

高等院校计算机应用系列教材

C 语言程序设计实践教程

(第二版)(微课版)

杨宏霞 赵彩 主 编
江若玫 刘宁 副主编

清华大学出版社
北 京

内 容 简 介

本书是赵彩、杨宏霞主编的《C语言程序设计(第二版)(微课版)》的配套实践指导教材,旨在帮助初学者突破“理论易、实践难”的学习瓶颈,通过系统训练强化对C语言核心概念的理解,掌握用计算机解决问题的基本方法。

本书内容分为四部分:“基础实验与算法积累”部分包含10个实验,每个实验设置基础练习、程序填空、程序改错和程序设计四种题型,循序渐进夯实编程基础;“算法实践与模拟训练”部分精选全国计算机等级考试二级C语言真题及“蓝桥杯”软件大赛题目,供读者在完成基础训练后进行进阶挑战;“课程专题设计训练”部分提供4个难度递进的综合项目,涵盖程序功能简介、训练目标、程序功能分析及参考代码;“参考答案及解析”部分为前两部分的习题提供详尽解答。重点内容配有讲解视频,辅助自主学习。

书中所有程序均在Visual C++ 6.0和Dev-C++ 5.10环境下调试通过,确保代码可复现。本书结构合理、案例翔实、讲解清晰,既适合作为高校计算机专业本科生的实验教材,也适合程序设计初学者及希望突破编程实践瓶颈的自学者使用。

本书配套的电子课件、实验报告模板和实例源文件扫描封底或前言中的二维码即可获取,扫描正文中的视频二维码可以直接观看所讲知识点的教学视频。

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

版权所有,侵权必究。举报:010-62782989, beiqinquan@tup.tsinghua.edu.cn。

图书在版编目(CIP)数据

C语言程序设计实践教程:微课版/杨宏霞,赵彩

主编.--2版.--北京:清华大学出版社,2026.6.

(高等院校计算机应用系列教材).--ISBN 978-7-302

-71717-1

I. TP312.8

中国国家版本馆CIP数据核字第2026Z73X80号

责任编辑:胡辰浩

封面设计:高娟妮

版式设计:妙思品位

责任校对:马遥遥

责任印制:刘 菲

出版发行:清华大学出版社

网 址: <https://www.tup.com.cn>, <https://www.wqxuetang.com>

地 址:北京清华大学学研大厦A座

邮 编:100084

社 总 机:010-83470000

邮 购:010-62786544

投稿与读者服务:010-62776969, c-service@tup.tsinghua.edu.cn

质 量 反 馈:010-62772015, zhiliang@tup.tsinghua.edu.cn

印 装 者:北京同文印刷有限责任公司

经 销:全国新华书店

开 本:185mm×260mm

印 张:11.5

字 数:266千字

版 次:2021年7月第1版

2026年7月第2版

印 次:2026年7月第1次印刷

定 价:69.80元

前 言

掌握和灵活运用C语言，是读者后续学习操作系统、计算机网络、编译原理、数据结构与算法等课程的重要前提。学习C语言能够建立对计算机系统的本质认知，C语言像一台显微镜——通过指针操作，可以深度理解计算机如何管理内存与数据；通过直接暴露底层硬件交互细节，可以清晰理解函数调用的底层代价。C语言这种看似“不便利”的特性，恰恰是它作为系统级编程语言的核心价值所在，也是培养系统级编程思维的绝佳工具。这种“痛苦”背后，隐藏着计算机底层运作的深刻逻辑。当下流行的Python、JavaScript等语言封装了太多细节，虽能快速产出，却容易让人产生“编程即调用库函数”的错觉。高楼起于深基，掌握C语言后再学习其他语言，您将拥有俯视技术栈的洞察力，避免疲于追逐框架。

本书共分为四部分。“基础实验与算法积累”部分包含10个实验，我们为其中的每个实验设计了基础练习、程序填空、程序改错和程序设计四种题型，并且是按照配套的《C语言程序设计(第二版)(微课版)》一书的知识结构进行编写的。“算法实践与模拟训练”部分参考全国计算机等级考试二级C语言考试真题和全国“蓝桥杯”软件大赛真题编写完成，读者可在系统地完成基础练习之后进行算法实践并完成复杂练习。上述所有题目的参考答案及解析可查阅第IV部分“参考答案及解析”。“课程专题设计训练”部分包含4个难度递进的题目，我们分别给出了每个题目的程序功能简介、训练目标、程序功能分析和参考代码。

本书第I部分的实验1～实验5以及第II部分和第III部分由杨宏霞编写；第I部分的实验6由赵彩编写；第I部分的实验7和实验8由刘宁编写；第I部分的实验9和实验10由江若玫编写。全书的统稿工作由杨宏霞完成。

本书内容丰富，既包含配套实验，又包含多套算法实践与模拟训练题。本书所有程序均在Visual C++ 6.0和Dev-C++ 5.10环境下调试通过。重点内容配有讲解视频，辅助自主学习。在学习任何一种程序设计语言时，除了通过多看书来理解基本概念，还需要不断练习编写与调试代码。多多练习代码编写、程序调试和程序改错是学习程序设计语言的唯一途径，除此之外没有其他捷径可走。

路虽远，行则将至；事虽难，做则必成。只要有愚公移山的志气、滴水穿石的毅力，脚踏实地，埋头苦干，就能积跬步以至千里。最后祝愿本书的读者，脚踏实地地从C语言基础开始学起，迈出把宏伟目标变为美好现实的第一步！

由于作者水平有限，书中难免有不足之处，恳请专家和广大读者批评指正。在编写本书的过程中参考了相关文献，在此向这些文献的作者深表感谢。我们的电话是010-62796045，邮箱是992116@qq.com。

本书配套的电子课件、实验报告模板和实例源文件扫描下面的二维码即可获取，扫描正文中的视频二维码可以直接观看所讲知识点的教学视频。



编 者

作于西安交通大学城市学院

2026年3月

目 录

第 I 部分 基础实验与算法积累	1	模拟训练 4	82
实验 1 开始编写 C 语言程序	1	模拟训练 5	85
实验 2 数据类型、运算符及 表达式	11	模拟训练 6	87
实验 3 常用的输入输出函数	17	第 III 部分 课程专题设计训练	91
实验 4 选择结构	22	题目 1 简单计算器小程序	91
实验 5 循环结构	29	题目 2 口算练习小程序	95
实验 6 数组	36	题目 3 文件加密小程序	102
实验 7 函数	41	题目 4 通讯录管理小程序	109
实验 8 指针	47	第 IV 部分 参考答案及解析	121
实验 9 结构体、共用体和枚举	54	“基础实验与算法积累”部分的 参考答案及解析	121
实验 10 文件	64	“算法实践与模拟训练”部分的 参考答案及解析	165
第 II 部分 算法实践与模拟训练	75	参考文献	175
模拟训练 1	75		
模拟训练 2	78		
模拟训练 3	80		

