

谭浩强 / 主审

计算机等级考试题典

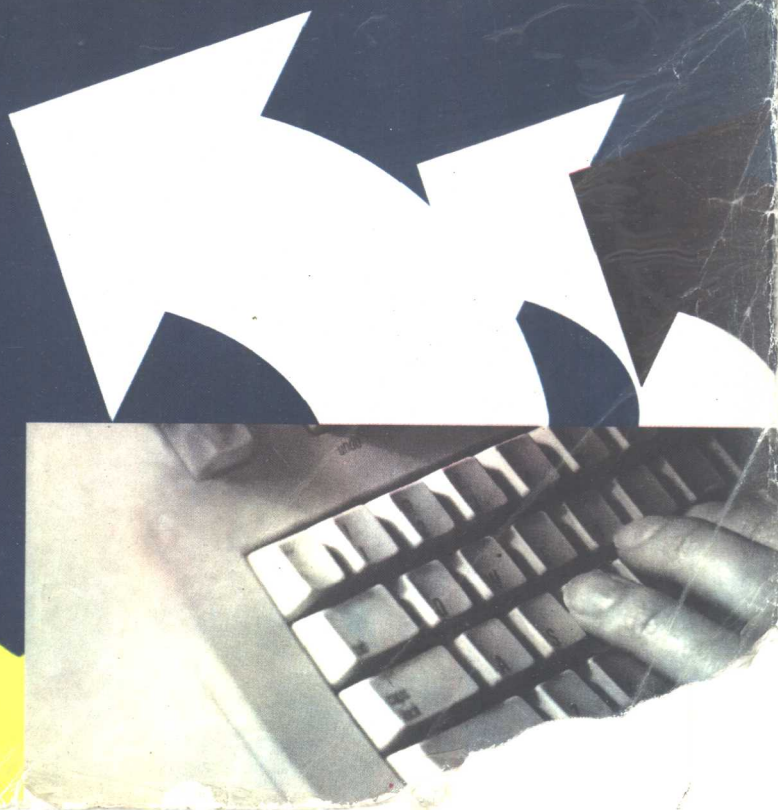
牛连强 王溪波 / 编著

● 例题精解 ● 权威辅导 ● 预测模拟 ● 顺利过关

C语言程序设计 重点难点强化训练

2级

大连理工大学出版社



谭浩强 主审

计算机等级考试题典

C 语言程序设计 重点难点强化训练

牛连强 王溪波 编著

大连理工大学出版社

《计算机等级考试题典》丛书编委会

主 审 谭浩强
主 任 郜荣春
副主任 赵维俊 刘晓晶
编 委 郜荣春 赵维俊 刘晓晶 牛连强
王溪波 李延珩 刘旭敏 司 丹
韩 露 李 鸽

丛书策划: 刘晓晶

图书在版编目(CIP)数据

C 语言程序设计重点难点强化训练/牛连强, 王溪波编著. —大连: 大连理工大学出版社, 1998. 6

计算机等级考试题典

ISBN 7-5611-1413-3

I. C… I. ①牛… ②王… II. C 语言-程序设计 N. TP312

中国版本图书馆 CIP 数据核字(98)第 00918 号

大连理工大学出版社出版发行

(大连市凌水河 邮政编码 116024)

大连业发印刷厂印刷

开本: 787×1092 毫米 1/16 字数: 470 千字 印张: 20.5

印数: 1—6000 册

1998 年 6 月第 1 版

1998 年 6 月第 1 次印刷

责任编辑: 李 鸽

责任校对: 胡 琦

封面设计: 孙宝福

定价: 19.00 元

出版说明

随着我国计算机应用技术的迅速发展,计算机已成为各行各业最基本的工具之一。社会的进步和经济的发展对计算机技术的依赖程度越来越大。作为国家经济信息化的核心技术,计算机技术将密切地同社会经济及文化生活联系在一起。计算机技术带动的高新技术正在不断地改变着人们的工作方式、生活方式、学习方式及思维方法。一个国家拥有计算机的数量和质量及计算机应用的广度和深度,不仅标志着一个国家的发达程度,而且影响着整个国家的现代化进程。

