

21世纪高等学校计算机规划教材

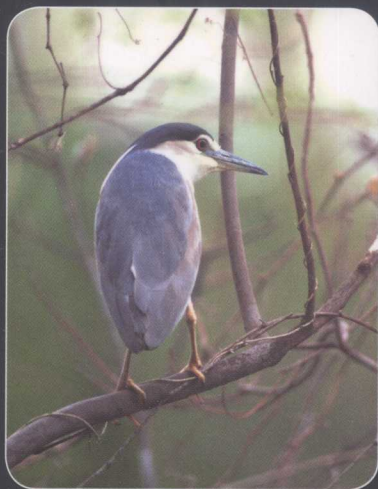
21st Century University Planned Textbooks of Computer Science

# C语言程序设计 习题解析与实验指导

The Answer and Practice of C Programming

朱立华 王立柱 主编 俞琼 郭剑 许棣华 编著

- 配合主教材 透彻解析课后习题
- 紧扣新考纲 补充大量典型习题
- 重编程能力 在实验中强化理论
- 学而后致用 理论实践能力并重



精品系列

 人民邮电出版社  
POSTS & TELECOM PRESS

21世纪高等学校计算机规划教材

21st Century University Planned Textbooks of Computer Science

# C语言程序设计 习题解析与实验指导

The Answer and Practice of C Programming

朱立华 王立柱 主编 俞琼 郭剑 许棣华 编著



精品系列

人民邮电出版社

北京

## 图书在版编目(CIP)数据

C语言程序设计习题解析与实验指导 / 朱立华, 王立柱  
主编; 俞琼等编著. —北京: 人民邮电出版社, 2009. 10  
21世纪高等学校计算机规划教材  
ISBN 978-7-115-21367-9

I. C… II. ①朱…②王…③俞… III. C语言—程序设计—  
高等学校—教学参考资料 IV. TP312

中国版本图书馆CIP数据核字(2009)第162815号

## 内 容 提 要

本书是《C语言程序设计》(朱立华、王立柱主编)一书的配套辅导教材。

全书共分三个部分: 第一部分是教材习题参考答案与解析, 对主教材后的每一个习题给出答案并结合教材内容给出透彻解析; 第二部分参考了最新的二级考试大纲, 对应于教材的章节, 补充了大量精选习题, 并给出了参考答案; 第三部分精心设计了10个实验, 在每一个实验中精选了具有代表性的实验题目, 选题体现分层思想, 对每一个实验题目给出了详细指导。

本书所有的程序以及实验都是基于 Visual C++ 6.0 开发环境。

本书内容紧扣C语言程序设计的相关知识点, 覆盖面广、题目量大、解析透彻、实验注重能力的培养, 是学习C语言程序设计的实用教材。

本书可作为C语言程序设计课程的辅导教材, 亦可供计算机编程爱好者自学使用。

21世纪高等学校计算机规划教材

## C语言程序设计习题解析与实验指导

- ◆ 主 编 朱立华 王立柱  
编 著 俞 琼 郭 剑 许 棣 华  
责任编辑 滑 玉  
执行编辑 武恩玉
- ◆ 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 电子函件 315@ptpress.com.cn  
网址 <http://www.ptpress.com.cn>  
北京艺辉印刷有限公司印刷
- ◆ 开本: 787×1092 1/16  
印张: 16  
字数: 425千字 2009年10月第1版  
印数: 1-4000册 2009年10月北京第1次印刷

ISBN 978-7-115-21367-9

定价: 27.00 元

读者服务热线: (010)67170985 印装质量热线: (010)67129223

反盗版热线: (010)67171154

# 前 言

C 语言作为读者学习高级语言程序设计的第一门语言,存在以下两大难点:概念虽然清楚但是不知如何运用,程序虽然能读懂但是不知如何编程。

出现这两大难点的主要原因是:缺乏一定的分析问题和运用知识的能力,缺少足够的实践与练习,开发工具的使用不熟练。

为了帮助读者学好 C 语言,掌握用 C 语言解答各种考题,编写出正确乃至完美的程序,本书重点解决好以下问题。

第一,概念如何用于解题。本书第一部分是主教材的习题参考答案与解析,通过对主教材的每一个习题的透彻解析,帮助读者更好地理解和运用所学的理论知识;本书的第二部分参考了最新的二级考试大纲,对应于教材的章节,补充了大量精选习题并给出了参考答案,使读者通过更多的练习,大量的训练,巩固所学知识,加强运用知识解题的能力。

第二,理论如何指导实践。本书的第三部分精心设计了 10 个实验,在每一个实验中精选了具有代表性的、能帮助读者掌握对应知识和技能题目。本书实验的一大特点是选材体现分层次思想:无“★”标识的题目要求每一个读者都必须完成,这样的训练能使读者掌握基本的编程方法;掌握程度更好的读者可以完成题目前面有“★”标识的具有较高难度的题目,这样的训练能使读者提高分析问题的能力,能写出尽可能好的程序。为使初学者在编程时少走弯路,帮助他们分析和解决问题,每一个实验题目给出了详细的指导,指出了思考时的重点与难点,强调了需要特别注意的问题。本书实验的另一大特点是,在每一次实验中循序渐进地提出了使用调试工具的要求,熟练地运用开发工具必定对解决编程中出现的各种问题大有帮助。通过实验的训练,读者将能熟练运用开发工具并在实践中灵活运用理论知识。

