



普通高等教育“十二五”重点规划教材·计算机系列
中国科学院教材建设专家委员会“十二五”规划教材

王晓云 陈业纲 主 编

C语言程序设计

C YUYAN CHENGXU SHEJI



科学出版社

普通高等教育“十二五”重点规划教材·计算机系列
中国科学院教材建设专家委员会“十二五”规划教材

C 语言程序设计

王晓云 陈业纲 主 编

陈 曦 胡志竹 贺代春 副主编

科 学 出 版 社

北 京

内 容 简 介

本书以高等学校计算机专业本科生“高级程序设计”课程教学大纲为骨架,融合等级考试大纲的要求,注重教材的可读性和适用性。全书内容包括C语言程序设计基础、常见数据类型、运算符与表达式、基本输入输出、顺序结构、分支(选择)结构、循环结构、数组、函数、指针、预处理命令、结构体、链表、文件等。

本书附有大量的图表和程序,引导读者自主解决问题,注重提高读者解决实际问题的能力。

本书可作为高等院校计算机专业教材和计算机等级考试用书,也可供广大工程技术人员和自学者学习参考。

图书在版编目(CIP)数据

C语言程序设计/王晓云,陈业纲主编. —北京:科学出版社,2012
(普通高等教育“十二五”重点规划教材·计算机系列)

ISBN 978-7-03-034462-5

I. ①C… II. ①王…②陈… III. ①C语言—程序设计—高等学校—教材 IV. ①TP312

中国版本图书馆CIP数据核字(2012)第107813号

策划:隽青龙

责任编辑:隽青龙 / 责任校对:马英菊

责任印制:吕春珉 / 封面设计:北京子时文化设计公司

科学出版社 出版

北京东黄城根北街16号

邮政编码:100717

<http://www.sciencep.com>

陈海印刷有限责任公司 印刷

科学出版社发行 各地新华书店经销

*

2012年7月第 一 版 开本:787×1092 1/16

2013年1月第二次印刷 印张:15 1/4

字数:345 000

定价:30.00 元

(如有印装质量问题,我社负责调换<路通>)

销售部电话 010-62142126 编辑部电话 010-62135517-2037

版权所有,侵权必究

举报电话:010-64030229; 010-64034315; 13501151303

前 言

C 语言是广泛流行的一种结构化程序设计语言，兼有高级语言和低级语言的功能，不仅可用于操作系统的设计，也可用于各类应用程序以及工业控制程序的设计。目前流行的面向对象程序设计语言，如 C++、Java、C#等都是在 C 语言的基础上发展派生而来的。因此，C 语言得到了广泛的认可和重视。另外，C 语言还蕴含了程序设计的基本思想，囊括了程序设计的基本概念，各类高等院校的很多专业都将 C 语言作为学习计算机的入门语言和重要的基础课程。

由于历史和现实原因，目前国内的大部分 C 语言教材都以 C 语言的语法规则为主线编写，教学重点是 C 语言语法规则，教学中过分注重 C 语言语句、语法和一些细节的讲解；对学生的考核，特别是等级考试和标准化考试也是偏重于语法，课程的教学基本上是以高级语言自身的体系为脉络展开的，学生花了很大精力去学习语法，但还是不知道如何编写一个比较简单的程序，不知怎样通过编程去解决实际问题。面对这些问题，编者在多年教学经验的基础上，采用 ANSI C99 标准，从 C 语言程序设计的基本原理及程序设计的基本思想出发，贯穿“基础—应用—专题”这一主线，紧扣基础，循序渐进，面向应用。本书具有以下特点。

1) 注重实践，每节的内容都用于解决实际问题。本书强调程序设计过程就是求解问题过程，注重问题求解能力的培养，突出“程序设计为主，语法规则为辅”这样一种教学理念。融合并加强了等级考试的上机要求；实验部分采用了基本实验和扩展实验，同时引入软件工程项目的概念。

2) “图灵”奖得主 Niklaus Wirth 教授曾提出一个著名论断：程序=算法+数据结构。这说明程序的核心是算法，算法的本质是处理数据，算法与数据不可分离。全书大部分内容都围绕算法和数据组织方式（数据结构）组织内容，重点介绍经典的排序与查找算法及其实现。