因此,当今社会对人才的素质培养和知识结构提出了全新的要求。各行各业的人员不论年龄、专业和知识背景如何,都应掌握和应用计算机,以便提高工作效率和管理水平。既掌握一定的专业技术,又具备计算机应用能力的人员越来越受到用人单位的重视和欢迎。21 世纪将是信息时代,计算机技能与外语技能一样,是跨世纪人才的必备条件,是当今世界的“第二文化”。

国家教委考试中心顺应社会发展的需要于 1994 年推出“全国计算机等级考试”,其目的是以考促学、向社会推广普及计算机知识,为选拔人才提供统一、公正、客观和科学的标准。现在全国每年都有几十万人参加这种考试。

参加全国等级考试的许多人认为这种考试与传统考试不同,除指定的教材外,感到缺少关于应试指南以及试题分析方面的资料,为此,我们为配合社会各类人员参加考试,并能顺利通过“全国计算机等级考试”,组织多年从事辅导计算机等级考试的专家在对近几年的考题深刻分析、研究基础上,编写出这套为应考者参加考试的导向辅导教材。在编写过程中,以“全国计算机等级考试”大纲为基础,以例题方式介绍相关知识、解题思路,并配以一定数量练习题,使应考者学习本书后,在短时间内,对应考内容的重点难点加深理解,配合精选的例题与习题,对考生能起到强化训练的作用,从而达到增强“全国计算机等级考试”适应能力,提高通过率的目的。而且还能在达到扩大计算机知识面的同时,提高实际操作编程、调试能力。

我们这套丛书分为《一级、二级基础重点难点强化训练》《C 语言程序设计重点难点强化训练》及《FoxBASE+ 程序设计重点难点强化训练》等三种。

为了提高图书的质量,我们特邀请计算机教育专家谭浩强教授主审了这套书。希望这套书能成为广大应考者的良师益友,并能顺利通过考试。

编委会

1998 年 5 月

前 言

当今,计算机应用的普及,极大地推动了人们学习计算机知识的热情,计算机知识水平的高低已成为衡量新型人才的重要尺度。国家教委在 1994 年举办了面向社会的“全国计算机等级考试”,而许多省市也在举办类似的计算机等级测试,吸引了众多的考生,某种意义上也形成了一种衡量一个人的计算机知识掌握程度的标准。

C 语言是国家计算机等级考试中的一个语种,有着极广泛的应用价值。从学习的角度看,C 语言复杂于其他计算机算法语言或数据库,语言中的特殊语法现象、难点内容也较其他语种为多。如何利用有限的时间,尽快地掌握 C 语言的精髓,顺利通过等级测试是考生所关注的问题,也是我们编写本书的目的。

本书不同于一般的语法类书籍,它通过典型试题分析的方式,按块组织内容,目的是向读者集中、清晰地展示应该掌握的重点难点知识。这些内容既互相独立,又根据循序渐进的学习过程彼此衔接,对于一个典型试题,书中不仅分析了题目的理解方法和答题技巧,还特别注重相关知识点介绍,以使考生能够抓住重点内容并形成整体概念。

本书主要内容共分为四部分,包括应试指南、笔试知识分析、上机试题分析及模拟练习。笔试知识分析部分根据内容分为七个部分:基础知识;程序结构及相关、相近运算;指针、数组和字符串;函数、变量的存储属性及编译预处理;高维数组和指针;构造数据类型;文件。根据实际考试的题型,笔试知识部分包括选择题和填空题两类,每部分有大量的习题。上机试题分析则包括改错题和编程题两类。

如果说笔试知识部分集中的是语法知识的学习,上机试题分析部分则充分为考生介绍了程序的调试技术和调试方法,此为这部分试题分析的宗旨。灵活、熟练地运用这些技术,将使上机调试程序过程中的诸多问题迎刃而解,也使考生不再为上机考试所困扰,并为今后学习编写 C 语言应用软件打下坚实的基础。

在学习本书的知识分析内容之后,我们为考生准备了四套完整的模拟试题(包括两套最近的等级考试试题),以帮助考生进行自我测试。全书所有练

习和模拟试题都配有参考答案。

本书中的每道典型试题分析均列出相关知识点,为了避免重复,一些知识点只列出了名目,它们的展开讲解可以从前面的试题分析中找到。所有出现的特殊的符号也都在首次出现的试题分析中做了说明。

