

全国高等院校计算机教育规划教材

C语言程序设计 实践教程

C YUYAN CHENGXU SHEJI SHIJIAN JIAOCHENG

陈惠明 张 静 何志国 主编

中国铁道出版社
CHINA RAILWAY PUBLISHING HOUSE

全国高等院校计算机教育规划教材

C 语言程序设计实践教程

陈惠明 张 静 何志国 主 编

朱正国 郝耀军 王兰春

副主编

郝 莉 何春燕

王建国 主 审

参 审
陈惠明

参 审
郝 莉
郝耀军
刘 颖
王 雷
王 雷
李 佳

出版发行：中国铁道出版社（100054，北京市西城区右安门西街8号）

网 址：http://www.tdpress.com

印 刷：北京新魏印刷厂

版 次：2015年1月第1版

开 本：787mm×1092mm 1/16 印张：11.25 字数：283千

印 数：1—2 000册

书 号：ISBN 978-7-113-19816-8

定 价：23.00元

中国铁道出版社
CHINA RAILWAY PUBLISHING HOUSE

内 容 简 介

本书为《C 语言程序设计》(陈惠明、赵青杉主编,中国铁道出版社出版)的配套教材,内容安排上在强化主教材知识点实践教学的基础上,给出综合练习。第 1 章除介绍 Visual C++ 6.0 环境外,还重点介绍了程序的调试技术;第 2~6 章(第 5 章除外)按“读程序画流程图”——“看流程图写程序”——“程序改错”——“程序设计”——“综合练习”等环节组织内容,使学生从“扶着走”到“独立走”顺利过渡;综合练习题的安排循序渐进,突出 C 语言的实用性和灵活性,有助于提高学生对实际问题的抽象描述能力。附录包括全国计算机等级考试大纲及 C 语言二级考试部分真题,有利于读者练习提高和备考。另外,本书对主教材每章后的习题做了详细解答,是一本针对性很强的辅导教材。

本书适合作为高等院校应用型本科学生 C 语言上机实验、等级考试培训和综合课程设计的辅导教材,也可作为计算机专业和相关专业高职学生、自考学员和教师的辅导教材。

图书在版编目(CIP)数据

C 语言程序设计实践教程/陈惠明,张静,何志国主编. —

北京:中国铁道出版社,2015.1

全国高等院校计算机教育规划教材

ISBN 978-7-113-19516-8

I. ①C… II. ①陈… ②张… ③何… 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(2014)第 258094 号

书 名: C 语言程序设计实践教程

作 者: 陈惠明 张 静 何志国 主编

策 划: 周 欣

读者热线: 400-668-0820

责任编辑: 周海燕 彭立辉

封面设计: 刘 颖

封面制作: 白 雪

责任校对: 汤淑梅

责任印制: 李 佳

出版发行: 中国铁道出版社(100054,北京市西城区右安门西街 8 号)

网 址: <http://www.51eds.com>

印 刷: 北京新魏印刷厂

版 次: 2015 年 1 月第 1 版

2015 年 1 月第 1 次印刷

开 本: 787mm×1092mm 1/16 印张: 11.25 字数: 263 千

印 数: 1~2 000 册

书 号: ISBN 978-7-113-19516-8

定 价: 23.00 元

版权所有 侵权必究

凡购买铁道版图书,如有印制质量问题,请与本社教材图书营销部联系调换。电话:(010) 63550836

打击盗版举报电话:(010) 51873659

全国高等院校计算机教育规划教材

编审委员会

主 任：沈复兴

副主任：胡金柱 焦金生 严晓舟

委 员：（按姓氏笔画排序）

王建国 叶俊民 曲建民

朱小明 刘美凤 孙 波

李雁玲 别荣芳 邹显春

沈 洁 罗运伦 秦绪好

詹国华

编 者

2014年8月

本书结合编者多年的教学经验,遵照学生学习C语言的规律,参考《中国高等院校计算机基础教育课程体系2008》中的要求编写而成。本书具有以下特点:

1. 循序渐进,降低学习难度

本书按照“扶着走”到“独立走”的思路编排内容,设定了“读程序画流程图”—“看流程图写程序”—“程序改错”—“程序设计”—“综合练习”等环节(第1、5、7章除外),有目的地使学生先从读懂程序入手,进而自己动手写程序,再到上机调试程序、改错,最后自己独立设计程序求解问题。

2. 案例翔实,内容丰富

本书既有传统教材中的经典案例,也有新编制的贴近生活的趣味案例;既有解决初、高中数学问题的案例,也有面向全国计算机等级考试问题的案例。大多数案例均配有流程图或屏幕截图等,图文并茂,便于问题的说明。