3) 在程序设计（算法设计）方面，编者严格按照“自顶向下、逐步求精”的结构化程序设计原则进行例题讲解，摒弃了大多数 C 语言教材中“提出问题—给出源程序—解释程序”的教学思路；而是根据学生的学习规律，采用“提出问题—分析问题—设计算法框架—算法细化—程序实现—程序测试”这样一个步骤来组织内容。这样的组织方式可以更好地培养学生的程序设计能力。

4) 本书增加了一定量程序阅读训练方面的内容，书中精选了大量不同类型的经典实例程序和近几年全国计算机等级考试真题作为习题和例题，实例丰富，难易得当，注重实际应用，便于读者学习和自测；重点放在编制程序和调试程序上，着力培养读者运用所学知识综合解决实际问题的能力。

5) 本书将指针的知识巧妙融入到各个章节，降低了学习的难度。每一章节的内容都是在克服前一章节的局限性和继承其优点之后引入的，并且从需要出发，用实例说明问题，使读者具有身临其境的创新体验，从而自主地分析和解决一些疑难问题。同时对每章的关键知识点和要求掌握的知识进行了详细的说明；富有大量的图标和程序，使读

者能正确、直观地理解问题；使结构、算法、代码、运行过程和结果同时展现，数据自由输入，运行过程步步可见。

同时还编写了与本书内容配套的实验指导教材，每个实验包括精心设计的编程和调试实例，以及与计算机等级考试对应的题型，同时还对学有余力的学生专门设计了扩展实验。通过“模仿—修改—设计”的上机实践过程，在循序渐进中逐步熟悉编程环境，掌握常见的算法以及设计技巧。

计算机科学与技术在不断f展，计算机教学的研究和改革也从未停顿，希望在从事计算机基础教育的各位同仁的不懈努力下，不断提高我国高等学校高级程序设计语言课程的教学质量和水平。

本书由王晓云、陈业纲担任主编，由陈曦、胡志竹、贺代春担任副主编，参加编写和审校工作的还有张金山。

由于编者水平有限，书中难免存在谬误之处，敬请读者指正。为方便教师的教学工作和读者的学习，本书有配套的源程序代码、习题答案和电子教案，可通过邮箱 wxyz93@163.com，或通过出版社与编者联系获取。

编 者

2012 年 4 月

目 录

第 1 章 C 语言概述	1
1.1 程序及程序设计的基础知识	1
1.1.1 语言的发展	1
1.1.2 数制的基本概念	2
1.2 C 语言概述	4
1.2.1 C 语言的特点	4
1.2.2 C 语言标识符与关键字	5
1.2.3 C 程序的基本结构	6
1.2.4 C 语言的语句	7
1.3 C 程序的开发过程	8
1.4 应用	9
1.5 习题	10
第 2 章 数据类型、运算符、表达式与基本输入输出	12
2.1 C 语言的数据类型	12
2.2 常量与变量	12
2.2.1 基本常量和符号常量	12
2.2.2 变量	13
2.3 整型数据	14
2.3.1 整型常量	14
2.3.2 整型变量	15
2.4 实型数据	16
2.4.1 实型常量	16
2.4.2 实型变量	17
2.5 字符型数据	18
2.5.1 字符型常量	18
2.5.2 字符型变量	19
2.5.3 字符串常量	20
2.6 基本输入输出函数	21
2.6.1 格式输出函数 printf	21
2.6.2 格式输入函数 scanf	23
2.6.3 字符输出函数 putchar	26
2.6.4 字符输入函数 getchar	27
2.7 运算符和表达式	27
2.7.1 算术运算符和算术表达式	28