多年来,本书的作者一直从事 C 语言的教学工作,并用 C 语言设计了大量的应用软件。本书是作者根据国家等级考试大纲(二级),结合自己的教学经验和应用体会编写而成。本书不仅可以作为等级考试准备用书,相信也是参加培训和系统学习 C 语言程序设计课程的一本好的参考书。全书的所有完整程序代码在 Turbo C 2.0 下调试通过。

在本书的编写过程中,江文周同志协助审阅了本书的内容,做了一些删改工作,提供了部分资料,同时也得到了沈阳工业大学计算机学院的张荣存院长和陈克宇、常艳等诸多同志的关心和帮助,在此,作者表示深深的谢意,感谢所有为本书的出版给予过支持的人们。

编 者

1998 年 3 月

目 录

出版说明

前言

第一部分 应试指南	1
(一)了解考试大纲与知识准备	1
(二)了解试题与掌握解题方法	1
第二部分 笔试知识重点难点分析	4
一、基础知识	4
(一)选择题	4
(二)填空题	24
习题一	25
二、程序结构及相关、相近运算	29
(一)选择题	29
(二)填空题	49
习题二	53
三、指针、数组和字符串	62
(一)选择题	62
(二)填空题	79
习题三	80
四、函数、变量的存储属性及编译预处理	85
(一)选择题	85
(二)填空题	104
习题四	118
五、高维数组和指针	127
(一)填空题	127
(二)填空题	143
习题五	149
六、构造数据类型	154
(一)选择题	154
(二)填空题	167
习题六	177

七、文件	182
(一)填空题	182
(二)填空题	189
习题七	193
习题参考答案	197
第三部分 上机试题重点难点分析	205
一、程序调试技术	205
(一)启动并设置好 TC 的集成化环境	205
(二)程序的编译和连接	207
(三)跟踪调试	209
(四)集成化环境中的热键	211
(五)简单程序的调试举例	211
二、改错题分析	215
(一)一般性说明	215
(二)试题分析	217
三、编程题分析	229
(一)一般性声明	229
(二)试题分析	230
四、模拟练习	246
模拟练习参考答案	258
第四部分 综合模拟试题	263
模拟试题一	263
模拟试题二	271
模拟试题三	281
模拟试题参考答案	291
第五部分 附录	293
附录 1 国家二级等级考试大纲——C 语言程序设计考试要求	293
附录 2 全国计算机等级考试二级笔试试题(1998 年 4 月)	295
附录 3 C 语言常见错误浅析	304
附录 4 常用字符与 ASCII 码对照表	313
附录 5 运算符的优先级与结合性	314
附录 6 C 语言关键字	315
附录 7 C 库函数浏览	316

第一部分 应试指南

(一) 了解考试大纲与知识准备

在考生准备参加等级考试之前,应该认真了解等级考试大纲。从内容上看,考试大纲中对考试的范围和深度都作了具体的规定。详细了解这些内容可以抓住重点,且不至于遗漏应该了解的知识点。

应该说,考试大纲中知识的涵盖面比较广,覆盖了 C 语言的所有标准语法内容,但侧重于掌握基本应用,对于计算机系统内部和对外设的操作则不涉及或极少涉及。尽管 C 语言本身更新的速度比较快,但考试大纲中主要是以传统语法为主,这样,对于考生回答问题也就没有特殊限制。

众所周知,不同的 C 语言版本在内容和提供的软件支持上有一定差异,考生应注意了解其一般性,而不是某一个 C 语言版本的特殊性。

鉴于 C 语言的灵活多样,我们认为,考试大纲仅是一种指导上的标准,并不能规定到某些实际的细节。例如,C 语言没有输入和输出语句,对于这部分,大纲要求掌握“数据的输入与输出,输入与输出函数调用”,但究竟掌握哪些输入和输出函数,又如对于一个库函数象 `printf`,究竟了解到什么程度,这些都不是大纲所能详细介绍的。这样的问题只能是以常用为准。在进行考试准备时,每一部分内容应该根据一般的教科书认真总结,作出较细致的内容提纲,并参考往年的试题进行校正和补充,以达到全面而又深入了解的目的。

