



普通高等学校计算机科学与技术应用型规划教材

C语言程序设计与应用 实验指导

C YUYAN

CHENGXU SHEJI YU YINGYONG SHIYAN ZHIDAO

主 编 杨新艳 房正华
副主编 梁宏涛 姚立新
林旭平 苏爱玲



北京邮电大学出版社
www.buptpress.com

C 语言程序设计与应用 实验指导

主 编	杨新艳	房正华
副主编	梁宏涛	姚立新
	林旭平	苏爱玲

北京邮电大学出版社
·北京·

内 容 简 介

本实验指导教材是与北京邮电大学出版社出版的《C 语言程序设计与应用》一书相配套的学生用书。

本书的主要内容有:实验、课堂练习及答案、习题及答案 3 大部分和一个附录(实验报告格式参考)。每一部分的章节都和《C 语言程序设计与应用》一书的章节相对应。第 1 部分“实验”,每个实验都先说明实验目的,再以实验案例引导学生,最后给出具体的实验内容,并要求学生进行总结和分析。第 2 部分“课堂练习及答案”以及第 3 部分“习题及答案”给出了《C 语言程序设计与应用》各章课堂练习和习题的参考答案。

本书内容丰富,条理清晰,案例程序设计讲解详细,既可作为高等院校 C 语言教学的配套教材,也可作为 C 语言程序设计初学者和爱好者的自学用书。

图书在版编目(CIP)数据

C 语言程序设计与应用实验指导/杨新艳,房正华主编.--北京:北京邮电大学出版社,2011.1
ISBN 978-7-5635-2547-8

I. ①C… II. ①杨…②房… III. ①C 语言—程序设计 IV. ①TP312

中国版本图书馆 CIP 数据核字(2011)第 004081 号

书 名: C 语言程序设计与应用实验指导
主 编: 杨新艳 房正华
责任编辑: 王丹丹 赵 静
出版发行: 北京邮电大学出版社
社 址: 北京市海淀区西土城路 10 号(邮编:100876)
发 行 部: 电话: 010-62282185 传真: 010-62283578
E-mail: publish@bupt.edu.cn
经 销: 各地新华书店
印 刷: 北京源海印刷有限责任公司
开 本: 787 mm×1 092 mm 1/16
印 张: 9.75
字 数: 242 千字
印 数: 1—3 000 册
版 次: 2011 年 1 月第 1 版 2011 年 1 月第 1 次印刷

ISBN 978-7-5635-2547-8

定 价: 18.00 元

· 如有印装质量问题,请与北京邮电大学出版社发行部联系 ·

前 言

学习 C 语言程序设计的最好方法就是动手编写程序,上机调试程序,在反复的实践和使用中才能更好地掌握。

本书是《C 语言程序设计与应用》的配套教材,按照相应的章节精选实验实例,并给出在 Microsoft Visual C++ 6.0 环境下的程序调试案例,旨在帮助学生掌握调试程序的技能。

本书还给出了《C 语言程序设计与应用》一书中的练习题和每章习题的参考答案,代码均能在 Microsoft Visual C++ 6.0 环境下编译运行。程序设计题的答案不是唯一的,给出的只是其中一种算法和参考代码,也并非是最优的,希望读者不局限于答案的思路,设计出更好的程序。

本书主要为学习理论知识后,进行实验训练而编写的,主要有以下特点:

(1) 与理论教材结合紧密。对应每章节的主要知识点设计了实验的内容,并与教材中的练习题结合,互为补充。有效的实验是理解和掌握 C 语言程序设计的重要环节。

(2) 实验题目简单实用。每个实验都有侧重知识点,重点章节设计多个实验,符合初学者学习 C 语言的基本要求。能够激发学生的学习兴趣,调动学生学习的积极性。

(3) 理论知识的实践与调试能力的提高同步进行。每个实验,从内容上既有知识的应用,也有调试技巧的基本训练。通过这些实验的训练,不仅能提高编程能力,还能够提高程序的调试技能。

(4) 侧重程序设计逻辑思维的训练。初学者往往对编程有畏惧感,面对问题感觉无从下手,对实际问题作为实验案例先给出算法思路分析、N-S 流程图,再根据分析过程逐步实现程序,最后通过调试和思考问题,体会设计算法和程序的思路和方法,逐步培养分析问题和解决问题的能力。