2.7.2	赋值运算符和赋值表达式	30
2.7.3	不同类型数据间的混合运算	32
2.7.4	关系运算符和关系表达式	33
2.7.5	逻辑运算符和逻辑表达式	34
2.7.6	逗号运算符和逗号表达式	36
2.7.7	求字节数运算符	36
2.7.8	位运算符和位运算	37
2.8	指针的概念与运算	39
2.8.1	地址及指针	39
2.8.2	指针变量的定义、赋值和应用	40
2.9	习题	43
第 3 章	顺序程序设计	47
3.1	算法	47
3.2	结构化程序设计方法	51
3.3	顺序结构程序设计应用	52
3.4	习题	53
第 4 章	分支结构程序设计	58
4.1	if 语句	58
4.1.1	if 语句的形式	58
4.1.2	if 语句的嵌套	62
4.2	条件运算符和条件表达式	63
4.3	switch 语句	65
4.3.1	在 switch 中不使用 break 语句	65
4.3.2	在 switch 语句的每个语句段中都使用 break 语句	67
4.3.3	在 switch 的某些语句段中使用 break 语句	67
4.4	分支（选择）结构程序应用	68
4.5	习题	72
第 5 章	循环结构程序设计	76
5.1	while 语句	76
5.2	do...while 语句	78
5.3	for 语句	80
5.4	几种循环语句的比较	82
5.5	continue 语句和 break 语句	84
5.5.1	continue 语句	84
5.5.2	break 语句	84

5.6	循环语句的嵌套	86
5.7	循环结构程序应用	88
5.8	习题	91
第 6 章	数组与指针	95
6.1	一维数组的定义和引用	95
6.1.1	一维数组的定义	95
6.1.2	一维数组元素的引用	96
6.1.3	一维数组的初始化	97
6.1.4	一维数组的应用	98
6.2	一维字符数组及指针	101
6.2.1	一维数组的指针和指向一维数组的指针变量	101
6.2.2	指针运算	103
6.2.3	通过指针引用数组元素	104
6.2.4	一维字符数组及字符指针变量的定义	106
6.2.5	一维字符数组及字符指针的初始化	106
6.2.6	字符数组的输入和输出	107
6.2.7	常用字符串处理函数	109
6.2.8	字符数组应用	111
6.3	二维数组	112
6.3.1	二维数组的定义	112
6.3.2	二维数组元素的引用	112
6.3.3	二维数组的初始化	112
6.3.4	二维数组的应用	114
6.3.5	多维数组的指针	116
6.4	指针数组和指向数组的指针	117
6.5	指向指针的指针变量	120
6.6	习题	121
第 7 章	函数与变量	127
7.1	函数	127
7.1.1	函数的定义	128
7.1.2	函数的引用	129
7.1.3	函数调用	131
7.1.4	返回值	133
7.2	函数的嵌套调用	133
7.3	函数的递归调用	136
7.4	指针作为函数的参数	138
7.4.1	指针变量作为函数参数	138

7.4.2 数组元素作为函数参数	140
7.4.3 数组名作为函数参数	141
7.5 变量的存储类型及分类	143
7.6 内部函数和外部函数	146
7.6.1 内部函数	146
7.6.2 外部函数	146
7.7 函数的指针和指向函数的指针变量	147
7.8 返回指针值的函数	150
7.9 局部变量和全局变量	152
7.9.1 局部变量	152
7.9.2 局部变量的存储类型	153
7.9.3 全局变量	154
7.9.4 全局变量的存储类型	155
7.10 函数的应用	156
7.11 习题	161
第 8 章 编译与预处理	167
8.1 宏定义	167
8.1.1 无参宏定义	167
8.1.2 带参宏定义	170
8.2 文件包含	172
8.3 条件编译	173
8.4 枚举类型	175
8.5 用 typedef 定义类型	178
8.6 习题	179
第 9 章 结构体与其他数据类型	184
9.1 结构体变量	184
9.1.1 结构体类型的声明	184
9.1.2 结构体类型变量的定义和初始化	185
9.1.3 结构体变量的引用	188
9.2 结构体数组	190
9.3 指向结构体类型数据的指针	193
9.4 共用体	196
9.4.1 共用体类型的定义	196
9.4.2 共用体类型变量的定义及初始化	196
9.4.3 共用体变量的引用	197