(二) 了解试题与掌握解题方法

试题是对大纲要求的全面反映,只有充分了解试题才能掌握一定的规律、积累经验,达到举一反三的效果。

试题的种类

国家二级考试的 C 语言总体上分笔试试题和上机试题两类。其中笔试试题包括选择题和填空题两种,上机试题包括改错和完善程序(编程)两种。

解题方法

无论回答什么类型的问题,都要求对考试的知识有透彻的了解,而计算机语言又是一种实践性极强的课程,因此,考生必须多注意上机训练,在实践中理解和体会程序设计的内涵并将其融会贯通。实际上机编程是掌握编程语言并学会程序设计的最佳途径。这里仅就一些一般性的答题方法给予介绍或提出建议。

1. 笔试选择题

这类题目每题包括四个选择项,即 A), B), C), D),但只有一个是应该选择的答案。

此类题目的回答可以先剔除其中含有语法或其他错误的选项,然后再进行正向或逆向(排除)判定。

【例 1】若 x 是整型变量, pb 是基类型为整型的指针变量,则正确的赋值表达式是()。

- A) $pb = \&x$ B) $pb = x$ C) $*pb = \&x$ D) $*pb = *x$

首先,选项 B)中将一个整数 x 赋值给指针变量,是非法的,选项 D)中的 $*x$ 是间接应用变量的值,只有 x 是地址才可以这样做,可见,B)和 D)都是错误的。

再者,选项 C)中的 $*pb$ 相当于一个普通的整型变量,而 $\&x$ 是地址,可见,互相间的赋值也不符合语法要求。

总之,只有 A)可能是正确的选择。

【例 2】下面函数的功能是()。

```
sss(s, t)
char *s, *t;
{ while(( *s) && ( *t) && ( *t++ == *s++ ));
  return *s - *t;
}
```

- A) 求字符串长度 B) 比较两个字符串的大小
C) 将字符串 s 复制到字符串 t 中 D) 将字符串接续到字符串 t 中

首先,计算一个字符串的长度只要一个循环即可,不会牵涉到两个字符串同时循环,可见 A)是不太现实的。再考虑一下 C)和 D),选项 C)中的主要操作是“复制”,选项 D)中的主要操作是连接,它们都离不开赋值运算,但原程序中只有比较运算而无赋值运算,可以肯定,C)和 D)的功能是不可能由上述代码实现。故只能选择 B),且这种考虑方法可能较直接阅读程序更快。不过,若时间充足,应对照程序肯定一下自己的判断。

2. 笔试填空题

此类题目一般要求考生读懂程序代码的含义,根据题目所给出的部分代码和文字要求,计算出表达式的值或程序运行结果,或推断出代码中缺少的部分。回答此类题目时,应注意以下几方面的问题:

(1)抓住典型的代码特征。例如,一个只使用 $==$ 运算的字符串操作函数,其功能不会是复制或连接,而很可能是字符串比较。

(2)根据其使用的算法,迅速勾画出程序的思路,与原题所给出的代码进行比较,找出缺少的部分。

(3)注意观察输入数据的特征,对了解程序的作用有一定帮助。

(4)在填空时,注意同类语句的相似性。

(5)计算表达式或简单程序的输出结果时,应注意其中的特殊语法现象,如运算的优先级别和结合次序等。例如,若要求说明下述语句的输出结果:

```
x = 3, 5;
printf("%d", x);
```

因为 $=$ 与 $,$ 的优先次序关系,输出的结果(即 x 的值)是 3 而不是 5。这方面的问题可

参考本书中的试题分析部分。

3. 上机试题的改错题

此类题目中已经指出了代码所含的错误个数和大致位置(位置是在一个 * * * * found * * * * 之后)。总体上看,应该先纠正运行之前的错误,尤其不能忽视警告类错误,这些错误通常集中在变量未初始化、函数的形式参数与实参数不匹配、循环次数多或少、指针的基类型不匹配、相近运算符用错,指针引用错误等方面。纠正这样的程序需要事先了解 C 语言错误提示的含义,另外,注意与 main 函数中的调用形式搭配起来进行观察。

在程序能够正确编译和连接后,运行中的错误则主要靠认真分析程序来纠正。应注意到以下几个方面的问题:

(1)首先,最好在 C 语言的集成化环境下进行程序运行和调试,其好处是在程序出现意外(如无限循环)时可以用 < Ctrl > + < Break > 键终止。

(2)对于大数据量的程序调试可以临时换成小数据量调试,以节省时间。待程序调通后,再换成大数据量运行。

(3)掌握调试工具。一些没有经过上机训练的考生通常只凭经验阅读程序,企图找到其中的错误,效率不高。为此,建议考生应该熟练掌握 C 语言所提供的调试工具,掌握调试方法,积累调试经验,以提高程序调试速度。

本书以例题的方式介绍了 Turbo C 2.0 集成化环境下调试工具的使用方法和一般的调试过程,请参考“上机试题”部分。

4. 上机试题的编程题

此类程序通常只要求考生在函数 fun 内填加自己的程序代码,实现题目要求的功能。为此,建议考生注意以下几个问题:

(1)认真分析函数原型,根据题目要求和 main 中函数的调用形式明确函数参数的意义,以及函数的返回值。当然,也有某些程序是根据较老的 C 语言编制的,函数前没有指定返回值类型,隐含为整型,但没有实际作用。

(2)了解库函数。不仅要了解大纲中所要求的库函数,应更多地了解 C 语言提供了具备哪些能力的库函数,这会使编程游刃有余。例如,若将一个浮点数按四舍五入保留小数点后两位,尽管可以自己编制一个循环程序来处理,但如果使用 sprintf 函数就会更快;又如,将一个数组中的某一部分移动位置,循环语句也可以实现,但使用 memmove 函数就更简单,等等。

(3)当然,并不是每个函数都需要记住其所有细节,通常多数函数只要知道其功能和名字就可以了。在 C 语言的集成化环境中,< Ctrl > + < F1 > 可以查询光标所在处函数的相关知识。

(4)了解常用算法。考生需要了解一些较常见的简单算法,如二分查找算法、排序法等。

无论是哪一种上机考试题目,在程序调试正确后,应该以正确的数据运行一遍程序。

第二部分 笔试知识重点难点分析

一、基础知识

(一) 选择题

【例题 1-1】()是构成 C 语言程序的基本单位。

- A) 函数 B) 过程 C) 子程序 D) 子例程

相关知识:

·C 语言函数:在 C 程序中,只有函数的概念,任何程序都是由一个或多个函数构成的。

·C 语言语句:一个函数的函数体由若干语句组成,语句又可以分为可执行性质语句和非可执行(说明性质)语句。每个语句(复合语句除外)都必须使用分号结尾。

·主函数 main():对于一个完整的程序来说,必须且只能有一个主函数,即 main 函数。程序从 main 函数的第一条语句开始执行,main 函数中的代码执行完毕则意味着程序执行完毕。

分析:过程、子程序和子例程是与函数接近的概念,在其他语言中使用,不是 C 语言中的概念。

结论:答案为 A)。

【例题 1-2】在下述程序段中,()是错误的程序注释方法(注:本例多个选择)。

A) `# in/* 包含 */clude < stdio.h >`
`void main( )`

`{`
`}`

C) `# include < stdio.h >`
`void main( )`
`{ int x/* 初始化 */= 10;`
`/* 打印 */printf("%d",x);`
`}`

B) `# include < stdio.h >`
`void main/* *//(/* 主函数 */)`

`{`
`}`

D) `# include < stdio.h >`
`void main( )`
`{ int x = 10;`
`printf("%d",x);`
`/* 打印 x/* x = 10 */的值 */`
`}`

相关知识:

·注释:在 C 语言中,注释是以“/*”和“*/”括起来的一部分内容,不形成语句,书写时较为灵活。插入注释的要求是:可以在任何允许插入空格符的地方插入注释。此外,C

语言的注释不能嵌套,即一对“/*”和“*/”之间可以包含任何字符,但不能再包含“/*”或“*/”。

分析:A)中的错误是将一个单词或名字拆开插入注释,非法。D)中的注释是嵌套的,也不正确。

结论:答案为 A),D)。

注意:C语言中的一个名字不能拆开,否则插入的注释就成了名字的一部分,但又属于非法字符。类似的运算符也不能拆开,如 $\geq$ 不能写成 $\geq$ 等等。

【例题 1-3】()是 C 语言提供的合法的数据类型关键字。

A) Float B) signed C) integer D) Char

FILE

相关知识:

·关键字:C语言有 32 个关键字,应了解其含义,并注意所有关键字都是小写的英文单词。有关 C 语言的关键字可参见附录 6。

·标识符:C语言的标识符区分大小写字母。

·与整型有关的关键字:C语言中的基本整数类型关键字是 int。除此之外,short, long, signed 和 unsigned 既可以用作 int, char 等的前缀,也可以单独用于表示一种整数类型,其含义分别为短整型、长整型、有符号整型和无符号整型,与 short int, long int, signed int, unsigned int 相同。在微机上的 C 语言中,short 和 signed 基本不用。

此外,程序员可以将基本数据类型(如 int)、长度(如 long)和符号(如 signed)组合成一个类型描述符,如 signed long int,此时,单词的书写次序是任意的,如 int signed long 或 long int signed 等。

结论:答案为 B)。

【例题 1-4】不合法的常量是()。

A) '\2' B) ''' C) '\"' D) "\483"

相关知识:

·字符常量:C语言字符常量有多种写法。首先,C语言中的转义字符有三类。

(1)特殊的转义字符,见表 1-1。

表 1-1

转义字符表

字 符	作 用	字 符	作 用
\a	响铃	\t	制表符,横向跳格
\b	退格	\v	竖向跳格
\f	走纸换页	\\	反斜线字符“\”
\n	换行	\'	单引号字符
\r	回车	\"	双引号字符

将表中的任何一个转义字符用单引号括上都表示某个特定的 ASCII 码字符,其作用是固定的。但在标准输出时,有些字符的实际意义会与表中所述有所差别。例如,微机中输出字符'\v'将显示字符“␣”而不起控制作用;输出字符'\r'将使光标定位在本输出行的行首;输出字符'\f'将显示字符“♀”而不起控制作用;书写在字符串常量中的字符“'”

可以直接书写,也可以写成转义字符“\””;书写单个字符常量“'”可以写成“'”或转义字符“\””等等。

(2)由八进制序列组成的转义字符

此类字符的标准写法是'\ddd',其中的d表示一个八进制数码。如果不足3位可在前面补0或不补,如'\004','\04'和'\4'都代表同一个字符。在试题中,考生应注意其中的任何一个d不能是数字8以上(包括8在内)的数码,如'\284'之类的错误字符,C语言的编译器不能查出此类错误。

(3)由十六进制序列组成的转义字符

此类字符的标准写法是'\xhh',其中的h表示一个十六进制数码。如果不足两位可在前面补0或不补,如'\x04'和'\x4'代表同一个字符。

其次,普通的字符只是两个单引号括起来的一个字符,如'A'。

在特殊情况下,一个整数也可以充当字符使用。

·字符串常量:字符串常量是由双引号括起来的一串字符,每个字符串常量的最后是结束符'\0'。计算字符串常量时'\0'不包括在内。在特殊情况下,可以一个字符也不包括,形成的字符串即是“”,称其为“空字符串”或“空串”。此字符串仅占用一个字节,存放的是字符串结束符,即'\0'。若用strlen(“)函数测试,此值为0。

·字符常量与字符串常量的区别:读者需要注意理解字符串常量与字符常量的区别。例如,尽管字符串常量"D"和字符常量'D'写法相近,但它们占用的存储空间和内容却有差别:



并且,字符串与字符的用法也完全不同,下面的输出都是错误的:

```
printf("%s",'D');
```

```
printf("%c","D");
```

分析:由上所述,B)和D)是字符串常量,A)和C)是字符常量。C语言允许空字符串,B)是正确的。选项D)中字符串"483"是值得注意的,因为初看起来它是一个八进制转义序列,但因为8超出了八进制范围,系统自动将其识别为由三个字符'\4','8'和'3'组成,在输出时显示为“◆83”。