本书所有的程序都在 Visual C++ 6.0 下调试通过,每一道编程题都可能存在多种解答,读者可以进一步思考,拓宽自己的思路。

本书由朱立华、王立柱主编。第一部分由朱立华和王立柱共同完成;第二部分题目由俞琼初步搜集,其中第 1、2 章由丛莹编写,第 3 章由蒋亚南编写,第 4 章由刘冬梅编写,第 5~13 章由俞琼编写;第三部分的实验 1、2、3、6、10 由许棣华初步整理、朱立华改编,实验 4、5、7、8、9 由郭剑编写。全书由朱立华统稿。

由于编写时间紧,作者水平有限,书中难免存在不准确之处,敬请读者批评指正。

编 者

2009 年 7 月于南京

# 目 录

## 第一部分 教材习题参考答案与解析

第 1 章 计算机组成与程序设计语言	1	第 8 章 结构、联合、枚举	49
第 2 章 基本数据类型	4	第 9 章 字符串	57
第 3 章 表达式与运算符	9	第 10 章 二维数组与指针	66
第 4 章 程序流程控制	11	第 11 章 流与文件	77
第 5 章 一级指针与一维数组	21	第 12 章 高级程序设计	84
第 6 章 函数	30	第 13 章 C++初步	90
第 7 章 模块化程序设计相关	43		

## 第二部分 补充习题与解答

第 1 章 计算机组成与程序设计语言	93	第 8 章 结构、联合、枚举	152
第 2 章 基本数据类型	94	第 9 章 字符串	160
第 3 章 表达式与运算符	98	第 10 章 二维数组与指针	170
第 4 章 程序流程控制	103	第 11 章 流与文件	182
第 5 章 一级指针与一维数组	116	第 12 章 高级程序设计	194
第 6 章 函数	129	第 13 章 C++初步	205
第 7 章 模块化程序设计相关	143		

## 第三部分 实验指导

实验 0 Visual C++ 6.0 集成开发环境 (IDE) 简介	213	实验 6 结构体编程练习	239
实验 1 初识 VC++ 6.0 上机环境及顺序 结构编程练习	225	实验 7 字符串编程练习	241
实验 2 选择结构编程练习	227	实验 8 二维数组编程练习	243
实验 3 循环结构编程练习	230	实验 9 文件编程练习	245
实验 4 一级指针与一维数组编程练习	232	实验 10 设计性综合实验: 学生成绩 管理系统	247
实验 5 函数编程练习	234	参考文献	252

# 第一部分 教材习题参考答案 与解析

## 第 1 章 计算机组成与程序设计语言

### 一、单选题

1. “程序存储思想”是\_\_\_\_\_提出来的。

- A. 丹尼尔·里奇    B. 冯·诺依曼    C. 贝尔    D. 马丁·理查德

【参考答案】B

【解析】根据教材 1.1 节的知识，正确答案为 B。

2. 若计算机有 24 根地址总线，则其存储器必定有\_\_\_\_\_个字节的存储空间。

- A.  $2^{24}$     B.  $2^{16}$     C.  $2^{32}$     D. 不能确定

【参考答案】A

【解析】根据教材 1.1 节的知识，地址总线的多少决定了地址的多少，也就是字节的多少。如果有 16 根地址总线，则存储器最多有  $2^{16}$  个字节；因此按照同样的计算方法，有 24 根地址总线，则存储器最多有  $2^{24}$  个字节，因此答案为 A。

3. 第一个体现结构化程序设计思想的高级程序设计语言是\_\_\_\_\_。

- A. FORTRAN 语言    B. C 语言    C. PASCAL 语言    D. Java 语言

【参考答案】C

【解析】根据教材 1.4 节的知识，1970 年，第一个结构化程序设计语言——PASCAL 语言出现，标志着结构化程序设计时期的开始，因此答案为 C。

4. CPU 从专用寄存器\_\_\_\_\_依次提取指令执行。

- A. IX    B. IY    C. SP    D. PC

【参考答案】D

【解析】根据教材 1.1 节的知识，CPU 从程序计数器 PC 依次提取指令执行，因此答案为 D。

5. 以下关于源程序与目标程序的关系，不正确的是\_\_\_\_\_。

- A. 用机器语言编写的源程序就是目标程序  
B. 用汇编语言编写的源程序需要经过汇编程序汇编为目标程序  
C. 用 C 语言编写的源程序需要经过编译程序编译为目标程序  
D. 不同高级语言的编译器应该是一样的，都负责将源程序编译为目标程序

【参考答案】D

【解析】根据教材 1.3 节的知识，可知答案 A 和 B 是正确的说法；根据教材 1.4 节的知识，用高级语言编写的源程序不是计算机最终能识别并运行的目标程序，需要一种机制，可以把每一条指令转化为一组低级的机器语言指令，编译器提供了这种转化机制。不同的高级语言所需要的编

译程序也是不同的。因此答案 C 的说法是正确的。答案 D 说法不正确，为所选答案。

## 二、问答题

1. 何谓“程序存储思想”？

【参考答案】“程序存储思想”是由美国普林斯顿大学的冯·诺依曼于 1945 年最先提出的，其主要内容如下。

- (1) 计算机应该采用二进制，与十进制相比，实现二进制运算的结构简单，容易控制。
- (2) 操作指令也是一种信息，不妨用二进制代码表示。
- (3) 程序 and 数据的存储形式可以完全相同。
- (4) 程序本身也可包含数据。程序中的每一条指令由操作码和操作数两部分组成，前者是操作内容，后者是数据所在的存储单元地址或直接就是数据。