(5) 练习题和习题按照理论教材安排。练习题侧重针对性,分布在各个章节中,以选择题、填空题为主,读程序写结果,帮助理解和掌握各知识点。每章后的习题侧重综合性,并与实验的内容紧密结合,注重综合应用和编程调试。

本书第 1 部分是对应理论教材的章节的实验内容,共分 8 章。可根据学时数进行选用,一般需要 34~40 学时。第 2 部分是理论教材中的练习题目及解答。第 3 部分理论教材中每章习题及解答。

由于编者的水平有限,书中难免存在错误和不妥之处,敬请读者批评指教。

编 者

目 录

第 1 部分 实验

第 1 章 C 语言初步	3
第 2 章 数据类型与表达式	10
第 3 章 程序控制结构	16
3.1 顺序结构和分支结构	16
3.2 循环结构	22
第 4 章 数组程序设计	28
4.1 一维数组	28
4.2 字符数组、字符串	32
4.3 二维数组	38
第 5 章 函数程序设计	42
5.1 函数的基本应用	42
5.2 函数的嵌套与递归函数	47
第 6 章 指针	54
6.1 指针与数组	54
6.2 指针与字符串	58
第 7 章 结构体	61
第 8 章 文件	65

第 2 部分 课堂练习及答案

第 2 章 数据类型与表达式	71
第 3 章 程序控制结构	74
第 4 章 数组程序设计	90

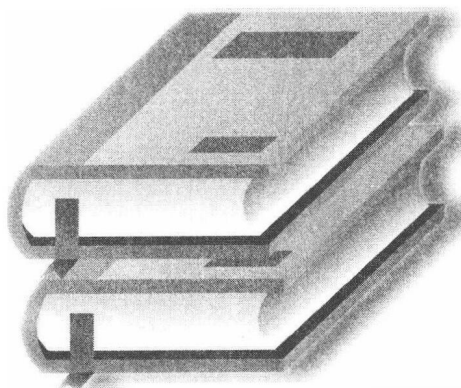
第5章 函数程序设计	94
第6章 指针	100
第7章 结构体	105

第3部分 习题及答案

第1章 C语言初步	113
第2章 数据类型与表达式	114
第3章 程序控制结构	117
第4章 数组程序设计	123
第5章 函数程序设计	130
第6章 指针	135
第7章 结构体	143
第8章 文件	145
附录 实验报告格式参考	149
参考文献	150

..... 第1部分

实 验



第 1 章 C 语言初步

实验目的

- (1) 了解 C 语言程序从编写、编译、连接到最后运行的全过程。
- (2) 了解 C 语言程序的基本框架,并掌握其规范要求,能够编写简单的 C 语言程序。
- (3) 熟悉 C 语言程序编译工具。了解程序调试的思想,能借助于 Visual C++ 6.0 编译环境修改程序中出现的简单语法错误。

实验案例

【例 1-1】 在屏幕上显示一个短句“This is my first program!”。

1. 解题步骤

C 语言程序的编写应按照以下思路进行。

- (1) 确定程序的目的或作用。
- (2) 分析问题。
- (3) 绘制程序流程图。
- (4) 根据流程图编写程序源代码程序。
- (5) 利用一种 C 语言编译器编译程序。
- (6) 执行程序。

2. 实际操作

- (1) 准备工作:建立自己的文件夹。

在磁盘(最好是除系统盘外的其他磁盘)上新建一个文件夹,用于存放 C 程序,如 E:\2000021051 名字。

- (2) 分析题目要求,寻找解决思路。

要在屏幕上显示字符串就要用到输出函数 printf,相应的需要添加编译预处理命令。

- (3) 绘制流程图或 N-S 图。

流程图的绘制对于程序的编写至关重要,将起到事半功倍的效果,而且可以规避很多错误,减少程序出错的可能。

- (4) 编辑 C 语言源程序(C++ Source File)。

根据流程图编写程序代码。

① 编辑工具

C 语言源程序的编辑可以采用各种编辑器,包括最简单的记事本程序,还可以采用专门的程序编辑器 UltraEdit、Editplus 等,还有一些 C 语言的 IDE(集成开发环境)环境中自带的编辑器。本书采用的是 IDE 中的编辑器。方法如下:

a. 准备工作。启动 Visual C++ 6.0,通过开始菜单—程序—Microsoft Visual Studio 6.0—Microsoft Visual C++ 6.0,启动后界面如图 1-1-1 所示。