选项A)是一个普通的八进制转义字符,ASCII码值是2。

选项C)仅由两个连续的单引号组成,不是正确的字符常量。不过,Turbo C 2.0 之前的C语言版本不能检查出此类错误。

结论:答案为C)。

注意:如果在转义字符中使用了超出进制范围的数码,很可能是个错误。例如,本例中的"\483"可以被正确识别,因为它是一个字符串常量,但'\483'却是错误的常量写法,因为它已经不是由一个字符组成了。

【例题 1-5】在以下各组标识符中,合法的用户标识符是____(1)____、____(2)____、____(3)____。

- | | | | |
|----------------------------|-------------------------|-------------------|---------------------|
| (1) A) <u>001</u>
Int | B) table_1
t * .1 | C) 0_t
W10 | D) k%
point |
| (2) A) Fast_
Fast + Big | B) void_
abs | ✓ C) pb1
fabs | D) < book >
beep |
| (3) A) xy_
变量 1 | B) longdouble
signed | C) * p
history | ✗ D) CHAR
Flaot |

相关知识:

·标识符:C语言规定,用户标识符只能由字母(大写或小写)、数字(0~9)和下划线(_)组成,不能由数字字符开头,大小写字母是不同的字符,且不能使用C语言的32个关键字作为用户标识符。关于标识符的长度,则至多可使用31个字符(内部名,外部名至多8个字符,但可能随机器及语言版本的不同而有差异)。

分析:分析标识符是否合法,只要按照规定的条款去比较即可。

第一组中,A)是正确的,尽管Int与关键字int接近,但C语言区分字母的大小写,二者是不同的;B)中的t * .1含有非法字符“*”和“.”;C)中的0_t是由数字字符开头的;而D)中的k%含有非法字符“%”。

第二组中,Fast + Big含有非法字符“+”;void为关键字;< book >中含有非法字符“<”和“>”。

第三组中,“变量 1”中的汉字是非法字符;signed是关键字;* p中的“*”是非法字符。

结论:答案是(1)A) (2)C) (3)D)。

注意:事实上,第二组的选项C)中将fabs用作了用户标识符,尽管从语法角度看是允许的,但因其与库函数同名,将其用作其它标识符可能引起不良后果,应尽量避免。

【例题 1-6】属于合法的C语言长整型常量的是()。

- A) 5876273 B) 0L C) 2E10 D) (long)5876273

相关知识:

·普通整型常量:是占用两个字节存储的常量,写法有三种:

十进制表示形式,如25, -3等;

八进制表示形式,以0开头,如063等;

十六进制表示形式,以0x开头,如0xa3等。

普通整型常量的使用范围在-32768和32767之间。

·长整型常量:在普通整常数之后加后缀“L”或“l”所形成的整型常量,存储时占4个字节,如12L。此类数据的使用范围在-2147483648与2147483647之间。

并非只有大的整数可以使用长数据。例如,10和10L都是正确的整型常量,只是后者占用更多的存储空间。

整数是以定点方式存储的。

·浮点型常量:有十进制表示(如-2.5)和指数表示(如1.0E3和-0.05E2)两种形式,占用8个字节,属双精度类型的浮点数,取值在 10^{-308} ~ 10^{308} 之间。

C语言中的十进制小数可以省略小数点前面的整数部分或后面的小数部分,如2.和

.3都是正确的写法。

应该注意指数形式浮点常量的写法。C 语言中的指数形式常量的标准格式是 fEm 或 fem, 其中的 f 为一般形式的十进制整数或浮点数, m 必须是整数, 且要求 f 和 m 都不能省略。

分析: 选项 A) 中的整数超出了普通整数范围且而无后缀 L 或 l; 选项 C) 中的数是浮点型常量; 而 D) 则是一个表达式, 不是常量。

结论: 答案为 B)。

说明:

(1) 在大多数 C 语言版本中, 将选项 A) 中的整数直接赋给一个长整型变量或者输出还是允许的。例如, 下述程序的输出结果并不会产生任何问题:

```
#include <stdio.h>
void main( )
{ long x = 5876273;          /* 若将 long 改为 int 将出错 */
  printf("\n%ld, %ld", x, 5876273);
}
```

尽管如此, 在回答问题和编程时, 应尽量按语法要求进行操作。

(2) 在 C 语言中, 以特殊字符做后缀的常量是比较常见的, 主要包括如: 2L(长整型常量)、3.14F(单精度浮点型常量)、20U(无符号整型常量)、3.24L(长浮点型常量)等, 后缀中的字母大小写均可。

