

National Computer Rank Examination

全国计算机等级考试

二级教程

——C 语言程序设计 习题分析与解答

田淑清



高等教育出版社

全国计算机等级考试

二级教程

——C 语言程序设计习题分析与解答

田淑清

高等教育出版社

内 容 提 要

本书为教育部考试中心组织编写的《全国计算机等级考试二级教程——C 语言程序设计》(以下简称《教程》)一书的配套用书。全书共分十六章,每章与《教程》相对应,分别对《教程》各章后的习题进行了分析与答案,并指出每一题在《教程》中可参考的相应章节。对于 C 语言学习中的难点,如指针、数组等,本书在给出答案之余,也做了经验性总结,使读者在熟悉 C 语言语法规范的同时,能逐步掌握模块化程序设计的方法并了解相关的数据结构和算法。对报考 C 语言程序设计的考生自学与自我测试会有较大的帮助。

本书可作为全国计算机等级考试 C 语言程序设计的辅导书,也可供普通高等学校 C 语言程序设计课程使用。

图书在版编目(CIP)数据

全国计算机等级考试二级教程. C 语言程序设计习题
分析与解答/田淑清. —北京:高等教育出版社,
2002. 6

ISBN 7-04-011127-6

I. 全... II. 田... III. ①电子计算机—水平考试
—解题②C 语言—程序设计—水平考试—解题
IV. TP3

中国版本图书馆 CIP 数据核字(2002)第 030939 号

全国计算机等级考试二级教程——C 语言程序设计习题分析与解答
田淑清

出版发行 高等教育出版社
社 址 北京市东城区沙滩后街 55 号
邮政编码 100009
传 真 010-64014048

购书热线 010-64054588
免费咨询 800-810-0598
网 址 <http://www.hep.edu.cn>
<http://www.hep.com.cn>

经 销 新华书店北京发行所
印 刷 中国青年出版社印刷厂

开 本 787×1092 1/16
印 张 11.5
字 数 270 000

版 次 2002 年 6 月第 1 版
印 次 2002 年 6 月第 1 次印刷
定 价 16.00 元

本书如有缺页、倒页、脱页等质量问题,请到所购图书销售部门联系调换。

版权所有 侵权必究

前言

本书为教育部考试中心组织编写的《全国计算机等级考试二级教程——C 语言程序设计》(以下简称《教程》)一书的习题解答。全书不仅对《教程》各章后的所有习题给出了答案,而且指出了每一题在《教程》中应参考的章节,以便读者进行复习。

对于习题中的选择题,本书除了给出答案外,还对每一个选项都进行了详细分析,进而说明选择原因。

对于习题中的填空题,本书也是在给出正确答案后对答案进行了充分的解释和分析。

在以上两种题型中包含了 C 语言的基本语法知识以及编程的基本算法。希望通过这些习题能对《教程》中介绍的知识有进一步的理解。

对于习题中的编程题,本书不只是简单地给出程序,而是对每一题所涉及的数据结构、基本语法和算法,以及较后的模块化程序设计方法进行了详细的介绍,试图一步一步地引导初学者自己动手来编写程序。很多初学者对编程感到困难,不知如何下手,我们希望读者能跟随本书中提示的步骤在计算机上编写这些简单的程序,并对自己的程序进行调试。在编程时,由于大意或理解不当,程序在编译和运行的过程中发生错误是难免的,把有错误的程序调试成正确的,这就锻炼了读者发现错误和改正错误的能力。因此,我们要特别强调程序设计是一门实践性极强的课程,只有深入理解算法,勤动手编写,多上机调试,才能真正掌握程序设计技术。