【解析】根据教材 1.1 节的知识。

2. 简述计算机的基本组成，以及每一部分的主要功能分别是什么？

【参考答案】冯·诺依曼型计算机硬件由 5 个部分组成：运算器、控制器、存储器、输入设备和输出设备，其系统结构如图 A1.1 所示。

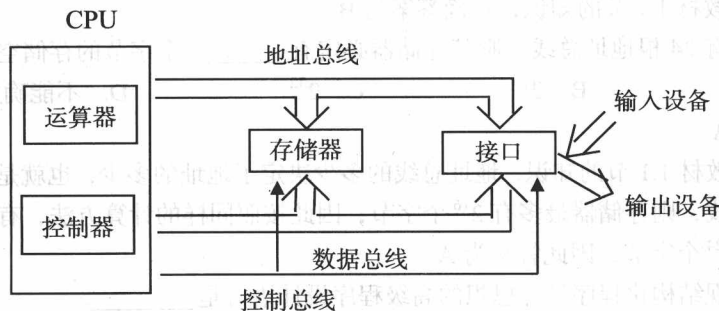


图 A1.1 电子计算机系统结构

**存储器**存放程序指令及数据。**运算器**进行各种算术运算和逻辑运算。**控制器**控制和指挥整个运算过程，使指令按要求一条一条执行。**输入设备**输入指令代码和原始数据，**输出设备**显示或打印计算结果。

**运算器和控制器**合称为中央处理器，简称 CPU。CPU 通过总线与存储器和接口交换信息。

【解析】根据教材 1.1 节的知识。

3. 什么是入口地址、断口地址、保护现场和恢复现场？

【参考答案】

**入口地址：**程序是一组指令，第一条指令的地址称为程序的入口地址。

**中断和断口地址：**一个程序在执行过程中去执行另一个程序，我们称之为中断，主程序中断时的下一条指令的地址称为主程序的断口地址。

**保护现场和恢复现场：**如果主程序中断时，有些寄存器还存放着中间结果，而子程序可能要占用这些寄存器，那么就要将主程序的中间结果放入异地（比如堆栈）暂存，把寄存器“让给”子程序的过程是保护现场。待子程序执行完毕，再将主程序的中间结果放回寄存器的过程是恢复现场。

【解析】根据教材 1.2 节的知识。

4. 简述 C 语言的优缺点。

【参考答案】C 语言有以下优点。

(1) 简洁紧凑、灵活方便：C 语言一共只有 32 个关键字，9 种控制语句，程序书写自由，主要用小写字母表示。

(2) 丰富的运算符：C 语言的运算符包含的范围很广泛，共有 34 个运算符。C 语言把括号、赋值、强制类型转换等都作为运算符处理。

(3) 丰富的数据类型：C 语言的数据类型有整型、实型、字符型、数组类型、指针类型、结构体类型、共用体类型等。被用来实现各种复杂的数据类型的运算。并引入了指针概念，使程序效率更高。

(4) C 语言是结构式语言：结构式语言的显著特点是代码及数据的分隔化，即程序的各个部分除了必要的信息交流外彼此独立。这种结构化方式可使程序层次清晰，便于使用、维护以及调试。C 语言是以函数形式提供给用户的，这些函数可方便的调用，并具有多种循环、条件语句控制程序流向，从而使程序完全结构化。

(5) C 语言语法灵活，程序设计自由度大：虽然 C 语言也是强类型语言，但它的语法比较灵活，允许程序编写者有较大的自由度。

(6) C 语言允许直接访问物理地址，可以直接对硬件进行操作：C 语言既具有高级语言的功能，又具有低级语言的许多功能，能够像汇编语言一样对位、字节和地址进行操作，而这三者又是计算机最基本的工作单元，可以用来写系统软件。

(7) C 语言程序生成代码质量高，程序执行效率高：一般只比汇编程序生成的目标代码效率低 10%~20%。

(8) C 语言适用范围大，可移植性好：C 语言有一个突出的优点就是适合于多种操作系统，如 DOS、UNIX，也适用于多种机型。

C 语言有以下缺点。

(1) C 语言的缺点主要表现在数据的封装性上，这一点使得 C 语言在数据的安全性上存在很大缺陷。

(2) C 语言的语法灵活，对变量的类型约束不严格，影响程序的安全性，对数组下标越界不作检查等。从应用的角度，C 语言比其他高级语言难掌握。

(3) C 语言指针是 C 语言的一大特色，可以说 C 语言优于其他高级语言的一个重要原因就是因为它有指针操作可以直接进行靠近硬件的操作，但是 C 的指针操作也给它带来了许多不安全的因素。

【解析】根据教材 1.4 节的知识。

5. 简述 C 语言程序的开发过程。

【参考答案】一个 C 语言程序从编写到执行出结果一般需要 6 个步骤：编辑 (edit)、预处理 (preprocess)、编译 (compile)、链接 (link)、装载 (load) 和执行 (execute)。下面是这几个步骤所完成的任务。

(1) **编辑**：编辑是将编写好的 C 语言源程序通过输入设备录入到计算机中，生成扩展名为“.c”的源文件。编辑源程序的方法有两种：一种是选用 C 语言集成开发环境中的编辑器，这是最常用的方法；另一种是使用其他文本编辑器，如写字板、记事本等。

(2) **预处理**：执行程序中的预处理指令（它们是为优化代码而设计的），每条预处理指令以符号#开头，结尾不带分号。预处理生成中间文件，又称**转译单元** (translation unit)。