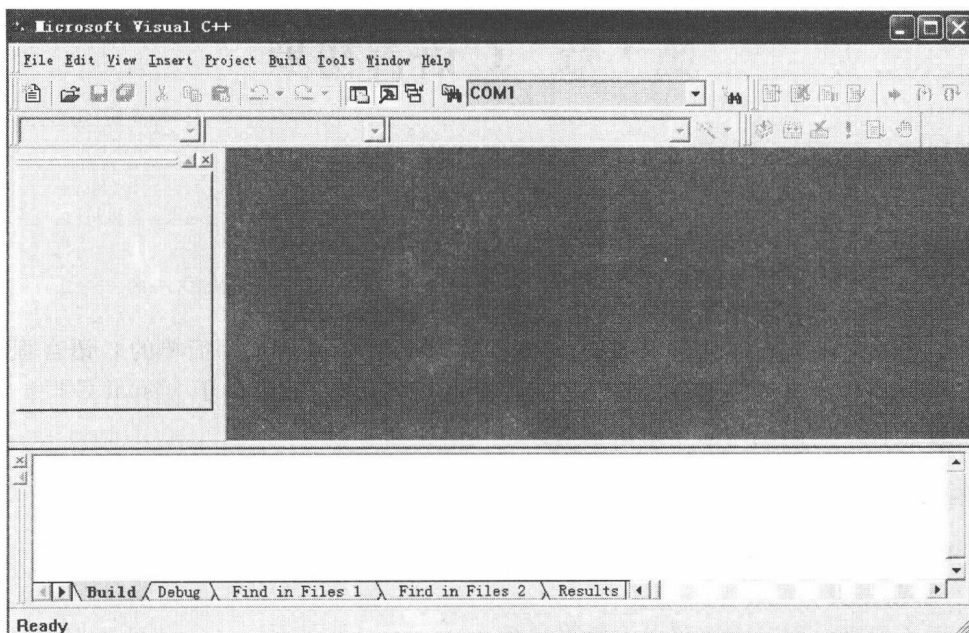


图 1-1-1 Visual C++ 6.0 启动界面

b. 建立工程。工程类型应为“Win32 Console Application”,即 Win32 控制台应用程序。所有的程序必须依托在工程下才能运行,如图 1-1-2 所示。工程位置可以设置为某一路径(如 a. 准备工作中建立的文件夹)。