9.5 链表	198
9.5.1 单向链表的数据结构	199
9.5.2 内存空间的动态分配和释放	199
9.5.3 单向链表的基本操作	201
9.6 习题	205
第10章 文件	209
10.1 C 文件概述	209
10.2 文件的打开与关闭	210
10.2.1 文件指针	210
10.2.2 文件的打开	210
10.2.3 文件的关闭	211
10.3 常用的文件读写函数	212
10.3.1 字符读写函数 fgetc 和 fputc	212
10.3.2 字符串读写函数 fgets 和 fputs	214
10.3.3 数据块读写函数 fread 和 fwrite	216
10.3.4 格式化读写函数 fscanf 和 fprintf	218
10.4 文件的定位	219
10.4.1 文件定位	219
10.4.2 文件的随机读写	220
10.5 错误检验	221
10.6 习题	221
附录	226
附录1 常见字符的 ASCII 码	226
附录2 常用的标准库函数	226
附录3 常见出错信息	229
参考文献	233

第 1 章 C 语言概述

本章要点

- C 语言程序结构和特点
- C 语言程序的标识符与关键字
- C 语言程序的编辑及运行

1.1 程序及程序设计的基础知识

1.1.1 语言的发展

1946 年,第一台计算机诞生时的情形与原始社会的出现类似。当时的“程序员”可谓精通电子技术的专家,他们通过设计复杂的电路板完成各种计算工作,而计算机也仅仅用于满足军事的需要。“程序员”作为计算机的主人,他们操纵所有的硬件资源,指挥那个笨拙的庞然大物完成种种不可思议的任务。

输入/输出设备的发展促使了语言的诞生。穿孔纸带的产生使程序员不必具有太多的电子技术知识,只需要懂得计算机的语言就能与计算机交流。计算机所能“听”懂的语言是由 0 和 1 组成的一长串数字,这就是机器语言。程序员在纸带上打孔,向计算机发号施令。新型的输入/输出设备,如磁带、键盘等的出现使穿孔纸带被丢进了垃圾桶。但程序员们依然平等、自由。随着特权阶级的出现标志着原始社会行将崩溃,贵族就像操作系统,掌控着部分特权,程序员们再不能像以前那样操纵所有的硬件资源了,相当多的资源由操作系统接管。程序员们失去了部分自由,却换来了开发效率的提高。

由于机器语言十分难记,人们逐渐使用助记符号代替 0、1 字串,如 ADD、SUB、MOVE 等。这种用助记符表示计算机指令的语言称为汇编语言,汇编语言只是对机器语言的简单替换,但仅仅这样仍算是一个巨大的进步,汇编语言的出现大大提高了程序员的开发效率。

计算机无法执行汇编源程序,编译程序负责将汇编语句翻译成机器语言。早期的汇编语句与机器语言语句之间是一一对应的。但这种机械的平均主义显然成了社会发展的障碍。能者多劳,一句宏汇编语句可能抵得上多句机器语言。这种抽象语言提高了程序员的工作效率,却使他们与机器越来越远。

为获得更高的效率,程序员们期待更加抽象的语言,可以屏蔽掉所有的机器细节。

FORTRAN 的出现标志着一个新的时代的到来,人们从此得以从复杂的机器细节中抽身,再不用关注于内存、寄存器等细节了,这是计算机历史上的一次伟大的革命。

高级语言出现伊始,goto 由于灵活和高效成了程序员们竞相追逐的工具,有关 goto 的种种技巧在程序员之间传播。goto 甚至成了衡量程序员水平的标尺。

程序员们虽然失去了控制硬件资源的自由,却在高级语言的使用上不受任何限制。他们可以使用任意的风格,诡异的技巧,写出除他们之外谁也看不懂的程序。

当大多数人还沉浸在 goto 带给他们的自由与荣耀时,天才的 Edsger Dijkstra 却敏锐地发现了 goto 所带来的种种问题。这位荷兰科学家发表了他的著名论文《goto 有害论》,轰动了整个计算机世界。他指出, goto 是导致程序复杂、混乱、难以理解的罪魁祸首,它还使效率难以度量,程序难以维护。

