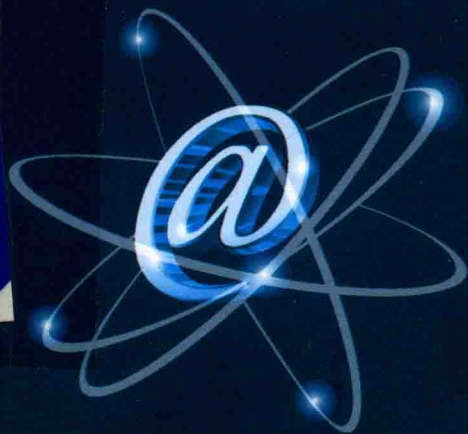


卓越工程师教育培养计算机类创新系列规划教材

C语言

程序设计实践教程 学习辅导

主 编 吉根林 陈 波
副主编 高 茜 赵 斌 季菊辉



科学出版社

卓越工程师教育培养计算机类创新系列规划教材

C 语言程序设计实践教学学习辅导

主 编 吉根林 陈 波

副主编 高 茜 赵 斌 季菊辉

科 学 出 版 社

北 京

内 容 简 介

本书是作者多年讲授“C 语言程序设计”课程及指导学生实验的教学经验的集成,与科学出版社出版的主教材《C 语言程序设计实践教程》相配套。本书内容与主教材完全同步。第一部分是理论知识辅导篇,每章由3个模块组成,分别是本章导学、重点难点释疑和习题解答,习题解答部分包括原书课后习题的解答,以及补充自测题与解答;第二部分是上机实践指导篇,每章由两个模块组成,分别是主教材“实验指导与参考程序”,以及实验拓展与参考程序。最后,在附录中给出主教材附录中4套模拟试卷的参考解答。

本书可以配合主教材《C 语言程序设计实践教程》,作为高等学校计算机类专业及其相关专业 C 语言程序设计课程的辅导资料使用,起到衔接课堂教学和指导实验教学的作用,也可作为计算机等级考试(C 语言程序设计)辅导用书,还可供计算机软件开发人员参考。

图书在版编目(CIP)数据

C 语言程序设计实践教程学习辅导/吉根林,陈波主编. —北京:科学出版社, 2018.2

卓越工程师教育培养计算机类创新系列规划教材

ISBN 978-7-03-056549-5

I. ①C… II. ①吉… ②陈… III. ① C 语言—程序设计—教材
IV. ① TP312.8

中国版本图书馆 CIP 数据核字(2018)第 025871 号

责任编辑:邹杰 张丽花/责任校对:郭瑞芝

责任印制:霍兵/封面设计:迷底书装

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>

天津翔远印刷有限公司印刷

科学出版社发行 各地新华书店经销

*

2018年2月第一版 开本:787×1092 1/16

2018年2月第一次印刷 印张:21 1/2

字数:510 000

定价:49.80元

(如有印装质量问题,我社负责调换)

前 言

《C 语言程序设计实践教程》一书自出版以来,受到广大读者的欢迎,大家对于该书分为理论知识辅导篇和上机实践指导篇两个部分的组织结构、采用理论和实践相结合的方式组织教学内容,以及以任务驱动方式进行讲授的特点给予了肯定和好评。

为了更好地体现《C 语言程序设计实践教程》的教改特色并发挥其作用,我们编写了本书。在编写过程中,我们努力做到以下几点。

(1) 紧扣教材,为学习者自主学习提供学习辅导和上机实践的指导。因为主教程的编排结构就已经突出了实践,分为理论知识篇和上机实践篇,所以本书的章节安排与教程完全一致,以避免目前已有同类辅导书籍的章节与原书不一致,给学习造成不便的问题。

(2) 内容充实,为计算机专业学习者和非计算机专业(参加计算机 C 语言等级考试)的学习者提供充分的选择余地。

(3) 融入过程性测试。与目前已有 C 语言辅导教程最大的不同是,本书为主教程每一章节都增加了过程性自测环节。这是考虑到程序开发课程的教学特点是强调学生的实践能力培养,注重学生在平时的项目训练和循序渐进的开发过程。本书包含对基本概念和方法的教学以及基本案例的讲解,更重要的是包含了对学生进行项目设计开发练习和撰写分析、设计、测试等能力的训练和提高。