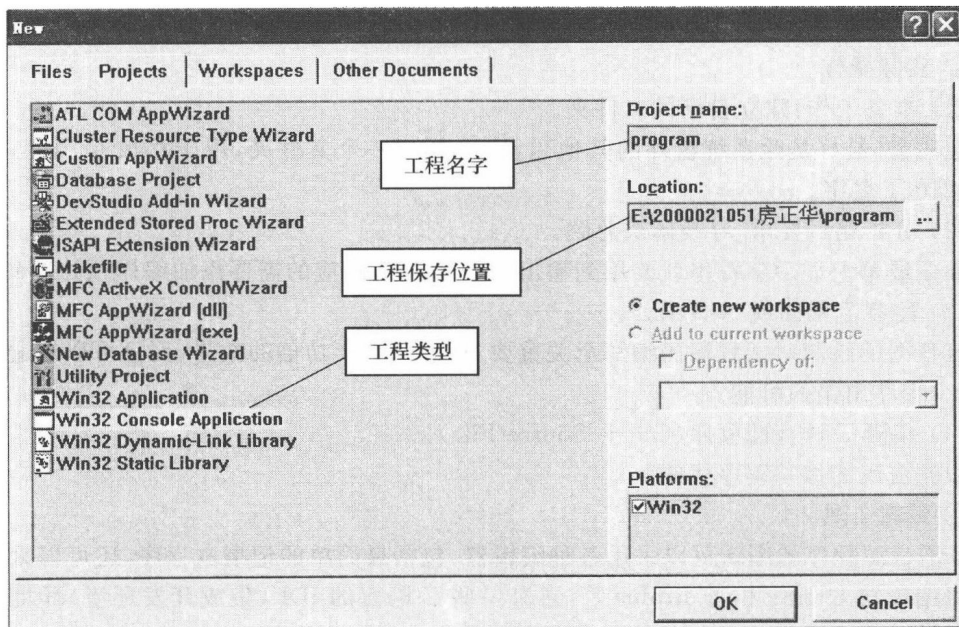


图 1-1-2 创建工程界面

c. 建立文件。如建立的是源程序文件,则文件类型应选择“C++ Source File”,如果要建立的是头文件(.h 文件),则文件类型应选择“C/C++ Header File”。C 语言的源程序文件扩展名为.c,在命名时,如果省略文件的扩展名,将默认为.cpp 文件(每次上交作业也只上交.cpp 文件或.c 文件)。如图 1-1-3 所示。

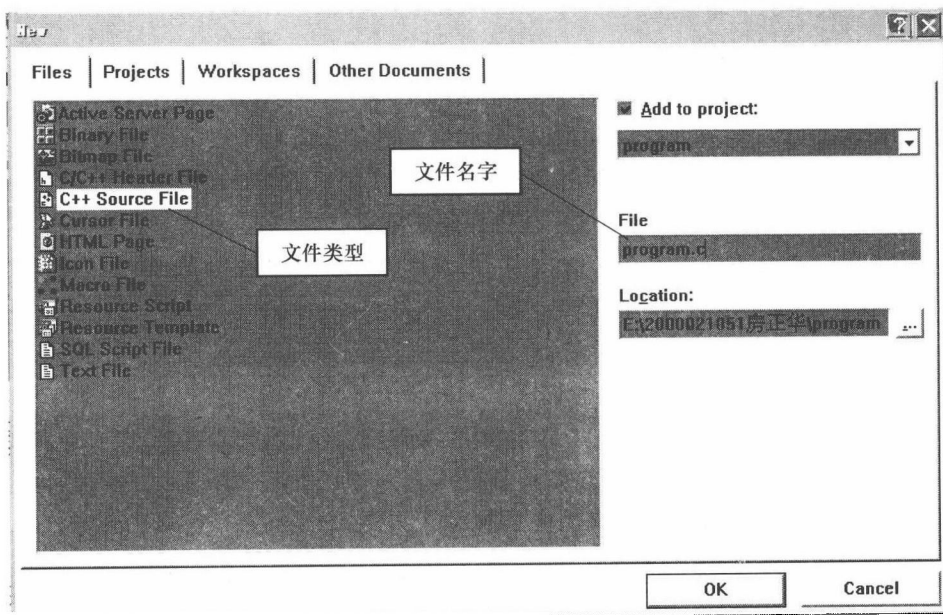


图 1-1-3 创建文件界面

d. 编辑并保存源程序文件。

PS: 源程序

```

1  /*****
2  文件名:program.c
3  作者:***
4  日期:2010.1.25
5  功能:在屏幕上输出字符串"This is my first program!"
6  *****/
7  #include <stdio.h>           //编译预处理命令
8  int main(void)              //主函数
9  {
10     printf("This is my first program!"); //在屏幕上输出字符串
11     return 0;
12 }
```



说明

以上程序代码每行行首的数字在本书中代表行号,是额外添加的,用 Visual C++ 6.0 来编写程序时,不需要加以上数字。

② C 语言程序的框架。

所有的 C 语言程序由有一个且只有一个 main 函数组成。函数体用“{ }”括起来。函数体内的每一条语句都以分号来结束。如果程序需要输入输出,则要添加预处理命令,即以上程序中第 7 行的代码,该行为命令,不属于语句,因此结尾没有分号。

③ C 语言程序的编辑规范。

a. 留白艺术(不要吝啬空间);b. 使用 Tab 键;c. 添加注释。

“大牛”并不是编写一个谁也看不懂的程序。因此,在保证程序正确性的前提下,更应该注重编程的规范性,从某方面来说,后者也无形中为前者提高起了很大的作用。

- 留白指的是留下空白。在某种程度上,空白在提高程序的可读性方面比注释更重要。人们在看程序时需要空白,程序代码不能都挤在一起。
- 留下空白的方法可用 Space 键,也可用 Tab 键,但很多情况下,特别是在行首的时候,我们都是采用后者,它可以方便进行对齐。
- 注释可能是给程序的作者看的,也可能是给其他读者看的。