通过众人的努力,人们总结出一套行之有效的程序设计方法,称为结构化程序设计。程序员不能再随心所欲地编码了, goto 成了程序员们避之不及的“毒药”。

C 语言的出现标志着结构化程序设计的到来,它是由 UNIX 的研制者丹尼斯·里奇 (Dennis Ritchie) 在 1970 年由肯·汤普逊 (Ken Thompson) 所研制出的 B 语言的基础上发展和完善起来的。目前, C 语言编译器普遍存在于各种不同的操作系统中,如 UNIX、MS-DOS、Microsoft Windows 及 Linux 等。C 语言的设计影响了许多后来的编程语言,如 C++、Objective-C、Java、C# 等。

后来在 20 世纪 80 年代,为了避免各开发厂商用的 C 语言语法产生差异,由美国国家标准局 (ANSI) 为 C 语言制定了一套完整的国际标准语法,称为 ANSI C,现在以 ANSI C99 作为 C 语言的标准。

1.1.2 数制的基本概念

要学好 C 程序设计就必须先了解 0 和 1,这就涉及了进制数。

1. R 进制计数制

从十进制计数制的分析得出,任意 R 进制计数制同样有基数 N 和 R^i 按权展开的表示式。R 可以是任意正整数,如二进制中的 R 为 2。

(1) 基数

一个计数所包含的数字符号的个数称为该数的基,用 R 表示。例如,对二进制来说,任意一个二进制数可以用 0, 1 两个数字符号表示,其基数 R 为 2。

(2) 位值 (权)

任何一个 R 进制数都是由一串数码表示的,其中每一位数码所表示的实际值的大小,除数码本身的数值外,还与它所处的位置有关,由位置决定的值就称为位值 (或位权)。位值用基数 R 的 i 次幂 R^i 表示。假设一个 R 进制数具有 n 位整数, m 位小数,那么其位权为 R^i ,其中 $i=-m \sim (n-1)$ 。

(3) 数值的按权展开

任一 R 进制数的数值都可以表示为各个数码本身的值与其权的乘积之和。例如,二进制数 101.01 的按权展开为 $101.01B = 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 + 0 \times 2^{-1} + 1 \times 2^{-2} = 5.25D$ 。

任意一个具有 n 位整数和 m 位小数的 R 进制数的按权展开为

$$R = d_{n-1} \times R^{n-1} + d_{n-2} \times R^{n-2} + \dots + d_2 \times R^2 + d_1 \times R^1 + d_0 \times R^0 + d_{-1} \times R^{-1} + \dots + d_{-m} \times R^{-m}$$

其中 d_i 为 R 进制的数码。

2. C 语言中的数制

C 语言中的数制如表 1-1 所示。

表 1-1 C 语言中的数制

进位制	计数规则	基数	各位的权	可用数符	修饰字符
二进制	逢 2 进一	2	2^i	0, 1	b
八进制	逢 8 进一	8	8^i	0, 1, ..., 7	o
十进制	逢 10 进一	10	10^i	0, 1, ..., 9	d
十六进制	逢 16 进一	16	16^i	0, 1, ..., 9, A, ..., F	ox

3. 进制之间的转换

(1) R 进制转换为二进制

整数部分：除 R 取余倒排列（除到商为零为止）。

小数部分：乘 R 取整顺排列（如果最后不为零，按要求保留位数）。

例如：把十进制数 25.3125 转换成二进制数，方法如下。

整数部分：

2	25	余数
2	12	1 ← 最低位
2	6	0
2	3	1
2	1	1
	0	1 ← 最高位

所以整数部分 (25) d = (11001) b。

注意：第一位余数是低位，最后一位余数是高位。

小数部分：

	取整
$0.3125 \times 2 = 0.6250$	0
$0.6250 \times 2 = 1.2500$	1
$0.2500 \times 2 = 0.5000$	0
$0.5000 \times 2 = 1.0000$	1

所以，小数部分有 (0.3125) d = (0.0101) b。

综上所述：(25.3125) d = (11001.0101) b

(2) R 进制转换为十进制

按权展开并求和：

$$s = \sum_{i=-m}^{n-1} D_i \times R^i \quad (\text{假设一个 } R \text{ 进制数有 } n \text{ 位整数, } m \text{ 位小数})$$