编者

2002 年 4 月

407103

目 录

第一章	程序设计基本概念习题分析与解答	(1)
第二章	C 程序设计的初步知识习题分析与解答	(2)
第三章	顺序结构习题分析与解答	(11)
第四章	选择结构习题分析与解答	(26)
第五章	循环结构习题分析与解答	(36)
第六章	字符型数据习题分析与解答	(48)
第七章	函数习题分析与解答	(59)
第八章	指针习题分析与解答	(71)
第九章	数组习题分析与解答	(85)
第十章	字符串习题分析与解答	(115)
第十一章	对函数的进一步讨论习题分析与解答	(131)
第十二章	C 语言中用户标识符的作用域和存储类习题分析与解答	(142)
第十三章	编译预处理和动态存储分配习题分析与解答	(148)
第十四章	结构体、共用体和用户定义类型习题分析与解答	(156)
第十五章	位运算习题分析与解答	(166)
第十六章	文件习题分析与解答	(169)

第一章 程序设计基本概念

习题分析与解答

1.1 在 TURBO C 环境中用 RUN 命令运行一个 C 程序时,所运行的程序扩展名是_____。

【参考答案】 .EXE

【分析与解答】 相关内容请参考《全国计算机等级考试二级教程——C 语言程序设计》(以下简称《教程》)1.1.1 节。

一个可以执行的文件,其后缀必定是:.EXE。

1.2 C 语言源程序文件名的后缀是____[1]____,经过编译后,生成文件的后缀是____[2]____,经过连接后,生成文件的后缀是____[3]____。

【参考答案】 [1] .C [2] .OBJ [3] .EXE

【分析与解答】 相关内容请参考《教程》1.1.1 节。

1.3 结构化程序由____[1]____、____[2]____、____[3]____三种基本结构组成。

【参考答案】 [1] 顺序结构 [2] 选择结构 [3] 循环结构

【分析与解答】 相关内容请参考《教程》1.3 节。

第二章 C 程序设计的初步知识

习题分析与解答

一、选择题(单选题)

2.1 以下选项中正确的整型常量是

- A) 12. B) -20 C) 1,000 D) 4 5 6

【参考答案】 B)

【分析与解答】 相关内容请参考《教程》2.3.3 节。

- (1) 本题所涉及的是 C 语言中十进制整型常量。
(2) 选项 A) 的最后有一个小数点, 因此这是一个实型数。
(3) 选项 C) 的数字间夹着一个逗号, C 语言规定, 整型常量由 0~9 数字组成。
(4) 选项 D) 的数字间插入了空格, 在 C 语言中, 任何常量在各数字之间都不允许有空格。
- 2.2 以下选项中正确的实型常量是

- A) 0 B) 3. 1415 C) 0.329×10^2 D) .871

【参考答案】 D)

【分析与解答】 相关内容请参考《教程》2.4.1 节。

- (1) 选项 A) 是一个整型数, 实型数“零”应当写成: 0.0 或 0.。
(2) 选项 B) 在小数点后插入了空格, C 语言不允许在常量的中间插入空格, 因此不合法。
(3) 选项 C) 是一种书写形式, 在 C 程序中, 常量中不可以包含 $\times$ 号, 也不可能用上角标的形式表示一个指数。
(4) 选项 D) 是一个正确的实型常量。C 语言中, 实型常量可以由数字和一个小数点组成, 且小数点前可以没有数字。

2.3 以下选项中不正确的实型常量是

- A) $2.607E-1$ B) $0.8103e\ 2$ C) -77.77 D) $456e-2$

【参考答案】 B)

【分析与解答】 相关内容请参考《教程》2.4.1 节。

- (1) 选项 A)、C)、D) 均为正确的实型常量。
(2) 选项 B) 在常量间插入了空格, 因此不正确。注意, 在 C 语言程序中, 任何常量的中间都不能插入空格。

2.4 以下选项中不合法的用户标识符是

- A) abc.c B) file C) Main D) PRINTF

【参考答案】 A)

【分析与解答】 相关内容请参考《教程》2.2.1 节。

(1) C 语言中,标识符由字母、数字字符和下划线字符组成,但不能由数字字符开头。用户标识符不得与 C 语言的关键字相同。

(2) 选项 A) 中包含了点号,因此是非法的。

(3) 选项 B) 中的 file 和选项 D) 中的 PRINTF 是合法的用户标识符。

(4) 选项 C) 中,Main 不同于关键字 main,因为 C 语言中,同一个字母的大、小写被认为是不同的字符,因此,Main 是一个合法的用户标识符。

2.5 以下选项中不合法的用户标识符是

- A) _123 B) printf C) A\$ D) Dim

【参考答案】 C)