(5) C 语言程序的编译、连接与运行。

① 编译。如图 1-1-4 所示可单击工具栏中的“编译”按钮,或选择 Build 菜单中的 Compile 子菜单。编译后会在编辑窗口的下方显示编译信息。如果此时源程序文件有错误,则应改正,否则无法继续进行。如果编译成功将生成 .obj 文件。



图 1-1-4 编辑界面

② 连接。可单击工具栏中的“连接”按钮,或选择 Build 菜单中的 Build 子菜单。同时也会在环境界面的下方显示连接信息。如果连接过程有错误,将显示错误原因。如果连接成功,将生成 .exe 文件。

③ 运行。可单击工具栏中的“运行”按钮,或选择 Build 菜单中的 Execute 子菜单。将显示运行结果窗口。

(6) C 语言程序的调试。

C 语言程序的编译如果出现错误,即源代码程序有 bug,为了能继续对目标代码进行连接以正常运行程序,必须找出相应的 bug,并加以修改,这个过程就成为 debugging,即调试。

C 语言程序的错误分为两种:语法错误和语义错误,其中后者又称为逻辑错误。语法错误可由 C 语言编译器找出,但语义错误只能由编程者通过对程序各条语句的实际运行结果和它的预期结果相比较后才能找出,这就需要编程者对程序非常熟悉,同时会使用各种调试工具,在后续的实验中大家会更深刻了解到这一点。

所有的 C 语言程序都是调出来的,而不是写出来的。

Visual C++ 6.0 编译器通知的语法错误分为 error 和 warning 两种,其中前者是致命的,将使程序无法继续连接也不能继续运行,warning 不影响程序的继续进行,但最后也对其进行修改。连接中出现的错误对于初学者来说最大的可能是 main 函数拼写错误,此外可以尝试关闭编译器,删除编译过程中新生成的文件,重新编译连接即可。

有以下程序源代码:编译后显示的错误提示如图 1-1-5 所示。在该提示中,给出了错误所在的行,错误的类型及原因,可根据该行号提示或双击该错误提示行,查找出现错误的源代码。并根据错误的原因对错误进行修改,直到没有错误为止。

注意

- ① 错误可能会出现在提示中给出的行号的上一行。
- ② 对于初学者,每修改完一个错误,就应马上重新编译程序。

运行示例:

Hello World!

This is my first program!

源程序如下:

```
1  / ****
2  文件名:program2.c
3  作者:***
4  日期:2010.1.25
5  功能:在屏幕上输出两行字符串,分别为 Hello world! 和 This is my first program!
6  ****/
7
8  #include <stdio.h>           //编译预处理命令
9  int main(void)              //主函数
10 {
11     printf("Hello World!");   //在屏幕上输出字符串
12     printf("This is my first program!")
13     return 0;
14 }
```