(3) **编译**: 编译是将已生成的 C 语言源文件和预处理生成的中间文件转换为机器可识别的目标代码 (即二进制代码), 生成相应的 “.obj” 文件。在编译过程中主要进行词法和语法分析, 发现有不符合的, 及时以 error 或 warning 信息提示用户, 用户必须重新修改源程序文件直至编译正确才能进入下面的步骤。

(4) **链接**: 链接把不同的二进制代码片段 (C 语言程序一般都包含了多处定义的函数和数据, 它们都分别被编译成二进制代码片段) 连接成完整的可执行文件 (executable file)。可执行文件的扩展名为 “.exe”。

(5) **装载**: 装载器将可执行文件装入内存存储器。

(6) **执行**: 在中央处理器控制下, 对装入内存的可执行文件指令逐条执行。所得结果一般在显示器上显示。

【解析】根据教材 1.5 节的知识。

6. 简述源程序与目标程序的关系。

【参考答案】程序员用某种编程语言所编写的程序称为源程序; 计算机能直接执行的程序称为目标程序。目标程序的操作码和操作数都是 0/1 序列, 而源程序却不一定是 0/1 序列。源程序必须转换为目标程序, 计算机才能读懂并执行。

用机器语言编写的源程序就是目标程序; 用汇编语言编写的源程序必须经过汇编程序翻译成目标程序; 用高级语言编写的源程序必须经过特定的编译程序将源程序翻译为目标程序。

【解析】根据教材 1.3 节的知识。

## 第 2 章 基本数据类型

### 一、单选题

1. 以下哪一个可以作为正确的变量名\_\_\_\_\_。

- A. 3X                      B. file\_1                      C. int                      D. X+Y;

【参考答案】B

【解析】此题考查的是合法的用户自定义标识符的概念, 变量名就是一个用户自定义标识符。根据标识符的定义, 必须以字母或下划线开头, 后接字母、数字、下划线的任意序列。因此答案 A 的开头不符合要求, 答案 D 中也出现了加号不符合要求; 而对于用户自定义标识符还要求不选关键字, 故答案 C 也是错的, 只有答案 B 正确。

2. 下列哪一个是合法的实型字面常量\_\_\_\_\_。

- A. 8E3.1                      B. E5                      C. 234.                      D. 234

【参考答案】C

【解析】实型字面值常量有两种表示方法: 小数形式和指数形式。小数形式要求必须有小数点, 左右两边是数字序列, 最前面可以有正负号, 而且左或右的数字也可以缺省; 指数形式中必须有 E (或 e), 并且 E 的左右两边都不能缺省。E 左边为一个小数形式的实数或整数, E 右边只能是整数。根据这些知识, 答案 A 的错误在于 E 右边写成了小数; 答案 B 的错误在于 E 左边缺少内容; 答案 D 的错误在于没有小数点, 是一个整数; 故答案 C 正确, 是缺省了右边数字的小数形式的实数。

3. 下列哪一个整数值最大\_\_\_\_\_。

- A. 012                      B. 0x12                      C. 12                      D. 120

## 【参考答案】D

【解析】此题实际考查的是整型字面值常量的表示，其可以用八进制、十进制、十六进制表示。将这4个答案所表示的数字统一为十进制来理解就可以了。答案A以0开头，是八进制表示，012就是十进制的10；答案B以0x开头，是十六进制表示，0x12就是十进制的18；答案C和D本来就是十进制表示。因此比较后，答案D正确。

4. 下列哪一个字符与其他3个字符不相等\_\_\_\_\_。

- A. 'a'                      B. 'A'                      C. '\x41'                      D. '\101'

## 【参考答案】A

【解析】此题实际考查的是字符字面常量的表示方法。以一对单引号为定界符，里面可以直接是一个字符，也可以是以反斜杠开头的转义字符，而反斜杠后又有3种形式：直接跟一个字符表示某种用处，如'\n'表示换行；3位八进制数；以x开头后面跟两位十六进制数。本题中，字符A的ASCII码为65，八进制的101和十六进制的41都是65，这三者是相等的，表示同一个字符A。只有答案A表达的是字符a，为所选项。

5. 假设已有定义：typedef int INTEGER; int a=1; INTEGET b=2;下面4条赋值语句中错误的是\_\_\_\_\_。

- A. a=b\*20;                      B. b=b+1;                      C. INTEGER=a+b;                      D. a='a';

## 【参考答案】C

【解析】此题主要考查typedef的作用以及变量的相关知识。typedef的作用是给已有类型起个别名，本题中的INTEGER就是类型名int的别名，实质上是一个类型名而不是变量名，根据这一理解，再加上赋值语句要求赋值号的左边只能是变量，因此，答案C即为所选答案。答案A、B是正确的赋值语句不必多解释，答案D也是正确的赋值，因为右边的'a'在这条赋值语句中取其ASCII码值97，相当于a=97。

6. 一个C语言程序的执行总是\_\_\_\_\_。

- A. 从main()函数开始                      B. 从任意一条语句开始  
C. 从任何一个函数开始                      D. 从第一个函数开始，直到最后一个函数结束

## 【参考答案】A

【解析】main()函数是系统调用程序的入口，因此程序的执行总是从main函数开始，正常情况下也是结束于main()函数。当然，可能会因异常而提前终止，也可能因调用类似exit函数而在main函数结束之前结束程序的运行。