3. 注重实践能力培养

本书注重学生上机实践能力的培养,以Visual C++ 6.0为开发环境,对程序的编写和调试过程进行了详细讲解,为今后更高级开发技术的学习打下良好的基础。

本书实例程序全部在Visual C++ 6.0环境下调试通过,在程序设计环节严格遵循编写程序的流程,对培养读者的自主编程能力有很大的帮助。各程序都配有详细的注释说明,为书写规范程序做了良好示范。

4. 强调编程方法和技巧

本书强调结构化程序设计的方法和编程技巧,注重流程图的作用,强调编程思路和程序设计方法,帮助提高学生的程序设计能力。另外,上机调试也是学习程序设计的重要内容和基本要求,通过调试发现和改正程序中的错误对提高编程水平非常有效,为此,本书还介绍了一些调试技巧。

在内容编排的顺序方面,本书结构与C语言程序设计课程体系相吻合,用大量的实例帮助读者掌握程序设计的思想,学会程序设计的方法,做到循序渐进、系统学习、广泛实践。

本书由陈惠明、张静、何志国任主编,朱正国、郝耀军、王兰春、鄢莉、何春燕任副主编,王建国主审。具体编写分工:第1章由何志国编写,第2章由郝耀军编写,第3章由张静编写,第4章由朱正国编写,第5章由王兰春编写,第6章由鄢莉编写,第7章由何春燕编写,最后由陈惠明统稿。

本书适合作为高等院校应用型本科学生学习C语言程序设计的辅导教材,也可作为高职高专教育的辅导教材。

由于时间仓促,编者水平有限,书中难免存在疏漏与不足之处,敬请读者批评指正。

编者

2014年8月

CONTENTS

目录

第 1 章 C 语言基础	1
1.1 C 语言程序的上机操作	1
1.2 C 程序的调试	5
1.2.1 语法错误调试	5
1.2.2 逻辑错误调试	7
1.2.3 其他调试技术	10
1.3 读程序写结果	11
1.3.1 相加溢出	11
1.3.2 字符输出	12
1.3.3 算术表达式求值	12
1.3.4 复合赋值表达式求值	13
1.3.5 自增自减运算求值	14
1.4 程序设计	14
综合练习	15
第 2 章 控制结构程序实例	16
2.1 读程序画流程图	16
2.1.1 分段函数求值	16
2.1.2 判断输入字符的类型	17
2.1.3 歌星大奖赛	19
2.2 看流程图写程序	21
2.2.1 身高预测	21
2.2.2 简易计算器	22
2.2.3 趣味数学题	24
2.3 程序改错	25
2.3.1 计算整数位数	25
2.3.2 四方定理	27
2.4 程序设计	30
2.4.1 加工资	30
2.4.2 最大公因子	31
2.4.3 调和级数的项数	32
2.4.4 一个奇异的三位数	33
2.4.5 求定积分	34
综合练习	35

第 3 章 数组与指针程序实例	37
3.1 读程序画流程图	37
3.1.1 求成绩平均分	37
3.1.2 求逆序	38
3.1.3 数据加密	40
3.2 看流程图写程序	42
3.2.1 计算字符串中空格数目	42
3.2.2 求矩阵鞍点	43
3.2.3 找素数	46
3.3 程序改错	49
3.3.1 改写算术表达式	49
3.3.2 求矩阵的转置矩阵	50
3.3.3 判断是否回文	51
3.4 程序设计	53
3.4.1 在有序序列中插数	53
3.4.2 杨辉三角形	55
3.4.3 合法标识符判定	56
3.4.4 求 100 的阶乘	58
综合练习	60
第 4 章 函数程序实例	61
4.1 读程序画流程图	61
4.1.1 求任意 3 个整数绝对值的和	61
4.1.2 宏#include 的应用	62
4.2 看流程图写程序	63
4.2.1 整数反向输出	63
4.2.2 字符串反向输出	65
4.3 程序改错	66
4.4 程序设计	68
4.4.1 统计字符串中的单词数	68
4.4.2 函数指针的应用	70
综合练习	71
第 5 章 结构体与共用体程序实例	72
5.1 结构体应用实例——倒计时器	72
5.2 单链表的基本操作	75
5.2.1 链表的创建	75
5.2.2 链表元素的输出	77
5.2.3 求链表长度	77
5.2.4 链表元素的查找	78