第 I 部分

基础实验与算法积累

本书第I部分的内容参照赵彩、杨宏霞主编的《C语言程序设计(第二版)(微课版)》一书的知识结构编写而成，目标是解决初学者在将理论应用于实践时面临的各种困难，并通过实践加强读者对理论知识的理解，同时积累使用计算机解决问题的各种基本算法。这部分包含10个实验，我们为其中的每个实验设计了基础练习、程序填空、程序改错和程序设计四种题型，所有题目的参考答案和解析请查阅第IV部分。

进行以下实验的前提条件是计算机中已经安装Visual C++ 6.0编译环境或其他可以用来编写C语言程序的编译软件。本书所有程序都已经在Visual C++ 6.0和Dev-C++ 5.10环境下调试并编译通过。

实验 1 开始编写 C 语言程序

一、实验目的

- 掌握在Visual C++ 6.0编译环境中编写C语言程序的步骤。
- 学会查看编译器给出的错误提示，并了解常见错误。
- 了解调试程序的步骤，并逐步熟练掌握该步骤。

二、实验内容

1. 基础练习：编写第一个 C 语言程序并熟悉 Visual C++ 6.0 编译环境

步骤一：启动Visual C++ 6.0。



教学视频

方法一：如果桌面上显示有图标，双击这个图标。

方法二：打开“开始”菜单，选择“程序”→Microsoft Visual Studio 6.0 →Microsoft Visual C++ 6.0，进入Visual C++ 6.0编译环境，如图1-1所示。

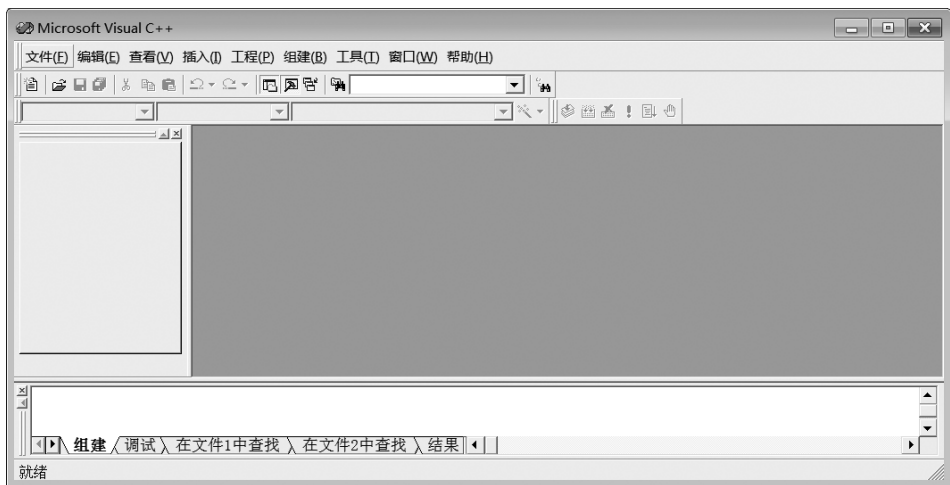


图 1-1 Visual C++ 6.0 编译环境

步骤二：新建源文件。

选择“文件”菜单中的“新建”命令，弹出“新建”对话框，如图1-2所示。



图 1-2 “新建”对话框

步骤三：设置源文件的名称和保存路径。

在“新建”对话框中打开“文件”选项卡，选择C++ Source File选项，在右侧的“文件名”文本框中输入源文件的名称，例如1-1.c，如图1-3所示。“位置”文本框中显示了源文件的默认保存位置，单击右边的按钮可以修改保存位置。设置完毕后，单击“确定”按钮，源文件就创建好了。

[注意] 在图1-3的“文件名”文本框中，请务必在1-1的后面输入.c以确保源文件的

扩展名为.c, 否则编译器会默认将.cpp设置为扩展名。.cpp是C++源文件的扩展名, C和C++是两种不同的语言。虽然C++兼容C语言的绝大部分语法特性, 但还是会经常出现一些编译错误。

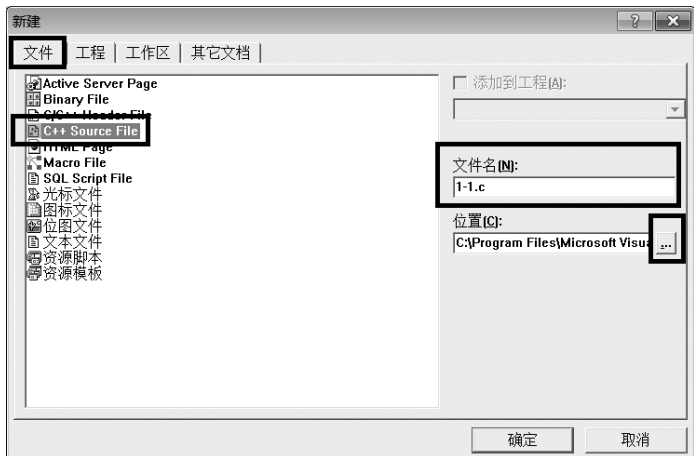


图 1-3 创建 C 源文件

步骤四：编写源程序。

如图 1-4 所示, 在编辑区域输入程序代码。观察图 1-4 中的标题栏部分, 注意方括号中显示了当前正在编辑的源文件的名称和扩展名, 最后的*表示当前源文件有改动但未保存, 在单击“保存”按钮之后, *就会消失。

[注意] 如果在创建源文件时忘记写扩展名, 那么通过观察标题栏可以看到, 此时扩展名为.cpp。选择“文件”菜单中的“另存为”命令, 可将当前源文件另存为扩展名为.c的源文件。

