

普.通.高.等.学.校
计算机教育“十二五”规划教材

C 语言程序设计教程 实验指导与习题解答

(第4版)

THE ANSWER AND PRACTICE OF THE C
PROGRAMMING LANGUAGE
(4th edition)

李丽娟 ◆ 主编

吴蓉晖 杨小林 谷长龙 洪跃山 李根强 杜四春 ◆ 副主编



普.通.高.等.学.校
计算机教育“十二五”规划教材

C 语言程序设计教程 实验指导与习题解答

(第 4 版)

*THE ANSWER AND PRACTICE OF THE C
PROGRAMMING LANGUAGE
(4th edition)*

李丽娟 ◆ 主编

吴蓉晖 杨小林 谷长龙 洪跃山 李根强 杜四春 ◆ 副主编

人 民 邮 电 出 版 社
北 京

图书在版编目 (C I P) 数据

C语言程序设计教程实验指导与习题解答 / 李丽娟主
编. — 4版. — 北京: 人民邮电出版社, 2013.1 (2013.3 重印)
普通高等学校计算机教育“十二五”规划教材
ISBN 978-7-115-29730-3

I. ①C… II. ①李… III. ①
C语言—程序设计—高等学校—教学参考资料 IV.
①TP312

中国版本图书馆CIP数据核字(2012)第259578号

内 容 提 要

本书是和《C语言程序设计教程(第4版)》配套使用的实验指导与习题解答教程。

全书主要包括与《C语言程序设计教程(第4版)》对应的实验指导、习题解答和常用编译环境介绍3部分。第一部分为实验指导,针对《C语言程序设计方法(第4版)》每一章的知识点,安排不同的上机实验内容,并给出所有上机实验的参考解答。第二部分为习题解答,包括《C语言程序设计教程(第4版)》各章的习题和参考解答,并对部分概念性较强的习题给出注解。第三部分为附录,简要介绍了Visual C++ 6.0、Turbo C、Borland C++ 5.0集成环境下程序的编辑、编译和运行的方法。

全书内容紧扣C语言程序设计的相关知识点,案例丰富,实用性强,可作为学习C语言程序设计的配套教材。

普通高等学校计算机教育“十二五”规划教材

C语言程序设计教程实验指导与习题解答(第4版)

- ◆ 主 编 李丽娟
副 主 编 吴蓉晖 杨小林 谷长龙 洪跃山 李根强 杜四春
责任编辑 邹文波
- ◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街14号
邮编 100061 电子邮件 315@ptpress.com.cn
网址 <http://www.ptpress.com.cn>
中国铁道出版社印刷厂印刷
- ◆ 开本: 787×1092 1/16
印张: 16
字数: 418千字
- 2013年1月第4版
2013年3月北京第3次印刷

ISBN 978-7-115-29730-3

定价: 29.80元

读者服务热线: (010)67170985 印装质量热线: (010)67129223
反盗版热线: (010)67171154

目 录

第一部分 实 验

第 1 章 实验引言.....2

- 1.1 目的和要求.....2
- 1.2 相关知识.....2
- 1.3 实验内容.....2
 - 1.3.1 熟悉 Turbo C 环境.....2
 - 1.3.2 熟悉 Visual C++ 环境.....4
 - 1.3.3 熟悉 Dev C++ 环境.....5
 - 1.3.4 熟悉 Borland C++ 环境.....6
- 1.4 C 语言开发环境综合练习.....8

第 2 章 基本的程序语句实验.....9

- 2.1 目的和要求.....9
- 2.2 相关知识.....9
 - 2.2.1 常量和变量.....9
 - 2.2.2 算术表达式.....9
 - 2.2.3 运算符.....10
 - 2.2.4 编程中应注意的事项.....10
 - 2.2.5 典型案例.....10
- 2.3 实验内容.....12
 - 2.3.1 表达式的计算.....12
 - 2.3.2 表达式的应用.....12
 - 2.3.3 字符与 ASCII 码.....13
 - 2.3.4 运算符与表达式的应用.....14
 - 2.3.5 顺序结构应用程序.....15
 - 2.3.6 分析程序.....16
- 2.4 程序案例.....17