【分析与解答】 相关内容请参考《教程》2.2.1 节。

(1) 选项 A) 是合法的用户标识符,在 C 语言中,标识符不能由数字字符开头,可以由字母或下划线开头。

(2) 选项 B) 是 C 语言中的预定义标识符,是一个用于输出数据的库函数名,用户可以把预定义标识符用作用户标识符而不会发生语法错,但在同一程序中,printf 不能再作为输出函数使用。

(3) 选项 C) 中字符 \$ 不能作为标识符的组成部分,因此 \$ 是不合法的标识符。

(4) 选项 D) 是合法的用户标识符。

2.6 C 语言中运算对象必须是整型的运算符是

- A) % B) / C) ! D) **

【参考答案】 A)

【分析与解答】 相关内容请参考《教程》2.5.1。

(1) 选项 A) 中,% 是求余运算符,它的运算对象必须是整型。

(2) 选项 B) 中,/ 是除法运算符,它的运算对象可以是整型也可以是实型。

(3) 选项 C) 中,! 是求逻辑“非”的逻辑运算符,这是一个单目运算符,在它右边可以是任意合法的表达式。

(4) 在 C 语言中没有选项 D) 中所示的“** ”运算符。

2.7 可在 C 程序中用作用户标识符的一组标识符是

- | | | | |
|---------|-----------|--------|-------|
| A) void | B) as _b3 | C) For | D) 2c |
| define | _123 | - abc | DO |
| WORD | If | case | SIG |

【参考答案】 B)

【分析与解答】 相关内容请参考《教程》2.2.1 节。

(1) 选项 A) 中,void 是 C 语言的关键字,这是一个类型说明符,不能用作用户标识符。

(2) 选项 B) 中的 3 个标识符都可以做为用户标识符。

(3) 选项 C) 中,- abc 的第一个字符是减号(-);case 是关键字,它在 switch 语句中构成语句中的标号。

(4) 选项 D) 中, 2c 以数字字符开头, 这不符合 C 语言的语法。

2.8 若变量已正确定义并赋值, 符合 C 语言语法的表达式是

A) $a = a + 7;$

B) $a = 7 + b + c, a++$

C) $\text{int}(12.3\%4)$

D) $a = a + 7 = c + b$

【参考答案】 B)

【分析与解答】 相关内容请参考《教程》2.5、2.6、2.7 节。

(1) 选项 A) 中, $a = a + 7;$ 赋值表达式的最后有一个分号 (;), C 语言规定, 语句用分号结束, 所以 $a = a + 7;$ 是一条赋值语句, 而不是表达式。

(2) 选项 B) 中, $a = 7 + b + c, a++$ 是一个逗号表达式, 它由 $a = 7 + b + c$ 和 $a++$ 两个表达式组成, 前者是一个赋值表达式, 后者是一个自加 1 的赋值表达式, 所以它是一个合法的表达式。

(3) 选项 C) 中, $\text{int}(12.3\%4)$ 看似是一个强制类型转换表达式, 但语法规则规定, 类型名应当放在一对圆括号内才构成强制类型转换运算符; 因此写成: $(\text{int})(12.3\%4)$ 才是正确的。在使用强制类型转换运算符时, 需要注意运算符的优先级, 例如, $(\text{int})(3.6 * 4)$ 和 $(\text{int})3.6 * 4$ 中因为 (int) 的优先级高于 $*$ 运算符, 因此它们将有不同的计算结果。

(4) 选项 D) 中, $a = a + 7 = c + b$ 看似是一个赋值表达式, 但是在 $a + 7 = c + b$ 中, 等号的左边是一个算术表达式 $a + 7$; 按规定, 赋值号的左边应当是一个变量或是一个代表某个存储单元的表达式, 以便把赋值号的右边的值存放在该存储单元中; 因此赋值号的左边不可以是算术表达式, 因为它不能代表内存中任何一个存储单元。