例如

$$(1AB.C8)_{16} = 1 \times 16^2 + A \times 16^1 + B \times 16^0 + C \times 16^{-1} + 8 \times 16^{-2} = (427.78125)_{10}$$

4. 原码、反码、补码

计算机中所有数据最终都是使用二进制数表达的。一个负数如何用二进制表达？在

计算机中，负数以其正值的补码形式表达。介绍补码之前，首先简述原码和反码。

(1) 原码

一个整数，按照绝对值大小转换成的二进制数，称为原码。最高位为符号位，正数的符号位为 0，负数的符号位为 1。

例如，-5 的原码为 10000000 00000000 00000000 00000101。

(2) 反码

1) 正数的反码=原码。

5 的反码是 00000000 00000000 00000000 00000101。

2) 负数的反码：符号位不变，其余各位取反（1 变 0，0 变 1）。

-5 的反码是 11111111 11111111 11111111 1111010。

(3) 补码

1) 正数的补码=原码。

5 的补码是 00000000 00000000 00000000 00000101。

2) 负数的补码=反码+1。

-5 的反码是 11111111 11111111 11111111 1111011。

5. ASCII 码

C 语言中字符编码是 ASCII 码（美国标准信息交换码），一个字节由 8 位组成，其中最高位是 0，所以 7 位二进制数可表示 128 种状态即 128 个字符。

第 0~32 号及第 127 号（共 34 个）是控制字符或通信专用字符；第 33~126 号（共 94 个）是字符，其中第 48~57 号为 0~9 十个阿拉伯数字，65~90 号为 26 个大写英文字母，97~122 号为 26 个小写英文字母。常见字符的 ASCII 码参见附录 1。

1.2 C 语言概述

C 语言是一种计算机程序设计语言，其既具有高级语言的特点，又具有汇编语言的特点，可以作为系统设计语言编写系统软件，也可以作为应用程序设计语言编写不依赖计算机硬件的应用程序。因此，C 语言的应用范围广泛，不仅仅是在软件开发上，而且各类科研都需要用到，如单片机以及嵌入式系统开发等。

1.2.1 C 语言的特点

1. 优点

1) C 语言简洁紧凑、灵活方便。C 语言包括 32 个关键字，9 条控制语句，书写形式自由，通常用小写字母表示。它把高级语言的基本结构和语句与低级语言的实用性结合起来，C 语言可以像汇编语言一样对硬件进行直接操作。

2) C 语言运算符丰富。C 语言的运算符共有 34 种，C 语言把括号、赋值、强制类型转换等都作为运算符处理，从而使 C 语言的运算类型极其丰富，表达式类型多样化。

灵活使用各种运算符可以实现在其他高级语言中难以实现的运算。

3) C语言数据结构丰富。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。C语言能用于实现各种复杂的数据结构的运算；并引入了指针概念，使程序效率更高；具有强大的图形功能，支持多种显示器和驱动器；且计算功能、逻辑判断功能强大。

4) C语言是结构式语言。C语言是以函数形式提供给用户的，这些函数可方便地调用，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语言语法限制不太严格，程序设计自由度大。C语言语法比较灵活，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，对硬件进行操作。由于C语言允许直接访问物理地址，可以直接对硬件进行操作。

7) C语言生成目标代码质量高，程序执行效率高。C语言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语言具有强大的绘图能力，可移植性好，并具备很强的数据处理能力，因此适于编写系统软件，三维、二维图形和动画，同时也是数值计算的高级语言。

2. 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在数据的安全性上有很大缺陷，这也是C和C++的一大区别。

2) C语言的语法限制不太严格，对变量的类型约束不严格，影响程序的安全性，对数组下标越界不做检查，等等。

1.2.2 C语言标识符与关键字

1. 概念

在C语言中用于标识名称的有效字符序列称为标识符，也就是指用户自定义的变量、函数、常量、符号名、数组名和指针名等。

C语言规定，标识符由字母、数字和下画线组成，第一位只能为字母或下画线。

注意：C语言标识符是区分大小写的，也就是说标识符a和A是不相同的，abc，Abc，ABC，aBc...也是不同的。C语言的不同标准(ANSI C89/C99)，不同编译器对标识符有效字符个数有不同定义，早先许多系统规定前8个字符有效，现在常见的编译器一般都支持32个(或更多)字符。