5.2.5	链表元素的插入	78
5.2.6	链表元素的删除	80
5.3	单链表的应用	80
5.3.1	集合的并集	81
5.3.2	集合的交集	81
5.3.3	集合的补集	82
5.3.4	集合操作应用	83
	综合练习	84
第 6 章	文件操作程序实例	85
6.1	读程序画流程图	85
6.1.1	成绩查询	85
6.1.2	成绩录入	87
6.2	看流程图写程序	90
6.2.1	统计文本文件中的单词数	90
6.2.2	人事管理信息系统	92
6.3	程序改错	96
6.4	程序设计	97
6.4.1	统计素数写入文件	97
6.4.2	字符串加密	98
	综合练习	100
第 7 章	主教材习题参考答案	101
7.1	C 语言基础知识习题参考答案	101
7.2	算法习题参考答案	102
7.3	C 程序的控制结构习题参考答案	104
7.4	数组与指针习题参考答案	108
7.5	函数习题参考答案	112
7.6	结构体与共用体习题参考答案	115
7.7	文件操作习题参考答案	119
附录 A	全国计算机等级考试二级 C 语言大纲	125
附录 B	全国计算机等级考试二级 C 语言模拟题	128
附录 C	二级 C 语言模拟题参考答案及解析	164
	参考文献	168

第1章 C语言基础

学习目标

- 熟悉在 Visual C++ 6.0 环境中调试 C 程序的步骤和方法。
- 熟悉程序调试的一般方法。
- 掌握 C 语言中各种运算符的运算规则和表达式的求值规则。
- 掌握格式化输入/输出函数和字符输入/输出函数的使用方法。

Visual C++ 6.0 (以下简称 VC++ 6.0) 是微软公司按 ANSI C++ 标准开发的, 面向 Windows 编程的 C++ 集成开发工具。它不仅支持 C++ 语言的编程, 也兼容 C 语言的编程。本章主要介绍如何在 VC++ 6.0 下运行和调试 C 语言程序。

1.1 C 语言程序的上机操作

C 是编译型语言, 设计好一个 C 源程序后, 需要经过编译、连接生成可执行的程序文件, 然后执行。一个 C 程序的具体开发步骤如下:

1. 启动程序

启动 VC++ 6.0, 界面如图 1-1 所示。

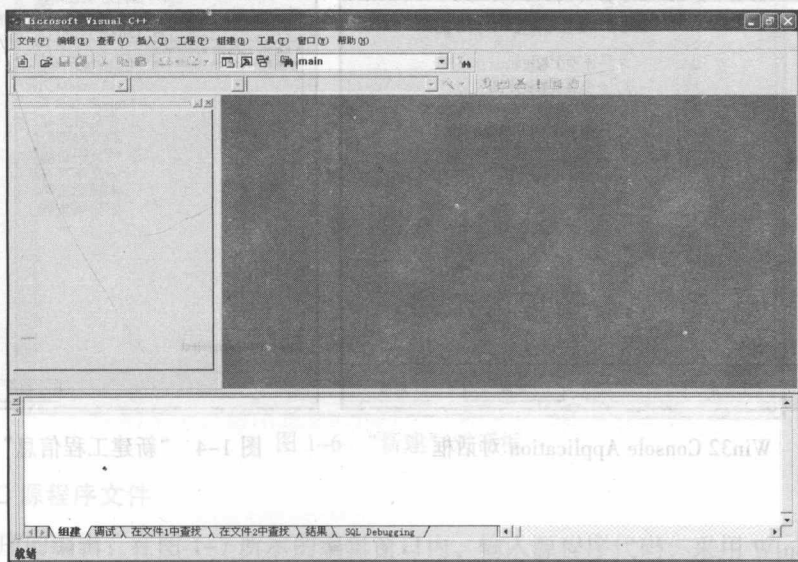


图 1-1 VC++ 6.0 环境窗口

2. 建立工程

(1) 选择“文件”→“新建”命令，弹出图 1-2 所示的“新建”对话框。



图 1-2 “新建”对话框

(2) 在“工程”选项卡中选择 Win32 Console Application 类型的工程，在“位置”文本框中选择路径“D:\MYPROG\”，在“工程名称”文本框中输入工程名称 myproject，单击“确定”按钮，弹出 Win32 Console Application 对话框，如图 1-3 所示。

(3) 在 Win32 Console Application 对话框中选择“一个空工程”单选按钮，然后单击“完成”按钮，弹出“新建工程信息”对话框，如图 1-4 所示。在“新建工程信息”对话框中单击“确定”按钮，完成工程的创建，弹出 myproject 工程窗口，如图 1-5 所示。