2.9 以下叙述中正确的是

A) a 是实型变量, C 允许进行以下赋值 $a = 10$, 因此可以这样说: 实型变量中允许存放整型值。

B) 在赋值表达式中, 赋值号左边既可以是变量也可以是任意表达式。

C) 执行表达式 $a = b$ 后, 在内存中 a 和 b 存储单元中的原有值都将被改变, a 的值已由原值改变为 b 的值, b 的值由原值变为 0。

D) 已有 $a = 3, b = 5$, 当执行了表达式 $a = a + b, b = a - b, a = a - b$ 之后, 已使 a 中的值为 5, b 中的值为 3。

【参考答案】 D)

【分析与解答】 相关内容请参考《教程》2.6 节。

(1) 在判断选项 A) 时, 首先应当建立这样的基本概念: 整型变量中只能存放整型数, 实型变量中只能存放实型数。如果赋值号右边表达式的值与赋值号左边变量的类型不一致, C 编译程序将按“赋值兼容”的原则, 自动转换右边表达式所得值的类型, 使之与左边变量的类型一致: 例如, $a = 10$, C 编译程序将首先把 10 转换为 10.0, 然后进行赋值操作, 把 10.0 赋给实型变量 a, 所以在变量 a 中存放的是一个实型数, 而不是整型数。

(2) 在赋值表达式中, 赋值号左边只能是变量或者是代表某个存储单元的表达式。而不可以是任意表达式。所以选项 B) 中的叙述不正确。

(3) 执行表达式 $a = b$ 后, 将把变量 b 存储单元中的值赋给变量 a, 从而覆盖了变量 a 中原有的值。但并没有给变量 b 重新赋值, 因此变量 b 中原有的值没有改变。所以选项 C) 中的叙述

也不正确。

(4) 在选项 D) 中, $a = a + b, b = a - b, a = a - b$ 是一个逗号表达式。执行了 $a = a + b$ 后, 变量 a 的值为 8; 接着执行了 $b = a - b$ 后, 变量 a 中的值不变, 仍为 8, 而 b 中的值已变成了 3; 最后执行了 $a = a - b$ 后, 变量 b 中的值不变, 仍为 3, 而 a 中的值已变成了 8。由此可见, 这个逗号表达式实现了两个变量中值的交换操作。

2.10 以下叙述中正确的是

- A) 在 C 程序中无论是整数还是实数, 只要在允许的范围内都能准确无误的表示。
- B) C 程序由主函数组成。
- C) C 程序由函数组成。
- D) C 程序由函数和过程组成。

【参考答案】 C)

【分析与解答】 相关内容请参考《教程》2.1 和 2.4.2 节。

(1) 在计算机内部由于整数和实数的存放形式不同, 因此内存中可以精确地存放整型数而不能精确地存放一个实型数。读者在程序设计中, 需记住实型数是存在误差的, 而整数没有误差。因此选项 A) 中的叙述不正确。

(2) 一个 C 程序可以包含任意多个函数, 但必须有一个、而且只能有一个主函数。因此可以说 C 程序是由函数组成的。选项 C) 中的叙述是正确的, 而选项 B) 中的叙述不正确。

(3) C 程序中没有“过程”这一结构, 因此选项 D) 中的叙述是错误的。

2.11 TURBO C 中, int 类型变量所占字节数是

- A) 1
- B) 2
- C) 3
- D) 4

【参考答案】 B)

【分析与解答】 相关内容请参考《教程》2.3.3 节。

TURBO C 中, int 类型变量占 2 个字节, 它们数值范围是 32767 ~ -32768。

2.12 不合法的八进制数是

- A) 0
- B) 028
- C) 077
- D) 01

【参考答案】 B)

【分析与解答】 相关内容请参考《教程》2.3.1 节。

(1) 八进制数由八进制数数字组成, 八进制数的数字有: 0、1、2、3、4、5、6、7。

(2) 在 C 语言中八进制数的表示必须用数字 0 开头, 其后跟八进制数数字。例如, 077 是八进制数, 而 77 则是十进制整数。

(3) 在选项 A)、C)、D) 中所列出的都是 C 语言中的八进制数。而选项 B) 中的 028, 虽是用 0 开始, 但其中的“8”不是八进制数数字, 因此不合法。