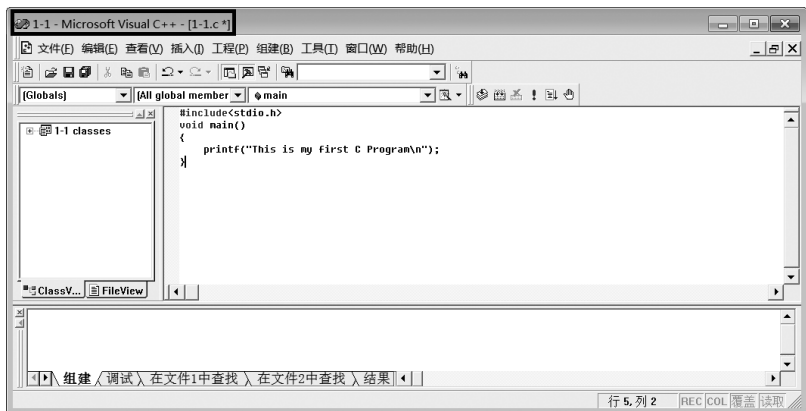


图 1-4 编写源程序

步骤五：运行程序。

方法一：使用快捷工具栏。

单击快捷工具栏    中的  按钮, 即可对源程序进行编译。如果不存在编译错

误,可单击按钮进行链接。如果不存在编译和链接错误的话,按钮会从灰色变成高亮显示。单击按钮,即可运行程序。

方法二:使用菜单中的“组建”菜单。

选择“组建”菜单中的“编译”命令,即可对源程序进行编译。如果不存在编译错误,请继续选择“组建”菜单中的“组建”命令进行链接。如果不存在编译和链接错误,请继续选择“组建”菜单中的“执行”命令,即可运行程序。

方法三:使用快捷键。

具体的步骤与前两种方法一致,快捷键与功能的对应关系如下。

- 编译: Ctrl+F7
- 链接: F7
- 执行: Ctrl+F5

[注意] 不需要强行记住这些快捷键,打开菜单栏中对应的菜单命令,如果有的话,快捷键会标记在对应的菜单命令的后面。不同编译器的快捷键设置也不同,使用时请注意观察。

步骤六:关闭程序。

调试完一个程序后,在编写新的程序之前,一定要使用“文件”菜单中的“退出”命令关闭原来的程序,否则后面编译或运行的将总是原来的程序。

[注意] 使用Visual C++ 6.0时不允许跳过关闭程序这个步骤。其他编译器(如Dev-C++ 5.10或C-Free 5.0)则允许,因为每个源文件都被放在了单独的选项卡中。

2. 程序改错

C语言作为结构化程序设计的基石,其语法规则的严谨性既是保障程序可靠性的前提,也是初学者面临的首要挑战。语法错误如同程序世界的“交通违章”——漏写分号、错用括号、误置逗号等细微疏忽,都会使编译器在词法分析与语法分析阶段拦截代码,无法生成可执行文件。

为了写出正确的程序,就必须严格遵守C语言的语法规则。漏掉一个该有的逗号,或添加不该有的分号和空格,都将导致编译器无法将程序的源代码转换成机器代码。即使有多年编程经验,程序中也很容易出现语法错误。值得庆幸的是,这些语法错误在编译或链接阶段通常能被编译器精准识别并指出位置,为开发者排查问题提供了便利。有些程序虽存在逻辑错误,却可以正常地完成编译和链接过程,只不过会在运行过程中不定时出错,这类错误往往需要开发者通过跟踪调试的方法进行排除。

建议养成“编译—排错—再编译”的迭代习惯,并尽最大努力阅读编译器报错信息,这是区分初学者与熟练开发者的重要标志。

步骤一:新建源文件。

新建源文件1-2.c,输入下列包含语法错误的程序代码,然后编译程序。

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



教学视频

```

void mian()
{
    printf("C program1")
    printf("C program2");
}

```

步骤二：发现错误。

观察屏幕底部的调试信息窗口，如图 1-5 所示。由于源程序有语法错误，因此经过编译后，调试信息窗口中将显示“1-2.obj - 1 error(s), 0 warning(s)”。

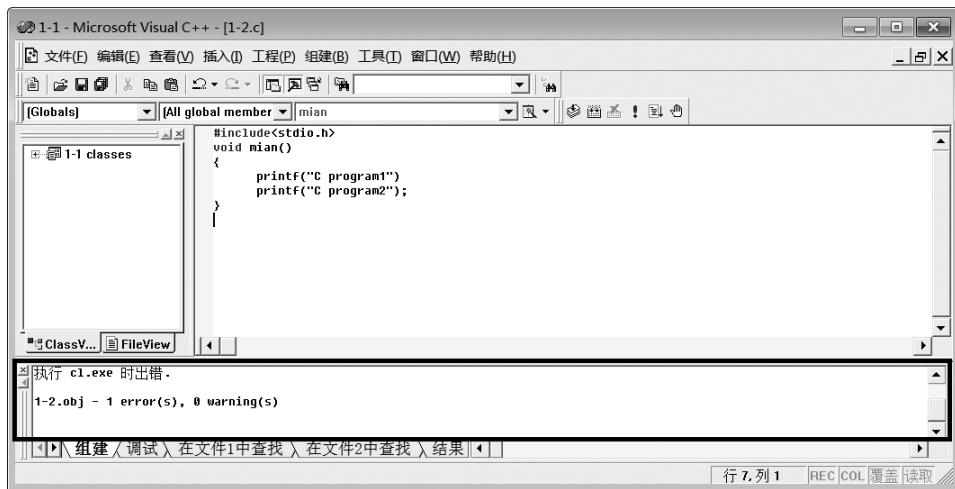


图 1-5 源程序有语法错误时的调试信息窗口

[注意] error(s)前面的数字代表错误个数，warning(s)前面的数字代表警告个数。编译系统能够检查出的语法错误分为两类：一类是致命错误，用error表示，程序如果有此类错误，就无法通过编译，从而无法形成目标程序，因此必须改正；另一类是轻微错误，用warning表示，此类错误不影响目标程序和可执行程序的生成，但可能影响运行结果。

步骤三：查看错误。

单击调试信息窗口中的空白区域，上下滚动窗口，即可看到更详细的错误提示，如图 1-6 所示。

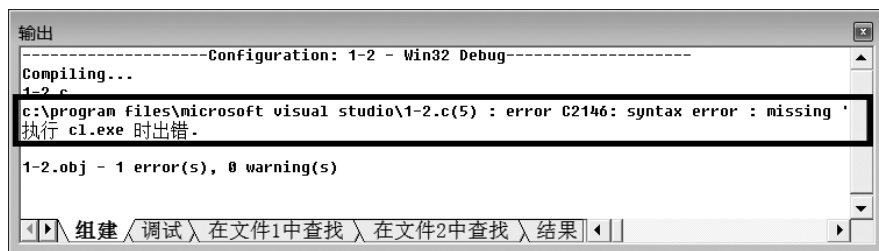


图 1-6 调试信息窗口中的错误提示

[注意] 在调试信息窗口的上侧边缘按住鼠标左键并上下拖动, 即可改变调试信息窗口的高度。

步骤四: 定位错误。

双击调试信息窗口中的错误提示, 在本例中也就是双击“c:\program files\microsoft visual studio\1-2.c(5) : error C2146: syntax error : missing ';' before identifier 'printf'”, 编译器给出的错误提示如图 1-7 所示。