第 3 章 程序的简单算法设计实验.....20

- 3.1 目的和要求.....20
- 3.2 相关知识.....20
- 3.3 实验内容.....20
 - 3.3.1 数学函数的算法描述.....20

- 3.3.2 鸡兔同笼的算法描述.....21
- 3.3.3 确定坐标的算法描述.....22

第 4 章 分支结构实验.....24

- 4.1 目的和要求.....24
- 4.2 相关知识.....24
 - 4.2.1 if 语句.....24
 - 4.2.2 switch 语句.....24
 - 4.2.3 编程中应注意的事项.....25
 - 4.2.4 典型案例.....25
- 4.3 实验内容.....28
 - 4.3.1 if 语句的应用.....28
 - 4.3.2 switch/case 语句的应用.....28
 - 4.3.3 switch/case 语句嵌套 if 语句的应用.....29
 - 4.3.4 switch/case 结构的嵌套应用.....30
 - 4.3.5 分析程序.....32
- 4.4 程序案例.....32

第 5 章 循环结构实验.....35

- 5.1 目的和要求.....35
- 5.2 相关知识点.....35
 - 5.2.1 while 语句.....35
 - 5.2.2 do_while 语句.....36
 - 5.2.3 for 语句.....36
 - 5.2.4 典型案例.....37
- 5.3 实验内容.....39
 - 5.3.1 用 for 语句实现循环.....39
 - 5.3.2 用 while 循环语句实现循环.....41
 - 5.3.3 用 do_while 语句实现循环.....43
 - 5.3.4 用 while 语句和 for 语句配合实现
循环.....43
 - 5.3.5 用 for 语句嵌套实现循环.....45
- 5.4 程序案例.....49

第 6 章 函数和宏定义实验.....52

- 6.1 目的和要求52
- 6.2 相关知识点52
- 6.3 典型案例53
 - 6.3.1 模块功能设计 153
 - 6.3.2 模块功能设计 254
- 6.4 实验内容54
 - 6.4.1 函数的定义和调用54
 - 6.4.2 模块化程序设计59
 - 6.4.3 函数的递归调用65
- 6.5 程序案例67

第 7 章 数组实验.....71

- 7.1 目的和要求71
- 7.2 相关知识71
 - 7.2.1 一维数组71
 - 7.2.2 多维数组71
 - 7.2.3 字符数组与字符串72
- 7.3 实验内容72
 - 7.3.1 一维数组的应用72
 - 7.3.2 二维数组的应用75
 - 7.3.3 字符数组应用77
- 7.4 程序案例78

第 8 章 指针实验.....87

- 8.1 目的和要求87
- 8.2 相关知识点87
- 8.3 实验内容89
 - 8.3.1 指针基础及指针运算89
 - 8.3.2 数据交换90
 - 8.3.3 字符串反转及字符串连接91
 - 8.3.4 数组元素奇偶排列93
- 8.4 综合练习94

第 9 章 构造数据类型实验.....96

- 9.1 目的和要求96
- 9.2 相关知识96
 - 9.2.1 结构型变量的定义和引用96
 - 9.2.2 共用型97

- 9.2.3 枚举型97

9.3 实验内容.....98

- 9.3.1 结构体变量的应用98
- 9.3.2 结构体数组的应用100
- 9.3.3 共用体的应用101
- 9.3.4 结构体指针的应用103

9.4 程序案例.....105**第 10 章 文件操作实验**.....117

- 10.1 目的和要求117
- 10.2 相关知识点117
 - 10.2.1 文件概述117
 - 10.2.2 文件型指针118
 - 10.2.3 操作文件的常用函数118
 - 10.2.4 典型案例118
- 10.3 实验内容121
 - 10.3.1 文本文件的读操作121
 - 10.3.2 文本文件的读写操作123
 - 10.3.3 二进制文件的读写操作125
 - 10.3.4 文件指针的定位操作127
 - 10.3.5 文件的加密操作128
- 10.4 程序案例130
 - 10.4.1 程序案例一130
 - 10.4.2 程序案例二133

第 11 章 位运算实验.....138

- 11.1 目的和要求138
- 11.2 相关知识点138
 - 11.2.1 位运算的概念138
 - 11.2.2 位运算及位运算赋值操作138
 - 11.2.3 位段138
 - 11.2.4 典型案例139
- 11.3 实验内容142
 - 11.3.1 按位与运算练习142
 - 11.3.2 按位异或运算练习143
 - 11.3.3 按位或运算练习144
 - 11.3.4 按位取反运算练习145
- 11.4 程序案例146
 - 11.4.1 程序案例一146
 - 11.4.2 程序案例二147