2.13 不合法的十六进制数是

- A) 0xff
- B) 0Xabc
- C) 0x11
- D) 0x19

【参考答案】 A)

【分析与解答】 相关内容请参考《教程》2.3.1 节。

(1) 十六进制数由十六进制数数字组成, 十六进制数的数字有: 1、2、3、4、5、6、7、8、9、a、b、c、

d、e、f。十六进制数中代表数字的字母也可以用大写字母。

(2) 在 C 语言中十六进制数的表示必须用数字 0 和字母 x(或大写字母 X)开头,其后跟十六进制数数字。例如,0XAF 是十六进制数,而 X0AF 和 xaa 则不是十六进制数。

(3) 在选项 B)、C)、D)中所列出的都是 C 语言中的十六进制数。而选项 A)中的 0xff 把开头的数字 0 误写成字母 o,因此是错误的。

2.14 若有定义:long k;,则能使 k 得到 32800 的赋值表达式是

A) $k = 328 * 100$

B) $k = 32700 + 200$

C) $k = 328, k * = 100$

D) $k = 30000 + 2800$

【参考答案】 C)

【分析与解答】 相关内容请参考《教程》2.3.3、2.5.1 和 2.6.3 节。

(1) C 语言规定,表示一个长整型常量时应当在最后加一个字母 L(或小写的 l)。但在发现一个整数的值超过 32767 时,如 40000,则自动将其转换为长整型数。

(2) 在选项 A)中,两个运算数都是 int 类型的常量,按 C 语言规定:双目运算符两边运算数的类型一致时,所得结果类型与运算数的类型一致。所以 $328 * 100$ 所得结果也是 int 类型的值,但这个值已超过 int 类型的最大值 32767,不可能得到 32800,所得结果不可预测,因此不可能给 k 赋 32800。

(3) 在选项 B)和 D)中,赋值号右边的表达式与选项 A)中表达式有相同的问题,因此都不能给 k 赋 32800。

(4) 在选项 C)中,是一个逗号表达式,在其中进行了赋值运算。第一个赋值运算 $k = 328$ 中,把 328 赋给了长整型变量 k。C 语言规定,若赋值号两边的类型不同,系统将自动把右边的值按左边变量的类型进行转换。所以 328 先转换成长整型后赋给了 k。表达式 $k * = 100$ 相当于 $k = k * 100$,按 C 语言规定:双目运算符两边运算数的类型不一致时,短整型数先转换成长整型数,然后进行运算;所以 100 先转换为长整型数,与 k 中的 328 相乘,所得结果 32800 也是长整型数,因此 k 中最后得到 32800。

二、填空题

2.15 若 k 为 int 整型变量且赋值 11,请写出运算 $k++$ 后表达式的值 [1],和变量 k 的值 [2]。

【参考答案】 [1] 11 [2] 12

【分析与解答】 相关内容请参考《教程》2.7.1 节。

$++$ 是自加运算符,++ 在变量 k 的后面时,取 k 的原始值作为表达式 $k++$ 的值,因此在 [1] 处应填 11。然后进行自加运算使 k 的值增 1,因此变量 k 的值便成了 12,[2]处应填 12。

2.16 若 x 为 double 型变量,请写出运算 $x = 3.2, ++x$ 后,表达式的值 [1] 和变量 x 的值 [2]。

【参考答案】 [1] 4.2 [2] 4.2

【分析与解答】 相关内容请参考《教程》2.7.1 节和 2.7.2 节。

回答本题时需要具有以下几个基本概念:

(1) $x = 3.2, ++x$ 是一个逗号表达式。

(2) 任意一个表达式都有一个值。

(3) 逗号表达式取最右边表达式的值作为逗号表达式的值;因此在这里是取 $++x$ 的值作为 $x = 3.2, ++x$ 的值。

(4) 自加($++$)和自减($--$)运算的运算对象可以是整型变量,也可以是实型变量。

(5) $++$ 在变量 x 的前面时,取 x 自加后的值作为表达式 $++x$ 的值。

2.17 函数体由符号 [1] 开始,用符号 [2] 结束。函数体的前面是 [3] 部分,其后是 [4] 部分。

【参考答案】 [1] { [2] } [3] 定义 [4] 执行

【分析与解答】 相关内容请参考《教程》2.1 节。

函数体必须用左花括号($\{$)开始,用右花括号($\}$)结束。C 语言遵循“先定义后使用”的原则,在函数体内使用的变量应当在使用之前进行定义;因此,所有的定义语句必须放在函数体的前部(定义部分),其后是可执行语句(执行部分)。

2.18 C 语言中的标识符可分为 [1]、[2] 和预定义标识符 3 类。

【参考答案】 [1] 关键字 [2] 用户标识符

【分析与解答】 相关内容请参考《教程》2.2.1 节。

C 语言中的标识符可分为关键字(程序专用)、用户标识符和预定义标识符 3 类。

2.19 在 C 语言程序中,用关键字 [1] 定义基本整型变量,用关键字 [2] 定义单精度实型变量,用关键字 [3] 定义双精度实型变量。

【参考答案】 [1] int [2] float [3] double

【分析与解答】 相关内容请参考《教程》2.3.2 节和 2.4.2 节。

在 C 语言程序中,用关键字 int 定义基本整型变量,用关键字 float 定义单精度实型变量,用关键字 double 定义双精度实型变量。

2.20 把 a1、a2 定义成单精度实型变量,并赋初值 1 的定义语句是_____。

【参考答案】 float a1 = 1.0, a2 = 1.0; 或 float a1 = 1, a2 = 1; (系统将自动把 1 转换为 1.0)

【分析与解答】 相关内容请参考《教程》2.3.2 节和 2.4.2 节。

2.21 C 程序中定义的变量代表内存中的一个_____。

【参考答案】 存储单元

【分析与解答】 相关内容请参考《教程》2.2.4 节。

2.22 表达式 $3.5 + 1/2$ 的计算结果是_____。

【参考答案】 3.5

【分析与解答】 相关内容请参考《教程》2.5.1 节。

回答本题时要求掌握以下几个基本概念:

(1) $3.5 + 1/2$ 是一个算术表达式。

(2) 按四则运算规则,应当先做除法($/$)后做加法($+$);因此以上表达式应当先做 $1/2$ 。

(3) 算术运算符两边运算数的类型必须一致才能进行运算,所得结果的类型与运算数的类型一致。因此, $1/2$ 的值是 0 而不是 0.5。

(4) 若算术运算符两边运算数的类型不一致,一边是整数,一边是实数时,系统将把整数转换为实数,然后对这两个实数进行运算;因此, $3.5 + 0$ 将首先转换成 $3.5 + 0.0$, 然后计算出结果: 3.5 。

2.23 对以下数学式,写出 3 个等价的 C 语言表达式 [1]、[2]、[3]。

$$\frac{a \cdot b}{c}$$

【参考答案】 [1] $a * b / c$ [2] $a / c * b$ [3] $b / c * a$

【分析与解答】 相关内容请参考《教程》2.5.2 节。

在 C 语言中“乘号”和“除号”的优先级相等,运算的顺序(结合方向)是从左到右。按此规则,以上代数式可以写成的 C 语言表达式可以是: $a * b / c$ 、 $a / c * b$ 或者 $b / c * a$ 。填空的顺序任意。