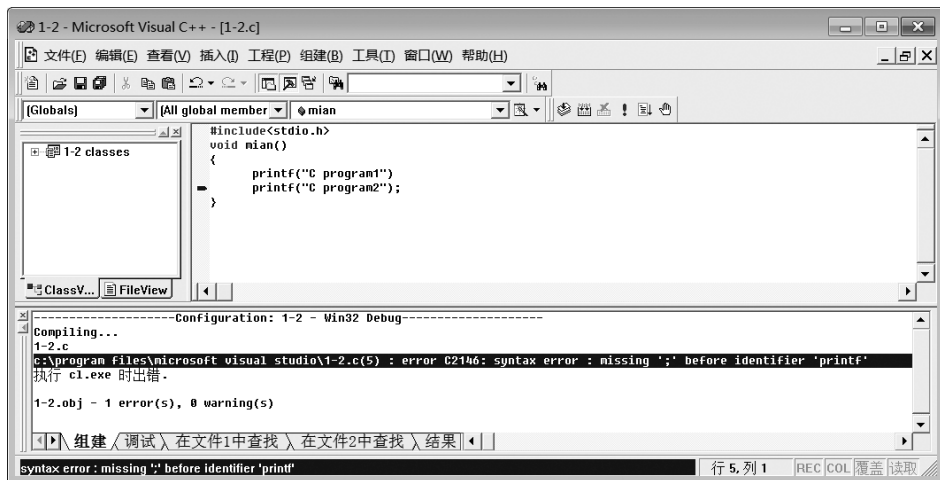


图 1-7 编译器给出的错误指示

这时可以看到, 错误提示已变成高亮显示状态。观察编译区域, 代码行的左侧出现了一个箭头。这个箭头指示了错误所在的大致位置, 通常情况下, 指示的代码出错位置是准确的。但在特殊情况下, 错误会出现在指示位置的附近。编译器基本能够精确地指出错误及出错位置。

经分析, 箭头指示行的上一行的末尾少了分号, 请添加分号以更正错误。

初学者刚开始编写程序时, 有很多错误是由简单的拼写错误造成的。例如, 忘了写逗号、右括号, 或者按错了键等。

[注意] 如果有多个错误的话, 请从第一个错误开始调试, 因为后续错误有可能因为第一个错误而引起。调试错误时, 请先阅读错误提示, 尽管错误提示是英文信息, 要努力去阅读和分析, 可借助各种英文翻译软件来理解它们的含义。遇到各种错误后, 识别错误、更正错误才是学习编程的唯一途径。

步骤五: 继续更正其他错误。

更正一个错误以后, 重新编译程序, 你会发现错误数量在减少, 请继续更正程序中的其他错误。

3. 调试程序

新建源文件 1-3.c 并输入如下程序。



教学视频

```
#include <stdio.h>
int max(int x,int y)    //传入两个整数，返回其中较大的那个整数
{
    int z;
    if(x>y)z=x;
    else z=y;
    return z;
}
void main()
{
    int a,b,c;
    printf("输入两个整数，例如4和5\n");
    scanf("%d %d",&a,&b);    //输入变量a和b的值
    c=max(a,b);              //调用max函数，传入变量a和b，然后将函数的返回值赋给c
    printf("max=%d",c);      //将其中较大的那个整数显示到屏幕上
}
```

初学者往往一看到自己编写的程序出现错误就不知所措了。有些读者认为，程序只要能够顺利运行就大功告成，他们未曾想过程中是否还存在某些隐患。要想不犯或少犯错误，就需要了解C语言程序设计中的错误类型和纠正方法。C语言程序设计中的错误类型可分为语法错误、链接错误、逻辑错误和运行错误。

(1) 语法错误：在编写程序时违反了C语言语法规则的规定。语法不正确、关键字拼错、标点符号漏写、数据与运算类型不匹配、括号不配对等都属于语法错误。对于语法错误，编译系统会在编译时指示出错位置并给出相应的提示。我们可以双击错误提示，将光标快速定位到代码中的出错位置。根据错误提示修改源程序，排除错误。

(2) 链接错误：如果使用了错误的函数调用，比如书写了错误的或不存在的函数名，编译系统在对它们进行链接时就会发现这种错误。纠正链接错误的方法与纠正语法错误的方法相同。

(3) 逻辑错误：虽然程序不存在上述两种错误，但程序的运行结果与预期不符。逻辑错误往往是因为程序采用的算法有问题，或是因为编写的程序逻辑与算法不完全吻合。逻辑错误比语法错误更难排除，需要程序员对程序分步进行调试，并检测循环、分支调用是否正确，另外需要跟踪变量的值能否按预期产生变化。

(4) 运行错误：程序不存在以上三种错误，但运行结果时出错。发生运行错误往往是因为程序的容错性不高，程序在设计时可能仅考虑了一部分数据的情况，因而对于其他数据就不适用了。例如，如果在打开文件时没有提前检测是否打开成功就开始对文件进行读写，那么在运行程序时，如果文件能够顺利打开，程序运行正确，反之，程序运行出错。为了避免发生这种类型的错误，需要对程序反复进行测试，完善算法，从而使程序能够适应各种情况的数据。

为了方便、快捷地排除程序中的逻辑错误，编译器提供了强大的调试功能。调试步骤如下。

步骤一：设置断点。

在设置断点之前，需要对程序进行编译。接下来，单击需要设置断点的代码行，然后单击工具栏或按F9功能键，光标所在代码行的最左边将出现一个深红色的实心圆点，这表示断点设置成功，如图1-8所示。