第二部分 习题解答

第 1 章 引言习题解答.....150	第 6 章 函数与宏定义习题解答.....179
第 2 章 基本的程序语句习题解答.....152	第 7 章 数组习题解答.....187
第 3 章 程序的简单算法设计习题 解答.....156	第 8 章 指针习题解答.....200
第 4 章 分支结构习题解答.....161	第 9 章 构造数据类型习题解答.....206
第 5 章 循环结构习题解答.....166	第 10 章 文件操作习题解答.....212
	第 11 章 位运算习题解答.....220

第三部分 附 录

附录 A Visual C++ 6.0 使用 简介.....226	附录 D Turbo C 2.0 常见编译 错误.....237
附录 B Visual C++ 6.0 常见编译 错误.....230	附 D.1 致命错误.....237
附录 C Turbo C 2.0 集成开发环境 简介.....232	附 D.2 一般错误.....237
	附 D.3 编译警告.....245
	附录 E Borland C++ 5.0 集成开发 环境简介.....247

第一部分

实验

- 第1章 实验引言
- 第2章 基本的程序语句实验
- 第3章 程序的简单算法设计实验
- 第4章 分支结构实验
- 第5章 循环结构实验
- 第6章 函数和宏定义实验
- 第7章 数组实验
- 第8章 指针实验
- 第9章 构造数据类型实验
- 第10章 文件操作实验
- 第11章 位运算实验

第 1 章

实验引言

1.1 目的和要求

- (1) 了解 C 语言程序的开发环境，掌握 C 语言源程序的编辑、编译、执行的过程。
- (2) 了解不同的 C 语言程序开发环境，并在不同的环境下编辑、编译、链接和运行 C 语言程序。
- (3) 通过运行简单的 C 程序，初步了解 C 语言源程序的特点。

1.2 相 关 知 识

- 1. C 语言程序要在特定的开发环境下才能生成可执行的程序，适合 C 语言的开发环境有：Turbo C 2.0、Borland C++、Visual C++ 6.0、gcc 等。
- 2. C 语言程序的基本结构主要由以下 6 部分组成：
 - (1) 文件包含部分；
 - (2) 预处理部分；
 - (3) 变量说明部分；
 - (4) 函数原型声明部分；
 - (5) 主函数部分；
 - (6) 自定义函数部分。若要生成可执行的程序，其中文件包含部分和主函数部分一般不可以缺少，其他部分可根据程序的需要而定。
- 3. C 语言源程序是不能直接运行的，必须要对源程序进行编译、链接后，才能生成可执行的程序。

1.3 实 验 内 容

1.3.1 熟悉 Turbo C 环境

Turbo C 是 Borland 公司推出的一种 C 语言程序集成开发环境，它将编辑、编译、链接和运行

集成在一起,使用起来方便简单,具有效率高、功能强等优点。

实验练习 安装 Turbo C 2.0,并在该环境下编辑一个如下所示的简单 C 语言程序:

```
#include <stdio.h>
main()
{
    printf("Hello,World!\n");
}
```

对其进行编译、生成可执行的程序,执行该程序,观察程序的运行结果。

1. 实验目的

本实验旨在巩固学生对编程基本概念的理解。主要目的如下:

- (1) 了解 Turbo C 环境下的 C 语言程序的开发过程;
- (2) 了解和学会 printf 函数的基本用法。

在强化练习中,练习如何输出指定的信息。

2. 问题描述

修改上面的程序,输出信息:

```
I am learning The C program.
Very good!
```

3. 示例输出

```
I am learning The C program.
Very good!
```

4. 程序样板

```
#include <stdio.h>
main()
{
    printf("____ ① ____");
    printf("____ ② ____");
}
```

5. 问题解答与提示

- (1) 启动 Turbo C 系统,进入 Turbo C 集成开发环境。
- (2) 在主菜单下,选择“File”菜单下的“New”命令(建立新文件),进入 Turbo C 编辑状态。
- (3) 在编辑窗口下,输入上述程序。