## 二、填空题

1. C语言程序中需要进行输入/输出处理时，必须包含的头文件是\_\_\_\_\_。

## 【参考答案】stdio.h

【解析】程序中如果要对数据进行输入/输出处理，在C语言中是通过调用一些特定的函数来实现的，如：printf、scanf、getchar、putchar等，这些函数在头文件stdio.h中被定义。

2. 用运算符\_\_\_\_\_可以计算某一数据类型的变量所占的内存字节数。

## 【参考答案】sizeof

【解析】sizeof运算符可以计算变量所占空间的大小，其运算对象除了变量名外，还可以是类型名、常量名或表达式，都是求该类型的数据在内存中所占的字节数。

3. 用printf输出一个double型数据，如果希望输出形式为指数形式，应该用格式说明符\_\_\_\_\_；如果希望输出形式为小数形式，可以用格式说明符\_\_\_\_\_或\_\_\_\_\_。两种形式

的区别是：前者小数点后 6 位不够时补 0，后者会去掉小数点后无效的 0。

【参考答案】%E（或%e）、%f、%G（或%g）

【解析】根据 printf 函数格式说明符的相关说明可得以上答案，需要强调的是，没有 %F 格式说明符。

4. 如果需要得到一个变量的地址，可以使用运算符\_\_\_\_\_，但是该运算符不能对字面值常量取地址。

【参考答案】&

【解析】根据取地址运算符的作用及常量与变量的区别可知。

5. 关键字\_\_\_\_\_的作用是给已存在的数据类型名引入一个别名，以后用该别名也可以定义它所代表的类型的变量。

【参考答案】typedef

【解析】根据 typedef 的作用可知。

### 三、读程序写结果

1. 写出下面程序的运行结果。

```
#include <stdio.h>
int main()
{
    int i=020,j=20,k=0x20;
    printf("%d,%d,%d\n",i,j,k);
    return 0;
}
```

【参考答案】16,20,32

【解析】C 语言中对于整型字面常量值可以用八进制、十进制、十六进制表示，程序中 i 用八进制的 20 初始化，该值为十进制的 16；j 直接用十进制的 20 初始化，k 用十六进制的 20 初始化，该值为十进制的 32。在 printf 中对 3 个变量所用的格式控制字符都是 %d，即以十进制形式输出 3 个变量的值，因此得到答案所示的结果，printf 中除了格式控制字符以外的其余字符原样输出，因此逗号也被输出。

2. 写出下面程序的运行结果。

```
#include <stdio.h>
int main()
{
    short int a=0x7FFF,b=a+1;
    printf("%d,%d\n",a,b);
    return 0;
}
```

【参考答案】32767, -32768

【解析】整数在计算机中以补码的形式存储，这样便于进行算术运算。有符号数的最高一位表示符号位，0 为正，1 为负。原码即除了最高一位表示符号位以外，其余各位就是该数绝对值的二进制表示。根据原码与补码的关系，正数的补码形式与原码完全相同，负数的补码：符号位为 1，其余各位为其原码的各位取反再加 1。如果已知负数的补码，则其绝对值为去掉符号位以外的其余各位取反再加 1。此题中，a 和 b 的类型为短整型，占两字节即 16 个二进制位，根据初始化，a

的补码 0 111 111 111 111 111 是正数, 且值为 32767; 则 b 的补码就等于 a 的补码加 1, 最终 b 的补码为 1 000 000 000 000 000, 最高位为 1, 是一个负数, 该数的绝对值为后 15 个 0 全部取反再加 1, 即 1 000 000 000 000 000, 绝对值为 32768, 因此变量 b 的值为 -32768。

3. 程序如下:

```
#include <stdio.h>
int main()
{
    char a,b,c,d;
    c=getchar();
    d=getchar();
    scanf("%c%c",&a,&b);
    putchar(a);
    putchar(b);
    printf("%c%c",c,d);
    return 0;
}
```

如果从键盘输入 (从下面一行的第一列开始)

```
1<回车>
234<回车>
```

则程序的输出结果为?

【参考答案】231

【解析】此题考查的是字符变量的输入、输出问题。getchar()函数不但能接收普通字符, 还能够接收空格、回车等特殊字符。所以本题中, c 获得值 1, 而 d 就获得了回车字符, a 获得值 2, b 获得值 3, 字符 4 是输入多余的, 被忽略。根据输出的顺序, 依次输出 a、b、c、d 这 4 个字符的值, 得到答案 231。注意, 从显示器上能看出换行符也输出了, 光标移到了下一行。

4. 程序如下:

```
#include <stdio.h>
int main()
{
    int a;
    float b,c;
    scanf("%2d%3f%4f",&a,&b,&c);
    printf("a=%4d,b=%f,c=%g\n",a,b,c);
    return 0;
}
```

如果从键盘输入 (从下面一行的第一列开始)

```
1234<回车>
56789<回车>
```

则程序的输出结果为?