编译后显示有错误,如图 1-1-5 所示。

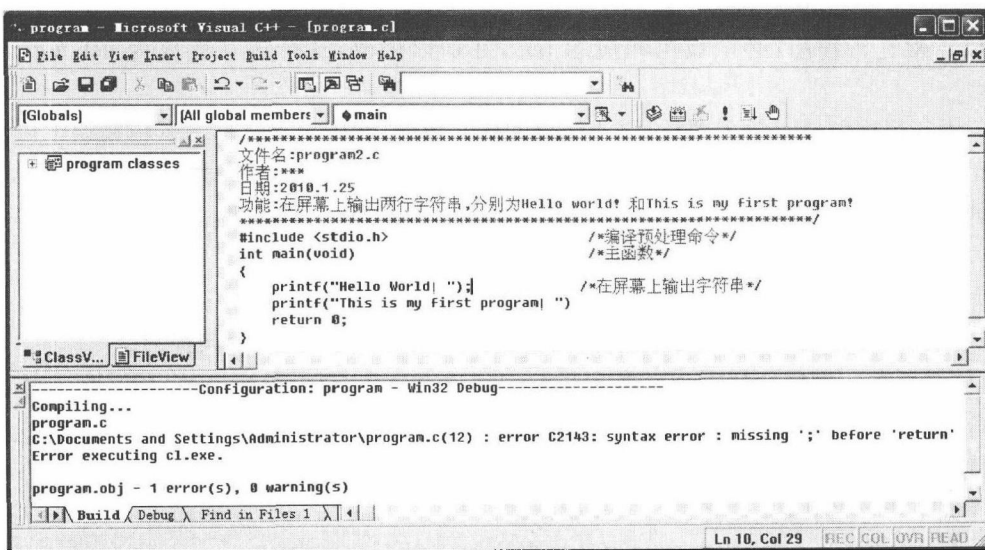


图 1-1-5 编译错误图

在信息窗口中,双击第一条错误信息,编辑窗口就会出现一个箭头指向程序出错的位置。一般在箭头的当前行或上一行,就可以找到出错语句。该例中错误的原因指示在第 13 行,属于语法错误(syntax error)。

missing ';' before 'return'

在第 13 行的 return 之前缺少分号。实际上是第 12 行的 printf 语句最后缺少分号,但因为 C 语言语法中规定一条语句可以书写在多行中,所以语句的分号可以写在下一行中,因此编译工具才在第 13 行中给出错误提示。改正错误,在第 12 行的“printf”语句末尾补上一个分号。重新编译,无误,然后连接,最后运行得到输出结果窗口如图 1-1-6 所示。

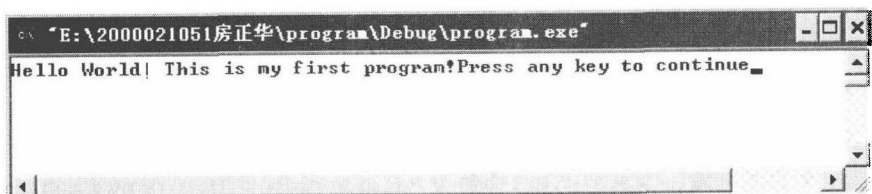


图 1-1-6 运行结果

在结果的最后“Press any key to continue”是运行时系统自动加上的,含义是按下任意键即可退出,返回到程序的编辑状态。

程序的运行结果与之前要求的仍然不一致,说明程序仍然存在错误,此时的错误就称为语义错误。错误原因是换行符是特殊的符号‘\n’应将程序第 11 行修改为:

```
11    printf("Hello World! \n")           //在屏幕上输出字符串
```

或者将第 12 行修改为

```
12    printf("\nThis is my first program!")
```

或者将 11 行和 12 行合并为 1 行

```
11    printf("Hello World!\n This is my first program!") //在屏幕上输出字符串
```

为了能清楚地显示结果,最好在“program”之后也加一个回车换行符。保存程序,重新编译、连接、运行后,得到程序运行结果如图 1-1-7 所示,与题目要求结果一致,按任意键返回。

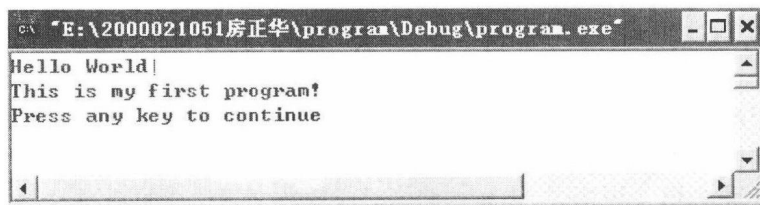


图 1-1-7 运行结果



思考

根据程序分别在编译、连接、运行之前和之后工作目录中文件的变化(即程序编译后,又生成哪些文件),了解相应文件起什么作用?

实验内容

(1) 请编写程序在屏幕上显示你的名字、爱好以及你想对老师说的话。同时将你的名字以注释的方式显示出来。请注意编程规范。

(2) 请编写一个程序,输出以下图形。注意编程规范。

```
*****
*      Hello World!      *
*****
```

(3) 有源程序如下,请说明程序是否有错误,如果有请注明错误所在的行,同时对程序的编程风格给出自己的评价。如果程序不规范,请按照编程规范进行修改。

```
1 //该程序可以显示出一年有多少周
2 include studio.h
3 int mian(){
4 printf(There are 52 weeks in a year!); return 0}
```