2.24 表达式 $s = 10$ 应当读做“_____”。

【参考答案】 把 10 赋给变量 s

【分析与解答】 相关内容请参考《教程》2.6.1 节。

赋值表达式的作用是把赋值号右边的值赋予赋值号左边的变量;因此,表达式 $s = 10$ 应当读做“把 10 赋给变量 s”,而不应当读成“s 等于 10”。

2.25 计算机中内存储器的最小存储单位称 [1],其中能容纳的数是 [2]。

【参考答案】 [1] 位 [2] 0 或 1

【分析与解答】 相关内容请参考《教程》2.3.4 节。

2.26 通常一个字节包含 [1] 个二进制位。在一个字节中能存放的最大(十进制)整数是 [2]、最小(十进制)整数是 [3]、它的二进制数的形式是 [4]。

【参考答案】 [1] 8 [2] 127 [3] -128 [4] 11111111

【分析与解答】 相关内容请参考《教程》2.3.4 节。

(1) 一个字节包含 8 个二进制位。

(2) 一个字节中能存放的最大(十进制)整数是 127。它的二进制数的形式是 01111111。

(3) 一个字节中能存放的最小(十进制)整数是 -128。它的二进制数的形式是 11111111。

(4) 用最高位上的数表示正数或负数,是 0 时,表示正数,1 表示是负数。

2.27 当计算机用两个字节存放一个整数时,其中能存放的最大(十进制)整数是 [1]、最小(十进制)整数是 [2]、它的二进制数的形式是 [3]。

【参考答案】 [1] 32767 [2] -32768 [3] 1111111111111111

【分析与解答】 相关内容请参考《教程》2.3.4 节。

(1) 两个字节能存放的最大(十进制)整数是 32767。它的二进制数的形式是 0111111111111111,共 16 位,最高位为 0,表示是正数。

(2) 两个字节能存放的最小(十进制)整数是 -32768。它的二进制数的形式是 1111111111111111,共 16 位,最高位为 1,表示是负数。

2.28 在 C 语言中,整数可用 [1] 进制数、[2] 进制数和 [3] 进制数三种数制表示。

【参考答案】 [1] 十 [2] 八 [3] 十六

【分析与解答】 相关内容请参考《教程》2.3.1 节。

(1) 在 C 语言中,整数可用十进制、八进制和十六进制 3 种数制表示。

(2) 用以表示八进制数的数字是 1~7,C 语言中,八进制数一定要用 0 开头,如 051,以便与十进制数 51 区别。

(3) 用以表示十六进制数的数字是 1~7 和字母 a~f(可用大写字母),十六进制数要用 0x 或 0X 开头(第一个字符是数字“0”)。

2.29 在 C 语言中,十进制数 30 的八进制表示形式是 [1],十六进制表示形式是 [2],在内存中它的二进制表示形式是 [3]。

【参考答案】 [1] 036 [2] 0x1e [3] 0000000000011110

【分析与解答】 相关内容请参考《教程》2.3.1 和 2.3.4 节。

三、上机改错题

2.30 请指出以下 C 程序的错误所在(为了便于说明,在每一行的最前面加了序号)

```
1    # include stdio.h;
2    main();          / *   main function   * /
3    float  r,s;      /* /* r is radius * / , /* s is area of circular * / * /
4    r = 5.0;
5    s = 3.14159 * r * r;
6    printf("%f \n",s)
```

【分析与解答】

第 1 行的错误:

(1) include 是一个程序行,因此在此行的最后不应当有分号(;)。

(2) include 程序行中后面的 stdio.h 是一个文件名,按规定,文件名应当放在一对双引号(" ")内,或放在一对尖括号(< >)内。

第 2 行的错误:

(1) main()是主函数的起始行,不是语句,因此最后不应当有分号(;)。

(2) 在同一行上的/ * main function * / 显然是注释;C 语言规定:注释由/* 开头,由*/ 结束,但在* 号和/ 之间不得插入任何空格,而在此处“/ * ”和“/ * ”之间存在空格,因此,/ * main function * / 的写法是错误的,而应写成:/* main function */。

第 3 行的错误:

(1) 在主函数的起始行 main()的后面是函数体,函数体由左花括号({)开始。但在函数体的第一行:float 的前面缺少了左花括号({)。

(2) 在同一行上的/* /* r is radius * / , /* s is area of circular * / * / 显然是注释;C 语言规定:注释由/* 开头,由*/ 结束,并且开头的/* 将去找最近的*/ 去配对,因此在/* /* r is radius * / 中,第一个/* 与最后的那个*/ 配上了对,结果紧跟在后面的那个逗号(,)落在了注释的外面,而构成了一个多余符号,为此,在编译时将报告“语法错”。