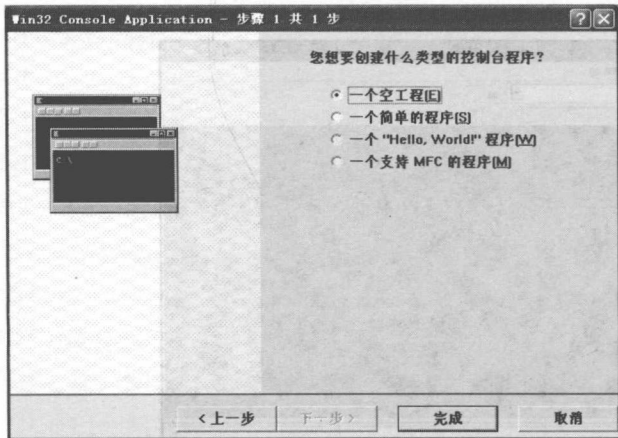


图 1-3 Win32 Console Application 对话框

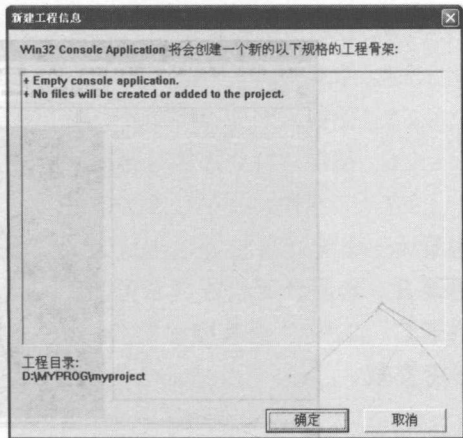


图 1-4 “新建工程信息”对话框

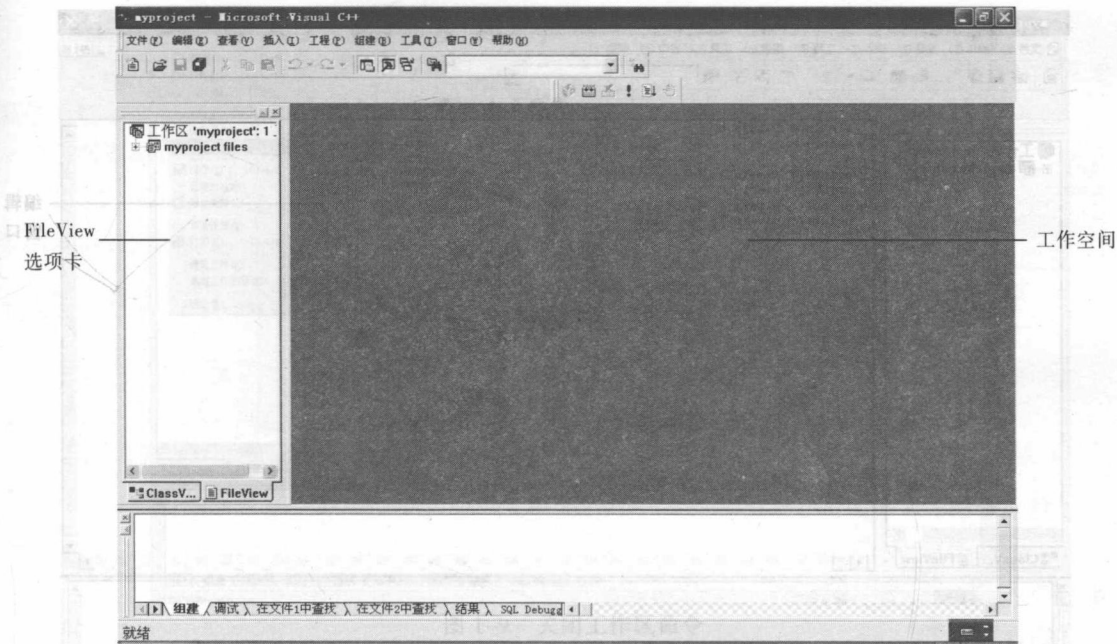


图 1-5 myproject 工程窗口

3. 新建 C 语言源程序文件

在菜单栏中选择“文件”→“新建”命令，弹出“新建”对话框，如图 1-6 所示。在“文件”选项卡中选择 C++ Source File，并在“文件名”文本框输入文件名 hello.c（读者可自行命名，若是 C 语言程序，文件的扩展名为.c；若是 C++程序，则扩展名为.cpp），单击“确定”按钮，完成新建 C 源程序文件。