如果光标所在的代码行已经设置了断点，那么再次单击工具栏或按F9功能键，将会清除设置的断点。

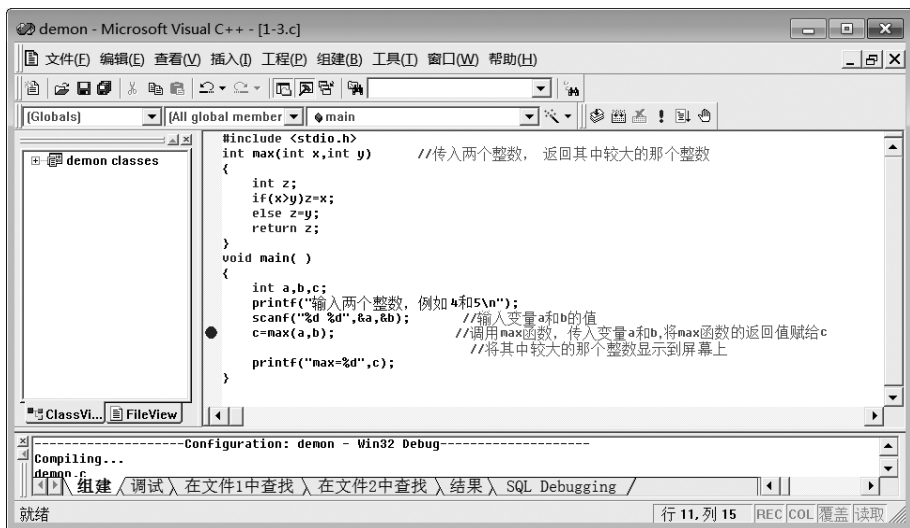


图 1-8 为程序设置断点

步骤二：调试程序。

首先编译程序。如果不存在编译错误，选择“组建”菜单中的“开始调试”→Go命令即可运行程序，直到程序结束。如果预先设置了断点，将在到达断点位置后，进入调试状态。

注意，由于程序中包含一条scanf语句，因此需要输入数据(例如输入4_5↵，其中的_代表空格，↵代表回车)，程序才能运行到断点位置。

进入调试状态以后，每按一次F11功能键，编译器就会将程序向后执行一行。因此，我们可以使用F11功能键控制程序逐行执行，直到程序运行结束。

进入调试状态以后，“组建”菜单会自动变为“调试”菜单。如果在调试过程中已经确定出错位置并且需要结束调试，可选择“调试”菜单中的Stop Debugging命令停止调试并返回到正常的编辑状态。

[说明] 如果希望一句一句地单步跟踪程序的执行过程，那么在编写程序时必须一行只写一条语句。

步骤三：查看变量。

设置断点的目的是使程序在执行到断点位置时停下来，此时可以检查各个变量的值，从而便于快速排错。

在单步调试程序的过程中，我们可以在Visual C++ 6.0工作界面的“变量”和“监视”子窗口中动态地查看变量的值。另外，将光标移到需要查看的变量上，在光标的右下角位置也可以看到对应变量的值。对于本例来说，在程序运行过程中，a被赋值为4，b被赋值为5，如图1-9所示。

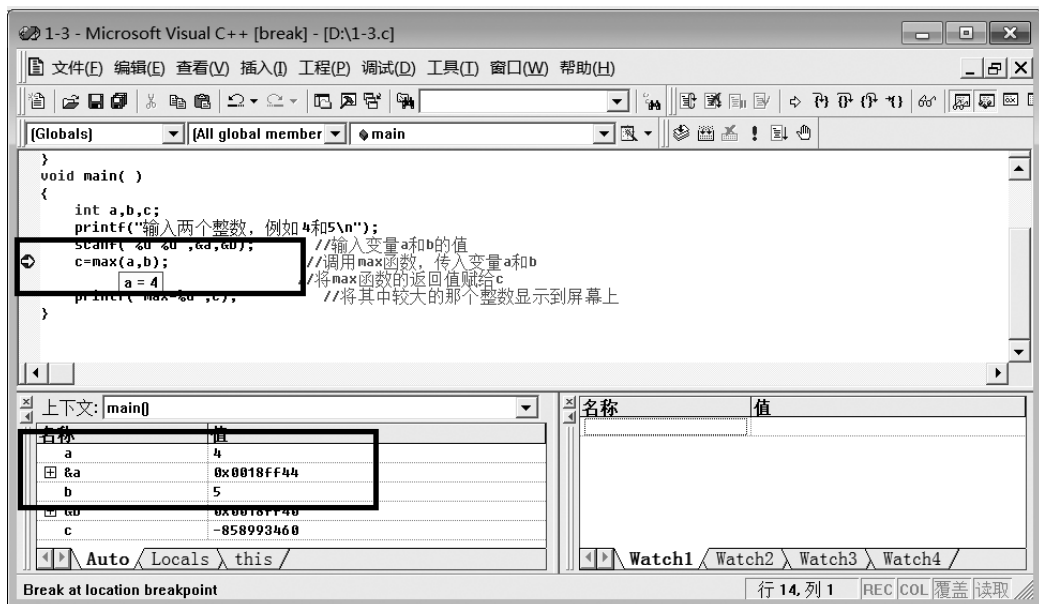


图 1-9 查看变量的值

4. 常见错误提示及可能的出错原因

1) syntax error : missing ';' before (identifier) 'xxxx'

直译：(标识符)xxxx前缺少分号。

错误分析：

这是最常见的编译错误。当出现这种错误提示时，往往指示的语句没有错误，而是上一条语句发生了错误，上一条语句的末尾缺少分号会导致编译器报这种错误。

2) unknown '\ooo' in program

直译：代码中含有无法识别的字符，这个字符的ASCII码值为(ooo)₈。

错误分析：

发生这种错误一般是因为在双引号之外使用了中文标点符号，如逗号、分号、括号、单引号和双引号等。可定位到对应的代码行，检查是否存在中文标点和全角空格。

3) 'xxxx' undeclared identifier

直译：标识符xxxx未定义。

错误分析：

如果xxxx是一个变量名，那么通常是由于程序员忘记了定义这个变量，也可能是由于拼写有误或大小写格式不一致而引起的，所以我们应该首先检查变量名是否正确。

如果xxxx是一个函数名,那就检查是否忘记了定义这个函数。既可能是因为拼写错误或大小写格式不一致,也可能是因为调用的函数根本不存在。当然,还有一种可能,就是定义的函数位于函数调用语句之后,而你没有在调用函数之前对函数进行声明。

如果xxxx是库函数的函数名,如sqrt、fabs,那么应该检查源文件的头部是否包含这些库函数所在的头文件(.h文件)。

标识符遵循“先定义,后使用”的原则。因此,无论是变量名、函数名还是类名,都必须先定义,后使用。使用在前,定义在后,也会导致这种错误。

前面的一条语句有错也可能导致编译器误认为当前语句也有错。如果前面的变量定义语句有错,那么编译器在后面的编译过程中会认为这个变量从来没有定义过,导致后面所有使用这个变量的语句都报错。如果函数的定义或声明语句有错,将引发同样的问题。

4) redeclaration of 'xxxx' with no linkage 或previous definition of 'xxxx' was here

直译:xxxx被重复定义。

错误分析:变量xxxx在同一作用域内被定义了多次。检查变量xxxx的每一次定义,只保留一个定义或者更改变量名。

5) empty character constant

直译:空的字符常量。

错误分析:连用了两个单引号,而中间没有任何字符,这是不允许的。

6) newline in constant

直译:常量中出现了换行。

错误分析:

- 检查字符串常量和字符常量中是否有换行符。
- 检查某个字符串常量的尾部是否漏掉了双引号。
- 检查某个字符常量的尾部是否漏掉了单引号。
- 检查是否在某条语句的尾部或中间错误输入了单引号、双引号或花括号。

7) too many characters in constant

直译:常量中的字符太多了。

错误分析:单引号中包含的内容表示字符型常量。通常,单引号中必须有且只能有一个字符(使用转义字符时,转义字符所代表的字符可当作一个字符看待)。如果单引号中的字符数量多于4个,就会引发这种错误。

另外,如果语句中的某个字符常量缺少右边的单引号,也会引发这种错误,例如:

```
if (x == 'x || x == 'y') { ... }
```

需要注意的是,如果单引号中的字符数量是2~4个,那么编译时不会报错。

8) unknown character '0x##'

直译:未知字符'0x##'。

错误分析:0x##是字符的ASCII码的十六进制表示形式。这里所说的未知字符,通常是指全角符号、字母和数字,也可能是指直接输入的汉字。全角字符和汉字如果使用双

引号包含起来，它们就将成为字符串常量的一部分，因而不会引发这种错误。

5. 编写程序时常用的快捷键

在编写程序时，经常会遇到需要复制、粘贴、删除代码的情况。使用快捷键可以提高代码的编写效率，表 1-1 列出了常用的快捷键。

表 1-1 编写程序时常用的快捷键

使用场合	快捷键	功能描述
Windows 系统通用 快捷键 (需要尽快记住)	Ctrl+Z	撤销一次操作，可撤销多次
	Ctrl+A	全选
	Ctrl+C	复制
	Ctrl+X	剪切
	Ctrl+V	粘贴
	Insert	覆盖模式/插入模式
	Ctrl+Alt+Delete	启动任务管理器，关闭一些无法关闭的进程
	Ctrl+Space	中文输入法的开/关键
	Ctrl+Shift	输入法切换键
Visual C++ 6.0 定义 的快捷键 (不用记忆)	Shift	在中文输入状态下，切换中/英文输入状态(特别方便)
	Ctrl+F5	运行当前程序
	F5	调试当前程序
	F9	设置/清除断点
	F11	单步执行一条语句
	Shift+F5	停止调试

[注意] 每一种编译器都会定义一些特有的快捷键，不需要记忆这些快捷键。打开菜单栏中对应的菜单命令，如果定义了快捷键的话，它们通常会标记在对应的菜单命令的后面。当使用不同的编译器时，请注意观察。

实验 2 数据类型、运算符及表达式

一、实验目的

- 练习不同进制下数字之间的转换。
- 掌握常量与变量的定义和赋值。
- 掌握对包含多种运算符的表达式进行混合运算的规律。
- 掌握数据类型之间的转换规律。

二、实验内容

1. 基础练习

【2-1-1】请将十进制数59分别转换为二进制、八进制和十六进制表示形式，写出预期结果，并上机检验计算结果，重点理解常用数制与整型常量的表示方法。检验程序如下：



教学视频

```
#include <stdio.h>
#include <stdlib.h> // itoa函数在stdlib.h文件中定义
void main()
{
    int a=59;
    char str[30];

    itoa(a,str,2); // itoa函数可以将a转换为二进制
    printf("59转换成二进制数是: %s\n",str);
    printf("59转换成八进制数是: %o\n",a);
    printf("59转换成十六进制数是: %X\n",a);
}
```

59转换成二进制数是_____

59转换成八进制数是_____

59转换成十六进制数是_____

【2-1-2】阅读以下程序，写出预期结果，分析原因并上机检验。重点理解整型常量的表示方法。



教学视频

```
#include <stdio.h>
void main()
{
    printf("我有%d个QQ好友\n",12);
    printf("我有%d个QQ好友\n",012);
    printf("我有%d个QQ好友\n",0x12);
}
```

程序运行结果：_____

原因分析：_____

【2-1-3】阅读以下程序，写出预期结果，分析原因并上机检验。重点理解变量的取值范围。

```
#include <stdio.h>
void main()
{
    short a;
```

```

char c;
a=32767; c=127;
printf("a=%d,c=%d\n",a,c);
a++; c++;
printf("a=%d,c=%d\n",a,c);
}

```

程序运行结果: _____

原因分析: _____

【2-1-4】分析下列程序, 写出当输入大写字母CK↵时的运行结果并上机检验。

[说明] ↵代表回车。

```

#include <stdio.h>
void main()
{
    char c1, c2;
    scanf ("%c%c", &c1, &c2);
    c1=c1+3;
    c2=c2-3;
    printf ("c1=%c,c2=%c",c1,c2);
}

```

程序运行结果: _____

【2-1-5】请计算下列表达式的值, 填入后面的括号中, 并上机验证。

假设a=3、b=5、c=-1、d=7。

$a/2 + (\text{int})(5/2.0 + 100/\text{062}' + a*2) / 2.0$ ()

$a-b \geq c \&\& b-a \geq d$ ()

$a||b+c \&\& b-c$ ()

$!(a < b) \&\& !c || 1$ ()

$(a=b+c, b > c-1) \&\& (b+=c) || (b*2==a+c)$ ()

验证程序如下:

```

#include <stdio.h>
void main()
{
    int a=3,b=5,c=-1,d=7;
    printf("%f\n",a/2 + (int)(5/2.0 + 100/\text{062}' + a*2) / 2.0 );
    printf("%d\n",a-b>=c&&b-a>=d );
    printf("%d\n",!(a<b)&&!c||1 );
    printf("%d\n",(a=b+c,b>c-1)&&(b+=c)|| (b*2==a+c) ); //注意==和=的区别
    printf("%d\n",a||b+c&&b-c );
    printf("%d\n",c+1&&b+c/2 );
}

```

【2-1-6】阅读下列程序，掌握自增及自减运算符的用法，分析程序运行结果，将结果写在括号中，并上机检验。

```
#include<stdio.h>
void main()
{
    //前缀++
    int n1 = 012;
    int n2 = ++n1 + 2;
    printf("n1=%d,n2=%d\n",n1,n2);    //(    )
    //后缀--
    int n3 = 012;
    int n4 = n3-- + 3;
    printf("n3=%d,n4=%d\n",n3,n4);    //(    )
    //前缀和后缀的练习
    int n5 = 10;
    int n6 = n5++ + 3;
    printf("n5=%d,n6=%d\n",n5,n6);    //(    )
    int n7 = 10;
    int n8 = --n7 - 10;
    printf("n7=%d,n8=%d\n",n7,n8);    //(    )
}
```