6. 强化练习

根据程序模板,现要求输出如下信息:

```
*****
                Very good
*****
```

请写出相应的源程序。

7. 实验结果小结

根据使用过程,回答下面的问题:

- (1) 怎样进入程序的编辑状态?
- (2) 使用什么命令对源程序进行编译?
- (3) 怎样生成可执行的文件?
- (4) 怎样查看程序的运行结果?
- (5) 给出 Turbo C 2.0 两个不同状态下的界面图,如编辑状态、编译状态、环境设置状态、文

件打开状态等。

1.3.2 熟悉 Visual C++ 环境

Visual C++ 6.0 (简称 VC 或者 VC6.0), 是微软推出的一款 C++ 编译器, 该编译器同样也适合于 C 语言, 它是一个基于 Windows 操作系统的可视化集成开发环境 (Integrated Development Environment, IDE), 相对于 Windows 操作系统的不同版本, Visual C++ 也有不同的版本。

实验练习 1 安装 Visual C++ 6.0, 并在该环境下重做 1.3.1 小节的练习。

实验练习 2 在 Visual C++ 环境下, 编辑一个如下所示的简单 C 语言程序:

```
#include<stdio.h>
main()
{
    int a,b,c;
    printf("please enter a,b:");
    scanf("%d,%d",&a,&b);
    c=a+b;
    printf("c=%d\n",c);
}
```

1. 实验目的

本实验旨在巩固学生对编程基本概念的理解。主要目的如下:

- (1) 了解算术运算的表达式;
- (2) 掌握 printf 函数的基本用法;
- (3) 了解 scanf 函数的基本用法。

在强化练习中, 练习整数的减法运算和乘法运算。

2. 问题描述

编写程序, 计算两整数的和与积。

3. 示例输出

```
please enter a,b: 3,5
3 + 5 = 8
3*5 = 15
```

4. 程序样板

下面的程序需要输出表达式的计算结果, 请在下面程序的空白处填入合适的内容, 将表达式的计算结果输出到屏幕, 运行程序, 观察程序的运行结果。

```
#include<stdio.h>
main()
{
    int a,b,c,d;
    printf("please enter a,b: ");
    scanf("%d,%d",&a,&b);
    c=a+b;
    d=a*b;
    printf("____ ① ____");
    printf("____ ② ____");
}
```

5. 问题解答提示

- (1) 程序功能是进行加法运算和乘法运算, 因此需要从键盘输入两个变量的值。

(2) 观察示例输出 $a+b$ 和 $a*b$ 的结果可知, 程序进行的是整型数据的计算, 因此变量类型均为整型。

(3) 用 `scanf` 语句接收数值, 并存储在变量 a 、 b 中。

(4) 分别将加法运算和乘法运算的结果赋给不同的变量保存起来。

(5) 输出计算结果。

6. 强化练习

如果要计算两整数的差, 上述程序又将如何修改呢? 请写出正确答案。

7. 实验结果小结

根据使用过程, 回答下面的问题:

(1) 怎样进入程序的编辑状态?

(2) 使用什么命令对源程序进行编译?

(3) 怎样生成可执行的文件?

(4) 怎样查看程序的运行结果?

(5) 给出 Visual C++ 两个不同状态的下的界面图, 如编辑状态、编译状态、环境设置状态、文件打开状态等。

1.3.3 熟悉 Dev C++ 环境

Dev C++ 是一个 Windows 环境下适合 C 语言和 C++ 的开发工具, 它是一款自由软件, Dev C++ 是一个非常实用的编程软件, 适合初学者与编程高手的不同需求, 是适合用于学习 C 语言或 C++ 的开发工具。

实验练习 1 安装 Dev C++ 5.0 (也可以是 Dev C++ 的其他版本), 在该环境下, 重做 1.3.1 小节的练习。

实验练习 2 在 Dev C++ 环境下编辑一个如下所示的简单 C 语言程序:

```
#include <stdio.h>
main()
{
    printf("\n");
    printf("***\n");
    printf("****\n");
    printf("*****\n");
    printf("*****\n");
}
```

对其进行编译、生成可执行的程序, 执行该程序, 观察程序的运行结果。

1. 实验目的

本实验旨在巩固学生对编程基本概念的理解。主要目的如下:

(1) 熟悉 Dev C++ 环境下的 C 语言程序的开发过程;