(4) 编写程序说明将一只大象放入冰箱中的步骤,每个步骤一行。注意编程规范。

实验总结与分析

第 2 章 数据类型与表达式

实验目的

- (1) 掌握不同类型变量定义的方法,并体会它们之间的不同。
- (2) 掌握不同类型变量的输入和输出方法。
- (3) 掌握各种 C 语言表达式,体会它们与对应的数学表达式的不同。
- (4) 继续体会程序调试的思想,借助于 Visual C++ 编译环境查出的错误对 C 程序中进行修改。

实验案例

【例 2-1】 任意给定两个整数,例如 4 和 5,实现简单的计算器功能,即求出它们的和、差、积、商和余数。



分析

要计算 2 个数的各种运算结果,首先应保存这两个数,方法是定义两个变量,同样还应定义 5 个变量来存放它们的 5 种运算的运算结果。定义完变量后给两个整数变量赋初值,然后进行计算,并将计算结果分别保存在 5 个变量中,最后显示计算结果,根据以上分析,绘制 N-S 图如图 1-2-1 所示。

4=>lOperator
5=>rOperator
lOperator+rOperator=>nResult1
lOperator-rOperator=>nResult2
lOperator*rOperator=>nResult3
lOperator/rOperator=>nResult4
lOperator%rOperator=>nResult5
输出nResult1,nResult2,nResult3,nResult4,nResult5

图 1-2-1 例 2-1 的 N-S 图

根据 N-S 图编写源程序代码如下:

```
1  / *****
2  文件名:compute.c
3  作者:***
4  日期:2010.1.25
5  功能:简单计算器功能
6  *****/
7
```

```

8  #include <stdio.h>                //编译预处理命令
9  int main(void)                    //主函数
10 {
11     int lOperator,rOperator; //定义 2 个整型变量,用于存放左右操作数
12     int nResult1,nResult2,nResult3,nResult4,nResult5; //定义 5 个整型变量,
        用于存放结果
13     lOperator = 4;
14     rOperator = 5;
15     nResult1 = lOperator + rOperator; //计算 lOperator 与 rOperator 相加,把
        结果存入 nResult1 中
16     nResult2 = lOperator-rOperator; //计算 lOperator 与 rOperator 相减,把结
        果存入 nResult2 中
17     nResult3 = lOperator * rOperator; //计算 lOperator 与 rOperator 相乘,把
        结果存入 nResult3 中
18     nResult4 = lOperator/rOperator; //计算 lOperator 与 rOperator 相除,把结
        果存入 nResult4 中
19     nResult5 = lOperator % rOperator; //计算 lOperator 与 rOperator 余数,把
        结果存入 nResult5 中
20     printf("%d",nresult1);        //在屏幕上显示和
21     printf("%d",nresult2);        //在屏幕上显示差
22     printf("%d",nresult3);        //在屏幕上显示积
23     printf("%d",nresult4);        //在屏幕上显示商
24     printf("%d",nresult5);        //在屏幕上显示余数
25
26     return 0;
27 }

```

保存后编译,发现有错,如图 1-2-2 所示。

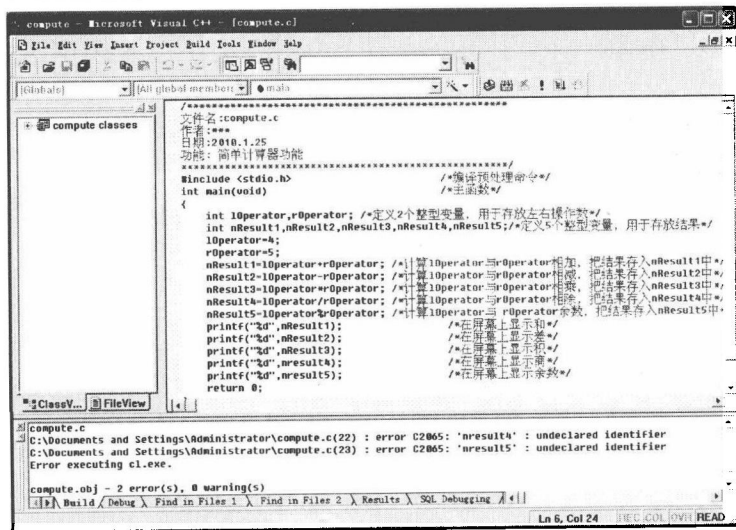


图 1-2-2 编译后错误提示图