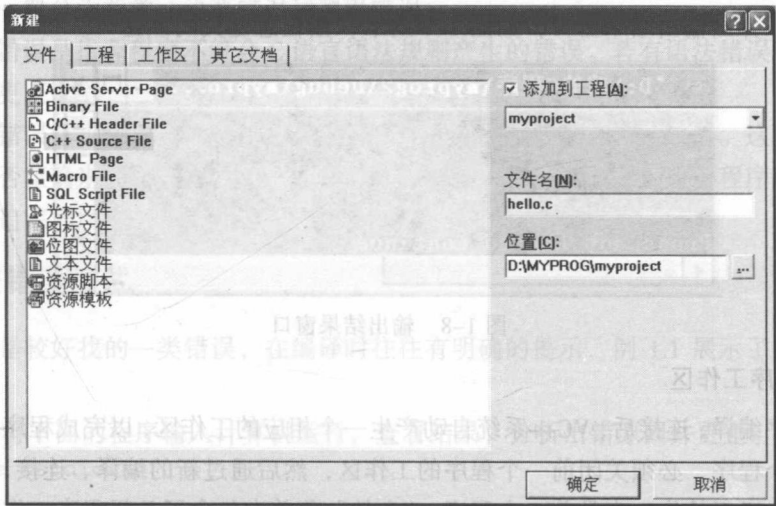


图 1-6 “新建”对话框

4. 编辑 C 源程序文件

(1) 源程序的编辑：在图 1-7 所示的编辑窗口内，输入源程序代码，采用 Windows 的编辑操作方法。

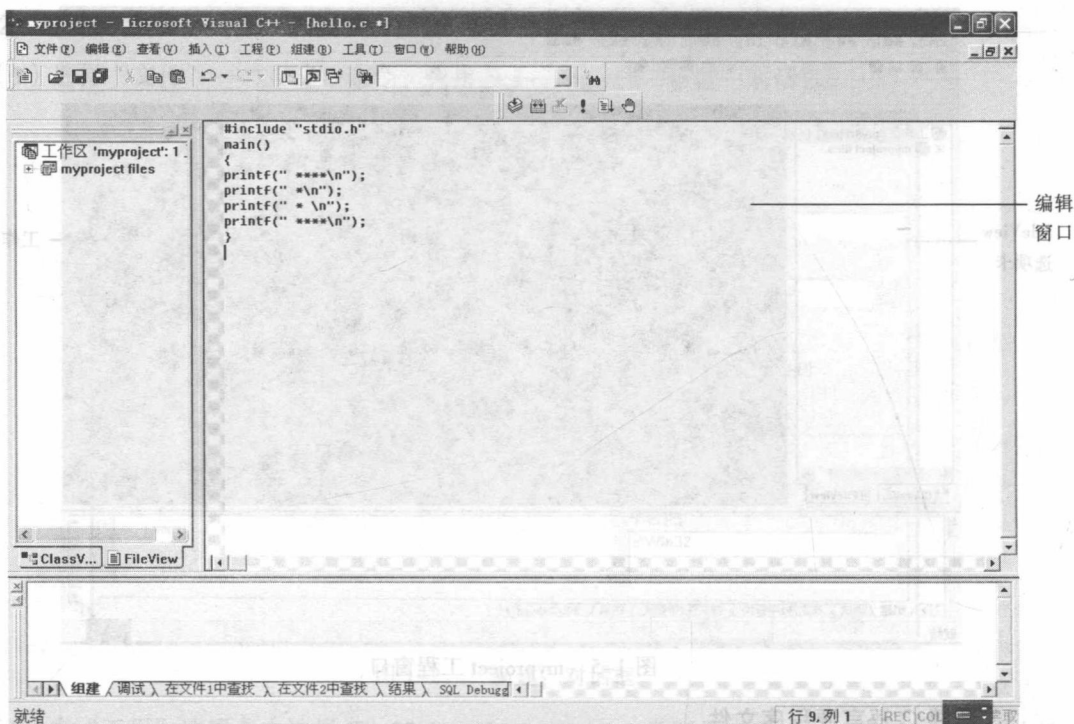


图 1-7 C 源程序编辑窗口

(2) 源程序的存储: 选择“文件”→“保存”命令或按【Ctrl+S】组合键保存该文件。

5. 编译运行

在 VC++ 6.0 环境下, 选择“组建”→“执行”命令或按【Ctrl+F5】组合键 (或直接单击 ! 按钮) 执行文件, 弹出输出结果窗口, 如图 1-8 所示, 按任意键关闭该窗口。