(2) 通过运行简单的 C 程序, 初步了解 C 语言源程序的特点;

(3) 改变 `printf` 中的符号, 输出其他自创图案。

在强化练习中, 练习用 C 语言的 `printf` 语句输出各种不同的图案。

2. 问题描述

编写程序, 输出一个由字母 A 组成的等腰三角形图案, 三角形的两边空白处用字母 O 填充。

3. 示例输出

```
OOOOOA00000
OOOOAAA0000
```

```

OOOAAAAOOO
OOAAAAAAOO
OAAAAAAAO
AAAAAA

```

4. 程序样板

请在下面程序的空白处填写你喜欢的字母符号, 能使程序输出不同的几何图案, 然后运行程序, 观察程序的运行结果, 了解 printf 语句的输出几何图案特点。

```

#include <stdio.h>
main()
{
    printf("_____(1)_____");
    printf("_____(2)_____");
    printf("_____(3)_____");
    printf("_____(4)_____");
    .....
    printf("_____(n)_____");
}

```

5. 问题解答与提示

(1) 根据问题描述进行分析, 程序的主要任务是输出有规律的几何图案。因此, 关键在于要给出图案的形状, 然后用 printf 语句逐行将图案输出。

(2) 通过这种固定方式来输出几何图案, 大小是固定不变的, 除非修改程序的源代码。

6. 强化练习

请在上面程序的基础上, 输出下列几种几何图案: 圆形、五角星、倒三角形。

7. 实验结果小结

根据使用过程, 回答下面的问题:

- (1) 怎样进入程序的编辑状态?
- (2) 使用什么命令对源程序进行编译?
- (3) 怎样生成可执行的文件?
- (4) 怎样查看程序的运行结果?
- (5) 给出 Dev C++ 两个不同状态的下的界面图, 如编辑状态、编译状态、环境设置状态、文件打开状态等。

1.3.4 熟悉 Borland C++ 环境

Borland C++ 也是一个 Windows 环境下适合 C 语言和 C++ 的开发工具, 它使用方便, 适合于学习 C 语言或 C++ 的开发工具。

实验练习 1 安装 Borland C++ 5.0 (也可以是 Borland C++ 的其他版本), 在该环境下, 重做 1.3.1 小节的练习。

实验练习 2 在 Borland C++ 环境下编辑一个如下所示的简单 C 语言程序:

```

#include <stdio.h>
main()
{
    printf("Data Types and Sizes :\n");
}

```

```

printf("char: %d\n", sizeof(char));
printf("float: %d\n", sizeof(float));
printf("double: %d\n", sizeof(double));
}

```

对其进行编译、生成可执行的程序，执行该程序，观察程序的运行结果。

1. 实验目的

本实验旨在巩固学生对编程基本概念的理解。主要目的如下：

- (1) 熟悉 Borland C++环境下的 C 语言程序的开发过程；
- (2) 通过运行简单的 C 程序，初步了解 C 语言源程序的特点；
- (3) 了解 sizeof 运算符的特殊用法。

在强化练习中，练习 C 语言的各种数据类型是否固定不变。

2. 问题描述

编写程序，第 1 行输出字符串“Data Types and Sizes:”，从第 2 行开始打印输出 Borland C++环境下 C 语言中各种数据类型存储空间的大小。

3. 示例输出

```

Data Types and Sizes:
long: 4
unsigned int: 2
double: 8

```

4. 程序样板

请在下面程序的空白处随意填写文字或者表达式，然后运行程序，观察程序的运行结果，了解 printf 语句的特点。

```

#include <stdio.h>
main()
{
    printf("____ ① ____");
    printf("____ ② ____");
    printf("____ ③ ____");
    printf("____ ④ ____");
}

```

5. 问题解答与提示

(1) 根据问题描述进行分析，程序分为两项任务，第 1 项要求输出指定的字符串，第 2 项要求输出 C 语言中各种数据类型存储空间大小。

(2) 对于第 1 项任务，与实验练习 1 内容相似，故可以采用 printf 语句进行编写。

(3) 对于第 2 项任务，首先分析 C 语言中有哪些主要的数据类型（int 型、char 型、float 型、double 型等），然后如何通过程序计算出这些类型所占用的存储空间（使用系统提供的求某种数据类型存储空间字节数的函数 sizeof()；最后将各种类型所占用的存储空间大小输出。