【参考答案】 $a = 12$ ,  $b = 34.000\ 000$ ,  $c = 5678$

【解析】此题考查的是输入输出格式的控制, 在 `scanf` 中, `%md` 或 `%mf` 中的 `m` 是指该数所取的宽度, 按指定宽度顺序读取, 宽度达到为止或遇到数据结束符(空格、Tab、换行符)。该题数据输入分为两行, 第一行的前两位数送到了变量 `a`, 因为达到其规定的宽度 2; 第三位数字 3 就读取到下一个变量 `b` 中, 由于此行中只剩下两位数不足规定的宽度 3, 因此遇到换行符结束, `b` 的值为 34.0; 变量 `c` 的值从第二行读取, 在读满了四位数之后, 多余的第五位数字被忽略, 变量 `c` 的值为 5678.0。输出时, 根据格式控制字符的规定, `a` 以宽度 4 输出其十进制形式, `b` 以 `%f` 规定的小数形式输出, 小数点后 6 位, 不够自动补 0, `c` 以 `%g` 规定的小数形式输出, 省略小数点后无效的 0, 如果没有小数部分, 则小数点也省略。

#### 四、编程题

1. 编写程序, 测试以下类型在内存中所占空间大小: `char`、`int`、`short`、`long`、`unsigned int`、`float`、`double`、`long double`, 输出时给出较清晰的提示信息。

【参考答案】

```
#include <stdio.h>
int main()
{
    printf("sizeof(char) = %d\n", sizeof(char));
    printf("sizeof(int) = %d\n", sizeof(int));
    printf("sizeof(short) = %d\n", sizeof(short));
    printf("sizeof(long) = %d\n", sizeof(long));
    printf("sizeof(unsigned int) = %d\n", sizeof(unsigned int));
    printf("sizeof(float) = %d\n", sizeof(float));
    printf("sizeof(double) = %d\n", sizeof(double));
    printf("sizeof(long double) = %d\n", sizeof(long double));
    return 0;
}
```

【解析】此题考查的是通过 `sizeof` 运算符求各种类型的数据在内存中所占空间的大小, 同时, 会用 `printf` 输出想要输出的提示信息及运算结果。对 C 语言源程序的结构有个大致的了解, 会完整地定义 `main` 函数。

2. 编写程序, 定义一个 `int` 型变量 `a`, 一个 `float` 型变量 `b`, 一个 `double` 型变量 `c`, `a` 变量初始化, `b` 变量赋值, `c` 变量的值从键盘输入, 调用 `printf` 函数输出 3 个变量的值及它们在内存中的首地址, 输出时给出较清晰的提示信息。

【参考答案】

```
#include <stdio.h>
int main()
{
    int a = 20;
    float b;
    double c;
    b=12.345f;
```

```
scanf("%lf",&c);
printf("address of a is: %x, value of a is: %d\n",&a,a);
printf("address of b is: %x, value of b is: %f\n",&b,b);
printf("address of c is: %x, value of c is: %f\n",&c,c);
return 0;
}
```

【解析】此题考查的是常量与变量的基本使用，会定义各种基本类型的变量，对变量获得值用字面值常量进行初始化或赋值，从键盘读入变量的值，注意 scanf 的正确使用，正确使用格式串并对相应的变量取地址。通过 printf 中格式串的设置及对应输出项的不同，输出变量的地址及变量的值，从而理解二者的区别。

## 第3章 表达式与运算符

### 一、单选题

1. 关系运算符对两侧的运算对象的要求是\_\_\_\_\_。

- A. 只能是 0 或 1
- B. 两个运算对象必须属于同一种数据类型
- C. 只能是 0 或非 0 值
- D. 可以是任意合法的表达式，两者类型不一定相同

【参考答案】D

【解析】在 C 语言中，关系运算符两侧的运算对象为任意合法的表达式，类型上也可以不同。任何类型的值总能解释为 0 和非 0 两种值，0 为假，非 0 为真，但关系表达式最终的计算结果只有两种，如果是假，结果为 0；如果是真，结果为 1。

2. 数学公式  $5\sqrt{a} + b^4/(2d-1) - |c|$  的 C 语言表达形式是\_\_\_\_\_。

- A.  $5*\text{sqrt}(a) + \text{pow}(b,4)/(2*d-1) - \text{fabs}(c)$
- B.  $5*\text{sqrt}(a) + \text{pow}(4,b)/(2*d-1) - \text{fabs}(c)$
- C.  $5**a/2 + \text{pow}(b,4)/(2*d-1) - \text{fabs}(c)$
- D.  $5*\text{sqrt}(a) + b^4/(2*d-1) - c$

【参考答案】A

【解析】该题的数学式中有根号、平方、绝对值符号，在 C 语言中它们分别用库函数 sqrt、pow 和 fabs 来表示，所以答案 C、D 首先被排除。答案 A 和 B 的差别就是 pow 函数内的参数次序正好相反。pow 函数的原型：double pow(double x,double y)，其功能是计算  $x^y$  的值，因此， $b^4$  的正确表示为 pow(b,4)，故答案 A 正确。

3. 在下列 C 语言表达式中，表示关系 x 大于等于 y，y 大于等于 z，正确的是\_\_\_\_\_。

- A.  $x >= y >= z$
- B.  $(x >= y) \text{ AND } (y >= z)$
- C.  $x >= y \&\& y >= z$
- D.  $(x >= y) \& (y >= z)$

【参考答案】C

【解析】对于此题初学者非常容易选择答案 A，但是 A 答案的这种表达虽然语法上正确，但是逻辑上是错误的。当 y 不在 z 和 x 之间的时候也能得到 1，例如：x=10，z=1，y=-9， $x \geq y$  即  $10 \geq -9$  得到真值为 1，第二步计算就变成  $1 \geq 1$ ，结果仍为 1，因此切忌用答案 A 的形式来表达某值在一段范围内。答案 B 的错误在于出现了 C 语言中没有的运算符 AND，答案 D 的错误在于混淆