【2-1-7】阅读下列程序，分析数据类型之间的转换过程，写出程序运行结果，并上机检验。

```
#include<stdio.h>
void main()
{
    float f,x=3.9,y=5.7;
    int i=2,a,b,c;
    a=x+y;
    b=(int)x+y;
    c=(int)x+(int)y;
    f=11.0/i;
    printf("a=%d,b=%d,c=%d,f=%f\n",a,b,c,f);
}
```

程序运行结果：_____

【2-1-8】阅读下列程序，掌握复合赋值运算符的用法，写出程序运行结果，并上机检验。

```
#include<stdio.h>
void main()
{
    int x,y,z;
```

```

x=1;
y=2;
z=3;
x+=y;
y%=x;
z/=x;
printf("x=%d\n",x+=y);
printf("y=%d\n",y);
printf("z=%d\n",z);
}

```

程序运行结果：_____

【2-1-9】阅读下列程序，掌握符号常量的用法，写出程序运行结果，并上机检验。

```

#include <stdio.h>
#define LENGTH 10
#define WIDTH 5
#define NEWLINE '\n'
void main()
{
    int area;
    area = LENGTH * WIDTH;
    printf("value of area : %d", area);
    printf("%c", NEWLINE);
}

```

程序运行结果：_____

2. 程序填空

【2-2-1】下列程序实现的功能如下：接收通过键盘输入的一个字母，如果输入的是大写字母，就转换为小写字母并输出；如果输入的是小写字母，就转换为大写字母并输出；如果输入了其他字符，就输出一条提示信息。请在空白处填入适当的语句。

```

#include<stdio.h>
void main()
{
    char c;
    printf("请输入一个字母： ");
    scanf("%c",&c);
    if(c>='A'&&c<='Z')
        printf("%c\n", ① );
    else if(c>='a'&&c<='z')
        printf("%c\n", ② );
    else printf("输入错误，输入的字符无大小写之分");
}

```

3. 程序改错

【2-3-1】下列程序中有5处语法错误，请改正这些错误，使程序运行正确。重点掌握符号常量的定义方法。



教学视频

```
#include<stdio.h>
#define PRICE 100;
#define DISCOUNT 0.8;
voidmian
{
    int total;
    total=10*PRICE*DISCOUNT;
    printf("total=%d\n",total);
    total=100*PRICE*DISCOUNT;
    printf("total=%d\n",total);
}
```

【2-3-2】下列程序中有3处语法错误，请改正这些错误，使程序能正确运行。

```
#include <stdio.h>
void main()
{
    int a = 10;
    c = a+ +;
    printf("%d\n", c );
    c =(a+c)--;
    printf("%d\n", c );
}
```

4. 程序设计

【2-4-1】编写程序，实现如下功能：输入华氏温度，输出对应的摄氏温度，计算结果保留两位小数。

计算公式为： $C = \frac{5}{9}(F - 32)$ (F 代表华氏温度， C 代表摄氏温度)

算法提示：

(1) 注意，数据类型的自动转换可能会引起计算结果不正确。例如，按照C语言中的除法运算规则，公式中的5/9可能导致无论输入的数据是多少，计算结果都为0。

(2) 在数学公式中，我们通常会省略乘号，但在书写C语言表达式时，不可以省略任何计算符号。

【2-4-2】已知三角形的三条边的长度分别是3、5、7厘米，使用海伦公式求这个三角形的面积。

海伦公式为： $S = \sqrt{p(p-a)(p-b)(p-c)}$ 。

其中： $p = \frac{1}{2}(a+b+c)$ ， a 、 b 、 c 分别代表三角形各条边的长度。

算法提示：

- (1) 开平方函数为sqrt，定义在头文件math.h中。
- (2) 存放 p 值与 S 值的变量应定义为实型。
- (3) 在计算 $p = (a+b+c)/2$ 时，应设法把计算结果转换为实型之后再赋值，否则计算结果有可能不正确。

【2-4-3】编写程序，输出心形图案。

算法提示：可以任选一种字符(如*、@或%)，将字符在屏幕上按图形规律显示。

【2-4-4】编写程序，给变量 a 赋值0.0，给变量 b 赋值1.2，给变量 c 也赋值1.2，判断变量 a 是否等于0以及变量 b 和变量 c 是否相等。

算法提示：

(1) 实数一般不判断是否“相等”，而是判断是否接近。判断变量 a 是否为0的方法，通常是判断变量 a 的绝对值是否小于 10^{-6} 。在判断变量 b 和变量 c 是否相等时，需要对它们执行减法运算，再取绝对值，然后判断这个绝对值是否小于 10^{-6} 。

(2) 如果参与比较的两个数都比 10^{-6} 小，则不能再使用 10^{-6} 作为判断标准，而应重新定义误差范围。

(3) 实数的求绝对值函数为fabs，定义在头文件math.h中。

(4) 10^{-6} 可以使用指数形式写为1e-6。

实验3 常用的输入输出函数

一、实验目的

- 掌握字符输入输出函数的用法。
- 掌握格式输入输出函数的用法。
- 在使用printf函数时，掌握输出格式控制字符串的使用方法。
- 在使用scanf函数时，能够理解输入格式需要与格式控制字符串中的要求保持一致。

二、实验内容

1. 基础练习

【3-1-1】完成如下程序的输入，运行程序，理解getchar函数和putchar函数的用法。

```
#include <stdio.h>
void main()
{
```

```

int c;
printf( "Enter a value :");
c = getchar( );
printf( "\nYou entered: ");
putchar( c );
printf( "\n");
}

```

[注意] getchar函数不带参数。

【3-1-2】使用scanf函数输入数据，为了使a=1、b=2、x=3.5、y=4.6、c1='A'、c2='z'，应如何输入数据？并上机检验。



教学视频

```

#include <stdio.h>
void main()
{
    int a,b;
    float x,y;
    char c1,c2;
    scanf("a=%d,b=%d",&a, &b);
    scanf("%f,%f",&x,&y);
    scanf("%c%c",&c1, &c2);
    printf("a=%d,b=%d,x=%f,y=%f,c1=%c,c2=%c\n",a,b,x,y,c1,c2);
}

```

请写出输入格式：_____

【3-1-3】使用scanf函数输入数据，为了使a=10、b=20、c1='A'、c2='a'、x=1.1、y=2.2、z=3.3，应如何输入数据？并上机检验。

```

#include <stdio.h>
void main()
{
    int a,b;
    float x,y,z;
    char c1,c2;
    scanf ("%5d%5d%c%c%f%f%*f,%f",&a, &b, &c1, &c2, &x,&y,&z);
    printf("a=%d,b=%d,c1=%c,c2=%c,x=%6.2f,y=%6.2f,z=%6.2f\n",a,b,c1,c2,x,y,z);
}