6. 强化练习

(1) 考虑上述类型所占用的存储空间大小是否是固定的？如果不是，你能说出原因吗？

(2) 如果在输出字符串下加一行“-----”，应怎么修改程序？请写出正确答案。

7. 实验结果小结

根据使用过程，回答下面的问题：

(1) 怎样进入程序的编辑状态？

- (2) 使用什么命令对源程序进行编译?
- (3) 怎样生成可执行的文件?
- (4) 怎样查看程序的运行结果?
- (5) 给出 Borland C++ 两个不同状态下的界面图, 如编辑状态、编译状态、环境设置状态、文件打开状态等。

1.4 C 语言开发环境综合练习

1. 练习安装几种不同的适合 C 语言的集成开发环境。要求了解和掌握以下几个方面的内容:
 - (1) 系统的目录结构;
 - (2) 系统每一个目录下的文件形式;
 - (3) 运行该集成编辑环境, 调入系统提供的程序范例, 练习对该程序的编辑、运行, 了解程序的运行机制。
2. 对一个已经安装好的 Turbo C 2.0 集成编辑环境系统, 要求了解和掌握以下几个方面的问题:
 - (1) 改变系统的根目录名;
 - (2) 运行该集成编辑环境, 调入系统提供的程序范例, 对该程序进行编译, 看看会发生什么问题, 寻找解决方案;
 - (3) 了解集成环境下的其他功能。

第 2 章

基本的程序语句实验

2.1 目的和要求

- (1) 掌握 C 语言几种基本数据类型，如整型 (int)、字符型 (char)、单精度型 (float)、双精度型 (double)，以及由这些基本类型构成的常量和变量的使用方法。
- (2) 掌握 C 语言的运算符和表达式的正确使用。
- (3) 掌握自增 (++) 和自减 (--) 运算符的正确使用。
- (4) 掌握基本的输入/输出函数 scanf()、getchar()、printf() 和 putchar() 函数的基本功能、调用方法、输入/输出格式控制规定等。
- (5) 能独立编写顺序结构程序并调试通过。
- (6) 用表达式语句、输入/输出语句解决简单的应用问题并上机实现。

2.2 相 关 知 识

2.2.1 常量和变量

常量和变量是 C 语言数据处理的基本对象。

1. 常量分为数值常量、字符常量和字符串常量

数值常量分为整型常量和实型常量两种，分别表示整数和实数。整数可以使用十进制数、八进制数和十六进制数表示；实数只使用十进制数，采用小数形式和指数形式具体表示。

字符常量是用单引号括起来的一个字符表示，字符常量的值是该字符所对应的 ASCII 值。

字符串常量是用双引号括起的一串字符表示，字符串在内存存储时，自动在其尾部加上一个空 (NULL) 字符。

2. 变量按照数据类型定义

基本数据类型有字符型、整型、单精度浮点类型和双精度浮点类型，由基本数据类型的数据长度不同及有无符号又可派生多种类型。

2.2.2 算术表达式

算术表达式是用算术运算符和括号将运算对象连接起来，且符合 C 语言语法规则的式子。

C 语言在不同数据类型混合运算时会按规则自动转换成相同的数据类型后再进行运算,即在 C 语言编译时自动把参与运算的各种常量和变量转换为它们中间数据长度最长的数据类型进行计算,但赋值操作按把赋值号右边类型转换成赋值号左边类型的原则进行。如果希望在一个算术表达式中按用户的要求确定结果的类型,则可以采用强制类型转换符号进行操作。

2.2.3 运算符

1. 算术运算

单目运算: $-$, $++$, $--$ 。

双目运算: $+$, $-$, $*$, $/$, $\%$ 。

2. 关系与逻辑运算

关系运算的结果是 1 和 0,若关系表达成立,则其值为 1,否则为 0。

逻辑运算的结果是真或假,如果为真,定义其值为 1,如果为假,定义其值为 0。

2.2.4 编程中应注意的事项

1. 用整型 (int、short、long) 说明变量时,要注意变量的取值范围,否则将出现意想不到的错误;字符常量和字符串常量是有区别的。

2. 运算符的优先级:关系运算符中的等号 ($=$) 不要写成赋值号;逻辑运算中,当有 $\&\&$ 和 $\|\$ 联结的表达式时,按从左至右顺序运算,且一旦能确定结果便终止后面的运算。