了位与运算符&和逻辑与运算符&&, 所以答案 C 正确。

4. 假设下面所有的变量均已定义为 double 型, 下列赋值表达式不正确的是\_\_\_\_\_。

- A.  $z*=2.8/4$       B.  $x=-x$       C.  $x=y=z$       D.  $x=3.5\%2$

【参考答案】D

【解析】该题考查赋值运算符及相关的算术运算符的使用, 答案 A 是复合赋值运算符, 先计算右边的除法, 当两个数据类型不同时, 先将 4 转化为 4.0 再进行运算得到结果 0.7, 再执行  $z = z * 0.7$  的赋值运算, 正确。答案 B 先对右边的部分进行运算, 变量 x 先执行自减, 将表达式的结果赋值给 x, 实际上直接写成  $-x$  能达到同样的效果, 这种用法不常见, 但是语法上是正确的。答案 C 是连续赋值, 按从右到左的顺序执行, 最终效果是 x、y 和 z 三个变量的值相等, 也是正确的。答案 D 的错误在赋值号右边, 运算符“%”要求两个运算对象都必须为整数, 3.5 不是整数, 故答案 D 为所选项。

5. 已知  $\text{int } b, a=12, n=5;$ , 则表达式  $a \% = (b = 2, n \% = b) + 4$  的结果是\_\_\_\_\_。

- A. 16      B. 2      C. 1      D. 4

【参考答案】B

【解析】本题主要考查复合赋值运算符及逗号运算符的优先级及运算方法。此题先计算圆括号内的逗号表达式, 结果为 1; 接着表达式就成为:  $a \% = 1 + 4$ , 根据计算方法, 先对复合赋值运算符右边部分进行计算, 得到 5; 最后一步:  $a \% = 5$ , 结果为 2, 所以选择答案 B。

## 二、填空题

1. 已知  $\text{int } a = 5, b = 4, c = 6, d;$ , 执行语句  $d = a > b ? (a > c ? a : c) : b$ , 则 d 的值为\_\_\_\_\_。

【参考答案】6

【解析】条件运算符是 C 语言中唯一的三目运算符, 允许发生嵌套。此题中, 最后一步是赋值运算, 赋值运算符的优先级低于条件运算符。括号部分先计算, 因此, 第一步计算  $(a > c ? a : c)$ , 即  $(5 > 6 ? 5 : 6)$  得到结果 6; 第二步计算  $a > b ? 6 : b$ , 即  $5 > 4 ? 6 : 4$ , 得到结果 6; 第三步计算  $d = 6$ , 因此变量 d 的值为 6。

2. 已知  $\text{float } x = 2.5, y = 4.7; \text{int } a = 7;$ , 则表达式  $x + a \% 3 * (\text{int})(x + y) \% 2 / 4$  的值为\_\_\_\_\_。

【参考答案】2.5

【解析】此题考查的是算术运算符的运用, 在双目算术运算符中, 有两个比较特殊: “%”要求两个运算对象均为 int 型, “/”当两个运算对象均为整型时结果一定是商的整数部分。当“+”、“-”、“\*”、“/”四个运算符运算对象的类型不一致时, 将进行自动类型转化——低类型自动转为高类型, 最后高类型的两个值进行运算。此题第一步计算  $(x + y)$  得到 7.2; 第二步用  $(\text{int})$  进行强制类型转化, 得到 7; 第三步计算  $a \% 3$  得到结果 1; 第四步计算  $1 * 7$  得到 7; 第五步计算  $7 \% 2$  得到结果 1; 第六步计算  $1 / 4$  得到 0; 最后一步计算  $2.5 + 0$ , 先类型自动转化变为  $2.5 + 0.0$ , 再求两个同类型值的和得到最终结果 2.5。

3. 已知  $\text{int } a = 5, b = 7;$ , 执行语句  $a += a ++ \parallel b ++$ , 则  $a =$ \_\_\_\_\_,  $b =$ \_\_\_\_\_。

【参考答案】7,7

【解析】此题比较综合, 由于复合赋值运算符优先级最低, 因此先计算赋值号右边表达式  $a ++ \parallel b ++$ , 这里对 a 和 b 都进行后缀 ++ 运算, 因此, 本题的第一步取 a 的原值 5, 此为非 0 值即代表逻辑真。根据“||”运算符的短路功能, “||”右边的  $b ++$  将不作运算, 变量 b 将保持原值 7, 整个逻辑表达式的结果为 1。注意, 变量 a 在逻辑表达式计算结束以后将按后“++”的要求进行自增 1 的运算, 在赋值之前, 变量 a 的值为 6, 最后的复合赋值表达式就是  $a += 1$ , 即  $a = a + 1$ , 因

此变量 a 的最终结果为 7。

4. 已知 x、a 为 int 型变量，则表达式  $x = (a = 3, a * 2, a + 6)$  的值为\_\_\_\_\_。

【参考答案】9

【解析】此题比较简单，但容易出错。逗号运算符原本优先级低于赋值运算符，但是通过括号提升了优先级，故此题最终的表达式是一个赋值表达式。第一步，先计算逗号表达式的值，即求 3,6,9 的值为 9，这里易将结果算成 12。因此要注意， $a * 2$  只是访问了原来 a 的值 3，并没有改变 a，所以最后一步  $a + 6$  的时候应该是  $3 + 6$  而不是  $6 + 6$ 。然后将 9 赋值给变量 x，因此整个表达式的结果就是 9，同时变量 x 的值为 9，变量 a 的值为 3。此题如果去掉括号，结果仍为 9，但那时应该理解为最后一步是逗号表达式的结果为 9。