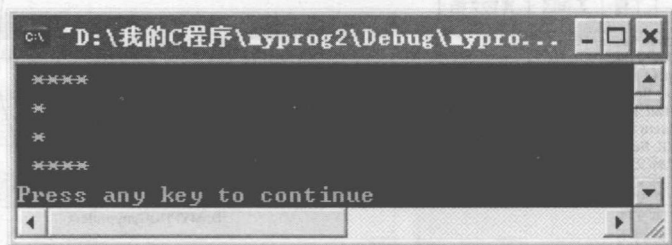


图 1-8 输出结果窗口

6. 关闭程序工作区

当一个程序编译、连接后, VC++ 系统自动产生一个相应的工作区, 以完成程序的运行和调试。若要执行另一个程序, 必须关闭前一个程序的工作区, 然后通过新的编译、连接, 产生新程序的工作区, 否则, 运行的将一直是前一个程序。“文件”菜单中包含了关闭程序工作区的命令, 如图 1-9 所示。

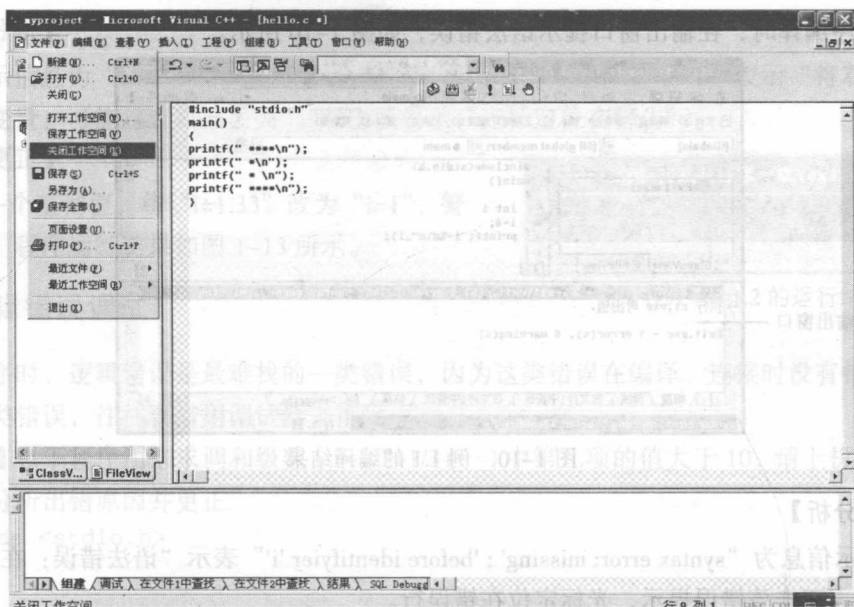


图 1-9 关闭工作区命令

1.2 C程序的调试

第一次编写的程序有错误是很正常的，即使是熟练的专业程序员也难以保证所写的程序没有错误，所以如何能将这些错误快速查找出来并进行修改是非常重要的。在程序中查找错误并修改错误的过程就是调试，调试技术是编程的一项基本技能。

程序错误一般分为两类：语法错误和逻辑错误。

(1) 语法错误是指源程序不符合 C 语言语法规则产生的错误。若有语法错误，则无法通过编译，必须进行更正。

(2) 逻辑错误是指程序运行后得到的结果与设计时期的结果不符的错误。这种错误需通过分析运行结果是否正确来判定，往往要重新分析算法，检验其正确性。在检查程序的过程中，可通过设置断点、追踪程序的运行过程来发现逻辑错误。

1.2.1 语法错误调试

语法错误是较好找的一类错误，在编译时往往有明确的提示，例 1.1 展示了排除语法错误的详细过程。

【例 1.1】将下面的程序输入计算机运行，查看结果，分析出错原因并更正。

```
#include <stdio.h>
main()
{
    int i
    i=0;
    printf("i=%d\n",i);
}
```

上述程序编译时,在输出窗口提示语法错误,如图 1-10 所示。



图 1-10 例 1.1 的编译结果

【错误分析】

错误提示信息为“syntax error: missing ';' before identifier 'i'”表示“语法错误:在标识符 i 前丢失分号 ';'”。双击该错误提示,光标定位在错误行。