```

请写出输入格式：_____

【3-1-4】假设圆的半径 $r=2.5$ ，圆柱高 $h=3$ ，求底面周长、底面积和圆柱体积。使用scanf函数输入数据，在输入数据之前要有提示信息，输出时要有文字说明，保留两位小数。请参考如下代码，完成上机练习。

```

#include <stdio.h>
void main ()

```

```

{
    float h,r,l,s,v;
    float pi=3.141593;
    printf("请输入圆的半径r和圆柱高h: ");
    scanf ("%f,%f",&r,&h);           //输入圆的半径r和圆柱高h
    l=2*pi*r;                          //计算底面周长l
    s=pi*r*r;                          //计算底面积s
    v=pi*r*r*h;                        //计算圆柱体积v
    printf("底面周长为: l=%6.2f\n",l);
    printf("底面积为: s=%6.2f\n",s);
    printf("圆柱体积为: v=%6.2f\n",v);
}

```

【3-1-5】测试float型数据的有效位数，程序如下，请上机验证。

```

#include <stdio.h>
int main()
{
    float a;
    a= 10000/3.0;
    printf(" %f\n",a);
}

```

运行结果：
3333.333252

计算的理论值应为3333.33333333…，但由于float型数据只能保证6位或7位有效数字，因此虽然程序输出了6位小数，但从左边开始的第7位及之后的数字无法保证绝对正确。

请问，如果变量a的数据类型为double，其他不变，输出结果如何？可上机一试。

请写出测试结果：_____

【3-1-6】阅读下列程序，重点理解非格式控制字符的用法，写出运行结果，并上机检验。

```

#include <stdio.h>
int main()
{
    int a=88,b=89;
    printf("%d %d\n",a,b);
    printf("%d,%d\n",a,b);
    printf("%c,%c\n",a,b);
    printf("a=%d,b=%d",a,b);
}

```

程序运行结果：_____

【3-1-7】阅读下列程序，写出运行结果，并上机检验。

```

#include <stdio.h>

```



教学视频

```

int main()
{
    int a=15;
    float b=123.456;
    double c=12345678.1234567;
    char d='p';

    printf("a=%d\n", a);
    printf("a=%d,a=%5d,a=%o,a=%x\n\n",a,a,a,a);

    printf("a=%f\n", b);
    printf("b=%f,b=%lf,b=%5.4lf,b=%e\n\n",b,b,b,b);

    printf("c=%f\n", c);
    printf("c=%lf,c=%f,c=%8.4lf\n\n",c,c,c);

    printf("d=%c\n", d);
    printf("d=%c,d=%8c\n",d,d);
}

```

程序运行结果: _____

2. 程序填空

【3-2-1】按照注释中的要求填入适当的格式控制字符，并上机检验。

```

#include <stdio.h>
int main()
{
    int a=1;
    float b=3.1415926;
    char c='A';

    printf("a= ① \n", a); //左对齐，占6个字符宽度
    printf("a= ② \n", a); //右对齐，占6个字符宽度

    printf("b= ③ \n", b); //占4个字符宽度 保留2位小数
    printf("b= ④ \n", b); //占8个字符宽度 保留4位小数

    printf("c= ⑤ \n", c); //按字符输出
    printf("c= ⑥ \n", c); //按十进制整型输出
    printf("c= ⑦ \n", c); //按八进制整型输出
    printf("c= ⑧ \n", c); //按十六进制整型输出
}

```



教学视频

【3-2-2】按照注释中的要求填入适当的格式控制字符，并上机检验。

```

#include <stdio.h>
int main()

```

```

{
    char a,b;
    int n1,n2,n3;

    scanf(" ① ", ② ); //输入两个字符，分别赋给变量a和b
    scanf(" ③ ", ④ ); //连续输入9个数字，每3个数字一组，作为整体分别赋给n1、n2、n3
    scanf(" ⑤ ", ⑥ ); //输入格式为n1=5、n2=6
}

```

3. 程序改错

【3-3-1】下列程序没有语法错误，但是得不到预期结果。例如，程序运行后，输入3.4，输出结果为1073741824，请找出原因后改正，并上机检验。

```

#include <stdio.h>
void main()
{
    float a;
    printf("请输入一个实型数据: ");
    scanf("%f",&a);
    printf("%d",a);
}

```

【3-3-2】下列程序的功能为：输入圆的半径后，计算圆的面积，输出结果保留两位小数，程序中有几处错误，请尝试改正，并上机检验。

```

#include <stdio.h>
voidmain()
    float r,zc,area,pi=3.14;
    printf("input r: \n");
    scanf("%f\n",r);
    zc=2*pi*r;
    area=pi*r*r;
    printf("r=%f,zc=%f,area=%f\n",r,zc,area);
}

```

4. 程序设计

【3-4-1】编写程序，使用getchar函数读入两个字符，分别赋给变量c1和c2，然后分别使用putchar和printf函数输出这两个字符。思考以下问题。

- (1) 变量c1、c2应定义为字符型还是整型？抑或二者皆可？
- (2) 要求将c1、c2变量按ASCII码输出，应使用putchar函数还是printf函数？
- (3) 整型变量与字符变量是否在任何情况下都可以互相替代？

【3-4-2】编写程序，练习scanf和printf函数的格式控制字符的使用方法。

- (1) 输入两个字符，分别赋给变量c1，c2，将它们按十进制整型输出。

(2) 输入两个整型数值，以逗号作为分隔符，分别赋给变量a1和a2，将它们按域宽为5输出。

(3) 输入两个实型数值，以冒号作为分隔符，分别赋给变量f1和f2，将它们按域宽为5输出并保留两位小数。

(4) 在使用每一条scanf语句之前，请在屏幕上显示合理的提示信息。

实验4 选择结构

一、实验目的

- 掌握使用关系表达式和逻辑表达式表示判断条件的方法。
- 掌握逻辑值的表示方法：非零值代表真，0代表假。
- 熟练掌握if语句的用法。
- 熟练掌握switch语句的用法。

二、实验内容

1. 基础练习

【4-1-1】以下程序可以判断输入的数字是奇数还是偶数，请阅读后进行上机验证。请尝试使用条件运算符?:改写程序。

```
#include <stdio.h>
void main()
{
    int num;
    printf("输入一个整数: ");
    scanf("%d",&num);
    if (num%2==0)
        printf("偶数\n");
    else
        printf("奇数\n");
}
```

【4-1-2】以下程序实现的功能为：输出三个数中的最大数。请阅读后进行上机验证。尝试使用if语句改写程序。

```
#include <stdio.h>
void main()
{
    int x,y,z,max;
    printf("请输入三个整数(使用空格作为分隔符): ");
```