第 6 行的错误:

(1) printf("%f \n",s)应当是一条输出语句,但在最后缺少了一个分号。

(2) `printf("%f \n",s);`是程序的最后一条语句,程序应当结束;但缺少了程序体结束所需的右花括号`}`;此右花括号可以放在 `printf("%f \n",s);`的后面,也可以放在 `printf("%f \n",s);`的下一行上。

2.31 请指出以下 C 程序的错误所在

```
1    # include    "stdio.a"
2    main                      /* main function */
3    {    float    a,b,c,v;    /* a,b,c are sides, v is volume of cube */
4        a = 2.0;  b = 3.0;  c = 4.0
5        v = a * b * c;
6        printf("%f \n",v)
7    }
```

【分析与解答】

第 1 行的错误:在 include 行中的文件名应该是 `stdio.h`,而不是 `stdio.a`。

第 2 行的错误:在 main 的后面缺少一对圆括号。

第 4 行的错误:在 `c = 4.0` 的后面缺少分号。

第 6 行的错误:在 `printf("%f \n",v)`的后面缺少分号。

【参考答案】 D)

【分析与解答】 相关内容请参考《教程》3.1 节。

(1) 选项 A) 中是一个合法的赋值表达式, 但不是 C 语言的赋值语句, 因为最后没有构成语句所必要的分号。

(2) 选项 B) 中虽然最后有一个分号, 但赋值号右边的强制类型转换符是错误的, 应该写成: `(int)(a + b)`, 当使用强制类型转换符时, 类型名一定要用一对圆括号括起来。

(3) 选项 C) 是一个逗号表达式, 但最后没有分号, 因此不是语句。

(4) 选项 D) 是一个由自减运算符构成的赋值表达式, 且用分号结束, 因此这是一个合法的赋值语句。

3.4 下面程序:

```
main()  
{  
    int x = 10, y = 3;  
    printf("%d \n", y = x/y );  
}
```

的输出结果是

A) 0

B) 1

C) 3

D) 不确定的值

【参考答案】 C)

【分析与解答】 相关内容请参考《教程》3.2.1 节。

本题输出语句中的输出项是一个赋值表达式: `y = x/y`, 输出的是这个赋值表达式的值, 按规定, 赋值表达式的值就是赋值号的左边变量的值; 按优先级的规则, 先求出赋值号的右边 `x/y` 的值, 是整数 3, 然后赋予左边的 `y`, 因此, 赋值表达式 `y = x/y` 的值为 3; 输出语句中正确使用了格式说明: `%d`, 因此能正确输出整数 3。

3.5 若变量已正确说明为 `int` 类型, 要从键盘给 `a`、`b`、`c` 输入数据, 正确的输入语句是

A) `read( a,b,c );`

B) `scanf("%d%d%d", a,b,c);`

C) `scanf("%D%D%D", &a,&b,&c);`

D) `scanf("%d%d%d", &a,&b,&c);`

【参考答案】 D)

【分析与解答】 相关内容请参考《教程》3.3.1 节和 3.3.2 节。

(1) 在选项 A) 中用 `read` 从键盘输入数据, 这在 C 语言中是错误的。

(2) 在 `scanf` 语句中, 各输入项只能是合法的地址表达式, 而在选项 B) 中的输入项 `a`、`b`、`c` 是变量名, 因此是错误的。

(3) 在选项 C) 的 `scanf` 语句中, 虽然各输入项是合法的地址表达式: `&a`、`&b`、`&c`; 但对于 `int` 整型数使用了 `%D` 格式转换说明, `D` 不是合法的格式字符, 应当用 `d`。

(4) 选项 D) 中的是一条合法的输入语句。

3.6 若变量已正确说明为 `float` 类型, 要通过输入语句: `scanf("%f %f %f", &a, &b, &c);` 给 `a` 赋予 10.0、`b` 赋予 22.0、`c` 赋予 33.0, 以下不正确的输入形式是

A) 10

B) 10.0,22.0,33.0

22

33