【错误更正】

在“int i;”后添加英文标点“;”,重新编译程序,无错误、警告提示信息,表明上述语法错误被排除,其运行结果如图 1-11 所示。

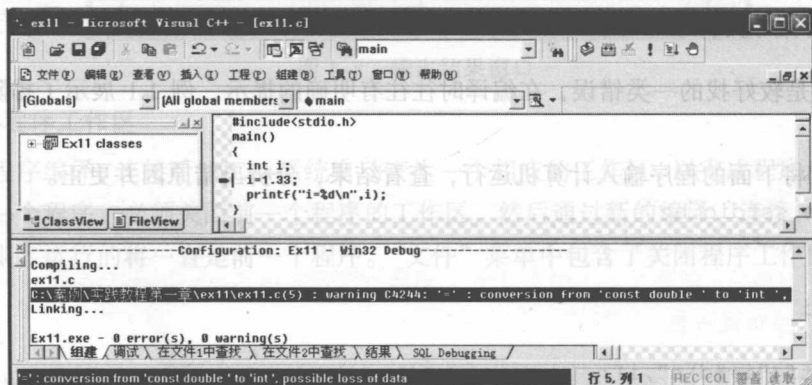


编译时的提示信息除了上面的错误信息外,往往还有警告信息。

【例 1.2】将下面的程序输入计算机运行,查看结果,分析出错原因并更正。

```
#include <stdio.h>
main()
{
    int i;
    i=1.33;
    printf("i=%d\n",i);
}
```

上述程序编译时,在输出窗口提示警告信息,如图 1-12 所示。



【错误分析】

警告提示信息为“conversion from 'const double' to 'int':possible loss data”表示“将双精度常量赋值给整型变量：可能导致数据丢失”。双击该提示，光标定位在警告行。

【错误更正】

给i赋一个整型值，将“i=1.33”改为“i=1”，警告信息消除，程序运行结果如图 1-13 所示。

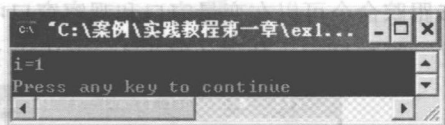


图 1-13 例 1.2 的运行结果

1.2.2 逻辑错误调试

调试程序时，逻辑错误是最难找的一类错误，因为这类错误在编译、连接时没有错误提示信息。对于这类错误，往往要借用调试器来排除。

【例 1.3】以下程序用来求调和级数 $(1+1/2+1/3+\dots)$ 中第几项的值大于 10，请上机调试运行，查看结果，分析出错原因并更正。

```
#include <stdio.h>
main()
{
    int n;
    double sum;
    sum=0;
    for(n=1;sum<10;n++)
        sum=sum+1/n;
    printf("n=%d\n",n);
}
```

上述程序无语法错误，可以通过编译，但在执行时进入死循环状态。

调试时，需要利用调试器暂停程序运行，观察程序运行过程中变量值的变化情况，用以诊断程序中的错误。

方法一：使程序执行到光标所在行暂停。其步骤如下：

- (1) 在需暂停的行“sum=sum+1/n;”上单击，定位光标。
- (2) 选择“调试”→“开始调试”→“Run to Cursor”命令，或按【Ctrl+F10】组合键，程序执行到光标所在行暂停，如图 1-14 所示。

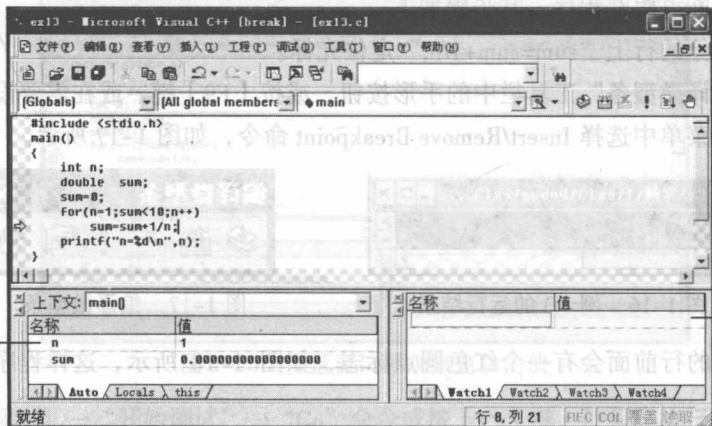


图 1-14 程序执行到光标所在行暂停

(3) 在变量窗口观察变量 n 和 sum 的当前值。

(4) 增加观察变量, 在观察窗口的“名称”框中输入表达式 $1/n$ 。

(5) 选择“调试”→“Step Into”命令或按【F11】键, 单步跟踪程序的执行, 每执行一次单步跟踪命令可以在变量窗口和观察窗口中观察变量值的变化情况, 观察发现变量 n 的值在不断递加, 而变量 sum 的值始终为 1, 表达式 $1/n$ 的值始终为 0, 如图 1-15 所示。

