle is 16286.016038
```