三、将下列表达式表示成合法的 C 语言表达式

1.  $|(2 + b)(c - d)| + 20$

【参考答案】`fabs((2 + b)*(c - d)) + 20`

【解析】绝对值需要调用数学函数 `fabs`，乘号不能省略，小括号要对应。

2.  $(\ln x + \sin y) / 4$

【参考答案】`(ln(x) + sin(PI*y / 180)) / 4`

【解析】绝对值 C 语言表达式中的函数调用必须带括号，正弦函数的自变量为弧度，式中的 PI 应该是自定义的符号常量代表圆周率。注意最外层的一对括号不能丢失。

3.  $\frac{1}{1 + 1/x}$

【参考答案】`1 / (1 + 1/x)`

【解析】此题注意两点：除号的正确使用，分母整体用小括号括起来。

4.  $2\pi r$

【参考答案】`2*PI*r`

【解析】此题比较简单，主要注意两点： $\pi$  是数学符号，C 语言表达式中不可以使用，因此定义一个符号常量 PI 代表圆周率；乘号在 C 语言表达式中不能缺省。

## 第 4 章 程序流程控制

一、单选题

1. 若变量已正确定义，语句“`if(a>b) k=0; else k=1;`”和\_\_\_\_\_等价。

A. `k=(a>b)?1:0;`

B. `k=a>b;`

C. `k=a≤b;`

D. `k=a≤b?0:1;`

【参考答案】C

【解析】此题较灵活，首先需要正确理解双分支 if 语句，当  $a > b$  时， $k = 0$ ；而当  $a \leq b$  时， $k = 1$ 。其次需要理解条件运算符和关系运算符的计算方法，答案 A 的计算结果正好与题干相反，当  $a > b$  时， $k = 1$ ；而当  $a \leq b$  时， $k = 0$ ，故不是正确答案。答案 D 的结果与答案 A 实际上是一样的。答案 B，当  $a > b$  时，关系表达式的结果为真，得到值 1，否则为 0，这与答案 A、D 的结论是一样的。只有答案 C，当  $a \leq b$  时，关系表达式的结果为 1，赋值给变量 k，否则将 0 赋值给 k，与题干中的一致。



2. 已知  $\text{int } a = 0, b = 2;$ , 对于下列 if 语句正确的是\_\_\_\_\_。

```
if (a = b) printf("A is equal to B\n");
```

- A. 输出: A is equal to B, a 等于 b;                      B. 无输出  
C. 输出: A is equal to B, 但 a 不等于 b;                  D. 编译出错

【参考答案】A

【解析】此题是初学者在使用 if 语句的条件编程时最容易犯的一个逻辑错误, 本意应该是判断 a 是不是等于 b, 如果相等则输出对应的提示信息。此处误将判断相等的运算符 “=” 写成了赋值运算符 “=”, 而 C 语言中的条件是以 0 表示假, 非 0 表示真, 因此即使写错了, 语法上还是正确的。故此处先进行赋值, 使 a 的值等于 b 的值, 同时 2 就是赋值表达式的结果, 而  $\text{if}(a = b)$  实质上就是  $\text{if}(2)$ , 这里的 2 是非 0 值, 因此条件为真, 执行输出语句, 故选择答案 A。

3. 下列程序段执行后, 变量 w 的值为\_\_\_\_\_。

```
int x=0,y=10,z=20,w=100;
if(x) w=x;
else if(y) w=y;
    else if(z) w=z;
```

- A. 0                      B. 10                      C. 20                      D. 100

【参考答案】B

【解析】此题主要考查嵌套 if 语句的使用, 以及对 C 语言中逻辑真和假的正确理解。首先,  $\text{if}(x)$  相当于  $\text{if}(x != 0)$ , 而 x 变量的值为 0, 因此这个条件为假。进入第一个 else 分支,  $\text{if}(y)$  相当于  $\text{if}(y != 0)$ , 而 y 变量的值为 10, 因此条件为真, 故执行  $w = y;$  语句, 使 w 重新获得了值 10 覆盖了初始化得到的 100, 下一个 else 分支自然是不会执行的, 因此 w 的最终结果是 10。

4. 下面程序段的运行结果是\_\_\_\_\_。

```
int n,s=0;
for(n=0;n++<=2; ) ;
    s+=n;
printf("n=%d, s=%d\n",n,s);
```

- A.  $n=3, s=6$               B.  $n=3, s=3$               C.  $n=4, s=4$               D. 有语法错误

【参考答案】C

【解析】此题考查的是 for 语句的灵活性。for 语句中的表达式 1 为初始化语句, 只执行一次; 表达式 2 用于控制循环的条件, 条件为真则执行循环体; 表达式 3 是在循环体执行结束后自动执行的语句。此题有两处容易出错: 第一处, 循环体为空, 而不是  $s += n;$ ; 第二处, for 语句中的表达式 2 为  $n++ \leq 2$ , 这里用到了后缀 ++, 因此, 在判断  $n \leq 2$  结束后, 该条件的真与假将决定循环体是否执行, 循环是否结束, 但是 ++ 运算是一定要执行的。当  $n = 3$  时,  $n \leq 2$  的条件为假, 循环将结束, 但是 ++ 运算仍然继续执行的, 因此最终 n 的值为 4。下面的  $s += n;$  语句是循环结束后的第一条赋值语句, 只执行一次, 因此, s 的值为 4。

5. 假设 i 已定义为整型变量, 以下程序段中, while 循环的循环次数是\_\_\_\_\_。

```
i=0;
while(i<10)
{ if (i%2) continue;
```