2. 分类

(1) 关键字

关键字是指被C语言保留的，不能用作其他用途的一些标识符，其在程序中都有专

用的含义。

常见的关键字有 auto、extern、sizeof、break、float、static、case、for、struct、char、goto、switch、continue、int、typedef、const、if、union、default、long、unsigned、do、register void、double、return、volatile、else、short、while、enum、signed。

(2) 预定义标识符

预定义标识符在 C 语言中也有特殊的含义，如库函数 (scanf、printf、sqrt) 和编译预处理命令名 (define、include)。

注意：预定义标识符可以作为“用户定义的标识符”，但此时标识符将失去系统规定的原意。

(3) 用户定义标识符

用户根据程序设计的需要定义的标识符称为用户定义标识符，用户定义标识符一般用于给变量、函数、数组、指针、结构体、共用体等命名，注意不能与关键字相同。

3. 例题

1) 以下选项中合法的标识符是 ()。

- A. 1-1 B. 1—1 C. _11 D. 1—

2) 以下不合法的用户定义标识符是 ()。

- A. Main B. _0 C. _int D. sizeof

1.2.3 C 程序的基本结构

任何一种程序设计语言都具有特定的语法规则和规定的表达方法。一个程序只有严格按照语言规定的语法和表达方式编写，才能保证编写的程序在计算机中能正确地执行，同时也便于阅读和理解。为了解 C 语言的基本程序结构，先介绍几个简单的 C 程序。

例 1.1 在屏幕上输出 “Welcome to C World!”。

```
#include <stdio.h> /* 文件包含命令 */
void main()
{
    printf("Welcome to C world!\n"); /*输出语句*/
}
```

运行结果为



```
ex "D:\1\Debug\1.1.exe"
Welcome to C world!
```

例 1.2 使用 C 程序实现简单计算。

```
#include <stdio.h> /* 文件包含命令 */
int add(int x,int y) /* 求两数的和 */
{
    return x+y;
}
```

```
int mul(int x,int y)           /* 求两数的积 */
{
    int t;
    t=x*y;
    return t;
}
void main()
{
    int a,b,c,d,e;             /* 定义变量 */
    scanf("%d%d%d%d",&a,&b,&c,&d); /* 键盘输入 */
    e=add(a,b)+mul(c,d);       /* 函数调用 */
    printf("%d\n",e);          /*输出语句*/
}
假设输入 1、4、3、5,
```

运行结果为



```
C:\D:\1\Debug\1.2.exe
1 4 3 5
20
```

结合以上示例，可以看出 C 语言程序结构有以下基本特点。

1) C 语言程序是由函数（如 main 函数和 max 函数）组成的，每一个函数完成相对独立的功能，函数是 C 语言程序的基本模块单元。main 是函数名，函数名后面的一对圆括号“()”用于写函数的参数。参数可以有，也可以没有（本程序没有参数），但圆括号不能省略。

2) 一个 C 语言程序总是从 main()函数开始执行。主函数执行完毕，程序执行结束。

3) C 语言编译系统区分字母大小写。C 语言把大小写字母视为两个不同的字符，并规定每条语句或数据说明均以分号(;)结束。分号是语句不可缺少的组成部分。

4) 主函数 main()既可以放在其他函数之前，也可以放在最后。习惯上将主函数 main()放在最后。

5) C 语言程序中所调用的函数，既可以是由系统提供的库函数，也可以是由设计人员自己根据需要而设计的函数。

6) /* */或//表示注释。/* */可以注释多行，而//只能注释一行。注释语句是用于给他人看的，编译器不作任何解释。

1.2.4 C 语言的语句

与其他高级语言一样，C 语言也是利用函数体中的可执行语句向计算机系统发出操作命令。按照语句功能或构成的不同，可将 C 语言的语句分为五类。

1. 控制语句

控制语句完成一定的控制功能。C 语言只有九条控制语句，又可细分为三种。