2.2.5 典型案例

案例一 计算面积

1. 问题描述

输入 r_1 、 r_2 , 求出圆形垫片面积 (网格部分), 如图 2-1 所示。

2. 具体分析

(1) 所求面积 s = 大圆面积 s_2 (半径 r_2) - 小圆面积 s_1 (半径 r_1)。

(2) 圆面积 = 半径 $\times$ 半径 $\times 3.14$ 。

(3) 程序接收 r_1 、 r_2 , 定义为 float 类型。

算法流程如图 2-2 所示。

3. 程序实现

```
/*sy2_1.c*/
#include<stdio.h>
#define P 3.14
int main()
{
    float r1,r2;
    double s1,s2,s;
    printf("Please enter r1,r2:\n" );
    scanf("%f%f",&r1,&r2);
    s2=r2*r2*P;
    s1=r1*r1*P;
    s=s2-s1;
```

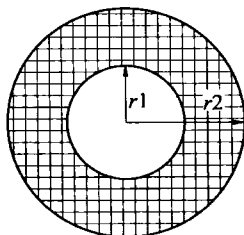


图 2-1 圆形垫片

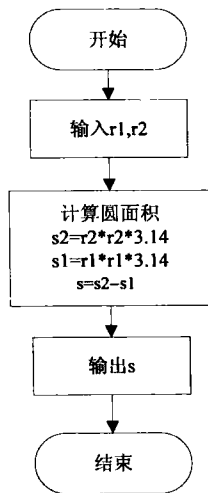


图 2-2 案例一算法流程

```
printf("s=%lf\n",s);
return 0;
}
```

4. 思考

如果输入的圆半径 r_1 大于 r_2 , 按程序计算出来会得负数, 要避免这种情况出现, 怎样修改程序?

案例二 温度计算

1. 问题描述

编写程序, 输入华氏温度 h , 输出摄氏温度 c (摄氏温度 = $5/9 * (\text{华氏温度} - 32)$)。

2. 具体分析

(1) 输入数据和输出数据均为温度, 因此用 `float` 型来定义, 其中, 输入数据为华氏温度, 输出数据为摄氏温度。

(2) 提示用户输入华氏温度, 并用 `scanf()` 语句接收该值。

(3) 根据华氏温度和摄氏温度之间的计算公式, 将华氏温度转化为摄氏温度, 并存储在变量 `c` 中。

(4) 用 `printf()` 语句显示对应的摄氏温度。

算法流程如图 2-3 所示。

3. 程序实现

```
/*sy2_2.c*/
#include<stdio.h>
int main()
{
    float h, c;
    printf("请输入华氏温度: ");
    scanf("%f", &h);
    c=5.0/9*(h-32);
    printf("\n 摄氏温度: %f\n", c);
    return 0;
}
```

4. 思考

若将上述程序改为输入摄氏温度, 输出华氏温度, 应如何改写程序?

案例三 自增运算符的使用

1. 问题描述

编写程序, 观察自增运算符放在变量之前和放在变量之后, 表达式的值有何不同。

2. 具体分析

(1) 设计两个整型变量 a 、 b , 赋相同的初值, 自增运算符分别放在两个变量的前面和后面。

(2) 再分别把两个表达式值赋给不同的整型变量 x 、 y 。

(3) 输出变量 x 、 y 的值, 观察自增运算符的使用, 及其位置对表达式值的影响。

算法流程如图 2-4 所示。

3. 程序实现

```
/*sy2_3.c*/
#include<stdio.h>
```

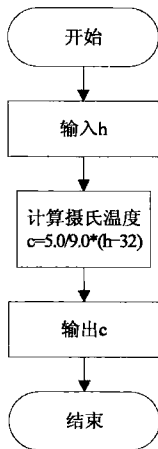


图 2-3 案例二算法流程

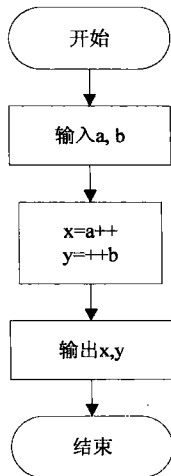


图 2-4 案例三算法流程