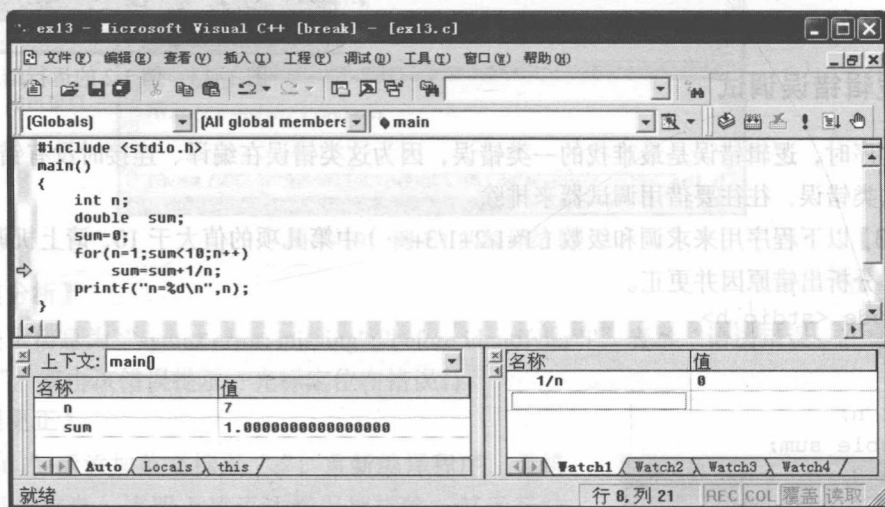


图 1-15 单步跟踪变量窗口和观察窗口的值

(6) 选择“调试”→“Stop Debugging”命令, 结束程序调试。

【错误分析】

在上述程序中, 随着循环变量 n 的增加, sum 变量始终为 1, 程序进入死循环状态; 对表达式 $1/n$, 当 $n=1$ 时值为 1, 随着 n 的增加, 其值变为 0。究其原因, n 定义为整型, 进行整除运算时, 在 $n>1$ 的情况下, 表达式 $1/n$ 的结果始终为 0, 所以需要将“ $1/n$ ”改为“ $1.0/n$ ”, 使其在计算时隐式转换为浮点型。

【错误更正】

将“ $sum=sum+1/n;$ ”改为“ $sum=sum+1.0/n;$ ”, 修改后程序运行结果如图 1-16 所示。

方法二: 设置断点跟踪程序, 其步骤如下:

(1) 在需要暂停的行上“ $sum=sum+1/n;$ ”定位光标。

(2) 单击“编译微型条”工具栏中的手形按钮, 或按【F9】键, 或在需要设置断点的行上右击, 在弹出的快捷菜单中选择 Insert/Remove Breakpoint 命令, 如图 1-17 所示。

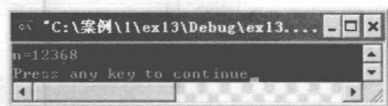


图 1-16 例 1.3 的运行结果



图 1-17 编译微型条

被设置了断点的行前面会有一个红色圆点标志, 如图 1-18 所示, 这样程序执行到断点处将会暂停。



图 1-18 设置断点后的程序

(3) 选择“调试”→“开始调试”→“Go”命令或按【F5】键，使程序执行到断点处暂停，如图 1-19 所示。

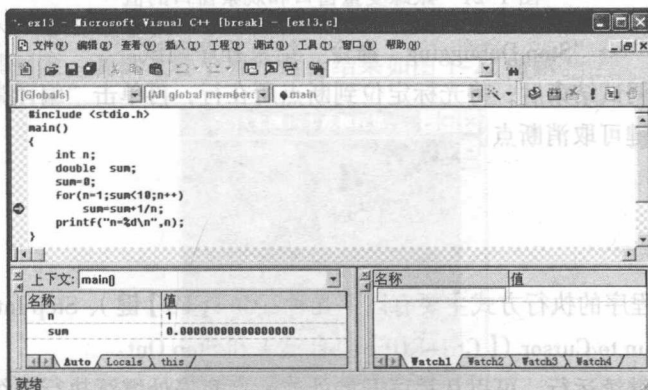


图 1-19 执行到断点处暂停

(4) 观察变量窗口 n 和 sum 的值，并在观察窗口的“名称”框中输入表达式 $1/n$ ，观察其值的变化，如图 1-20 所示。



图 1-20 变量窗口和观察窗口的值

(5) 选择“调试”→“开始调试”→“Go”命令或按【F5】键，使程序执行到下一断点处暂停，每暂停一次可以看到变量窗口和观察窗口值的变化情况，变量变化情况同方法一，如图 1-21 所示。