【例题 1-7】在通常微机上的 C 语言中, int 类型数据占用 2(1) 个字节存储; unsigned int 类型数据占用 2(2) 个字节存储; short 类型数据占用 2(3) 个字节存储; long 类型数据占用 4(4) 个字节存储。

A) 1

B) 2

C) 4

D) 8

相关知识:

·整型数据的存储: 尽管在不同的系统中, 数据的存储长度会有差异, 但一般数据有无符号不影响存储长度, short 类型的数据不会超过普通类型数据的长度, 而 long 类型的数据则是普通类型数据长度的 2 倍。

·浮点数据的存储: float 是单精度浮点类型, 其对应常量如 3.14F, 存储长度为 4 字节。考生应注意一般的浮点常量如 3.14 和 0.0314E2 总是双精度浮点常量, 与 double 类型数据相对应, 占 8 字节。特别地, 长浮点数 long double 类型的数据在微机中(如 Turbo C)占 10 字节存储(而不是一般书籍中讲述的 16 字节或 12 字节), 其对应的常量形如 3.14L。

·字符数据存储: 任何一种字符型的数据总是占用一个字节存储, 这与有无符号无关。

分析: 由于数据占用空间与其有无符号无关, 故 int 和 unsigned int 类型数据都占两个字节, 微机版本中的 signed 类型数据也占两个字节。long 类型数据占用四个字节。

结论: 答案为(1)B) (2)B) (3)B) (4)C)。

【例题 1-8】有关下述语句输出的判断中, 正确的是()。

```
int x = 0xaffbc;
printf("%x", x);
```


- A) 赋值非法 B) 输出值不确定 C) 输出值为 `affb` D) 输出值为 `ffbc`

0xa	0xffbc
-----	--------

高 低

相关知识:

·数据类型的隐式转换:在不同类型的数据进行运算时,系统将对其中的某个或某几个进行类型转换,使其一致。转换的基本规则主要包括:

(1)算术运算:类型按升格处理,即由定点数转换为浮点数,短存储数据转换为长存储数据,有符号数据转换为无符号数据。

0xa	0xffbc
-----	--------

高 低

(2)赋值运算:赋值号右边的表达式的数据类型转换为左边变量的数据类型。

(3)输出:按函数的参数和格式说明进行转换。

·长整数与普通整数间的转换:在 C 语言中,如果一个整型常量的值在 $-32768 \sim +32767$ 之间,可以有或没有后缀 `L`(或 `l`),并可以将其赋给任何一种整型变量。一旦超出此范围,值在 $-2147483648 \sim +2147483647$ 之间,书写时最好加上后缀,只能将其赋给一个 `long int` 类型变量。

当一个大的长整数赋给普通整型变量时,C 语言将会“截断”多余部分。此时,两个低字节的值被保留下来,而高字节的值“丢失”。

分析:首先,一般来说,对于不同类型数据如整数、浮点数和字符进行混合运算或赋值时,不管值的大小,C 语言中不会出现语法错误,但值可能不正常。故 A) 不正确。

其次,定点数之间、浮点数之间的输出转换只能在值的大小、范围上有取舍,但结果应该是定值,B) 错误。

最后,对此例来说, `0xaffbc` 显然已超出了普通整数所能容纳的范围。`ffbc` 组成两个字节的普通整数,高位的 `a` 丢失。最后输出的是 `ffbc`。而选项 C) 并不是取高字节的结果。

结论:答案为 D)。

【例题 1-9】有关下述语句输出的判断中,正确的是()。

```
char x = '\x0-1';
printf("%d", x);
```

- A) 赋值非法 B) 输出值不确定 C) 输出值为 -32 D) 输出值为 224

相关知识:

·整数与字符的转换:在 C 语言中,整数与字符型数不需特殊转换即可混合使用,因此,一些程序员将字符型变量当作“短的整型”变量使用,以节约空间。

当一个整数转换为字符时,其值是整数的低字节值。而在字符转换成整数时,值不变。

·补码存储:事实上,定点数都是以补码形式存储的。无论从哪个角度看,这都是一个应该理解的问题。