本书内容与主教材完全同步。第一部分是理论知识辅导篇,每章由 3 个模块组成,分别是本章导学、重点难点释疑和习题解答,习题解答部分包括主教材课后习题的解答,以及补充自测题与解答;第二部分是上机实践指导篇,每章由两个模块组成,分别是主教材实验指导与参考程序,以及实验拓展与参考程序。最后,在附录中给出主教材附录中 4 套模拟试卷的参考解答。

本书由吉根林教授和陈波教授主编,高茜副教授、赵斌副教授和季菊辉老师担任副主编。陈波教授负责编写理论知识辅导篇每章的“本章导学”和“重点难点释疑”,高茜副教授负责编写理论知识辅导篇每章的“习题解答”和上机实践指导篇每章的“实验指导与参考程序”部分;季菊辉老师负责编写理论知识辅导篇每章的“过程性自测及解答”和上机实践指导篇实验一~实验九的“实验拓展与参考程序”部分;赵斌副教授负责编写上机实践指导篇“综合实训”的“实验指导与参考程序”以及“实验拓展与参考程序”部分。吉根林教授和陈波教授负责统稿和审定。本书在写作过程中,参考了季菊辉老师编写的《江苏省计算机等级考试二级 C 语言复习资料》。

本书的出版是江苏省高等教育教学改革重点课题(2015JSJG034)和南京师范大学“信息安全素养与软件工程实践创新”教学团队建设的成果。在出版过程中还得到了科学出版社邹杰编辑的大力支持,在此表示衷心的感谢。由于作者水平有限,书中难免存在不妥之处,敬请读者批评指正。本书为任课教师提供书中源代码,请登录 <http://www.sciencereading.cn>,选择“网上书店”,检索图书名称,在图书详情“资源下载”栏目中获取。

作者 E-mail: bchen@njnu.edu.cn

作 者

2017 年 10 月

目 录

理论知识辅导篇

第 1 章 绪论	3	4.2.1 实现顺序结构的语句	43
1.1 本章导学	3	4.2.2 实现分支结构的语句	44
1.2 重点难点释疑	3	4.2.3 实现循环结构的语句	48
1.2.1 C 语言的特点	3	4.2.4 其他语句	50
1.2.2 程序的基本结构	3	4.3 习题解答	51
1.2.3 书写格式	4	4.3.1 课后习题解答	51
1.3 习题解答	4	4.3.2 过程性自测及解答	55
1.3.1 课后习题解答	4	第 5 章 函数和模块化程序设计	69
1.3.2 过程性自测及解答	5	5.1 本章导学	69
第 2 章 数据的基本类型与基本运算	8	5.2 重点难点释疑	69
2.1 本章导学	8	5.2.1 非递归函数的定义、声明、调用及执行过程	69
2.2 重点难点释疑	8	5.2.2 函数调用时参数的传递	72
2.2.1 基本数据类型	8	5.2.3 递归函数的定义、声明、调用及执行过程	73
2.2.2 运算符与表达式	13	5.2.4 变量的作用域	73
2.2.3 数据类型转换	21	5.3 习题解答	75
2.3 习题解答	22	5.3.1 课后习题解答	75
2.3.1 课后习题解答	22	5.3.2 过程性自测及解答	80
2.3.2 过程性自测及解答	27	第 6 章 数组	99
第 3 章 数据的输入/输出	33	6.1 本章导学	99
3.1 本章导学	33	6.2 重点难点释疑	99
3.2 重点难点释疑	33	6.2.1 数组的声明及初始化	99
3.2.1 C 语言的输入/输出	33	6.2.2 数组元素的引用	101
3.2.2 字符的非格式化输入/输出函数	33	6.2.3 字符数组	102
3.2.3 格式化输出函数 printf	34	6.2.4 数组名作为函数参数	104
3.2.4 格式化输入函数 scanf	35	6.3 习题解答	104
3.3 习题解答	35	6.3.1 课后习题解答	104
3.3.1 课后习题解答	35	6.3.2 过程性自测及解答	109
3.3.2 过程性自测及解答	38	第 7 章 指针	143
第 4 章 程序的基本结构与基本语句	43	7.1 本章导学	143
4.1 本章导学	43	7.2 重点难点释疑	143
4.2 重点难点释疑	43		

7.2.1 指针的概念	143	11.2 重点难点释疑	235
7.2.2 指向变量的指针	144	11.2.1 按位运算符	235
7.2.3 指向数组的指针	145	11.2.2 移位运算符	235
7.2.4 指针数组和指向指针的指针	147	11.3 习题解答	236
7.2.5 指针与函数	149	11.3.1 课后习题解答	236
7.3 习题解答	151	11.3.2 过程性自测及解答	237
7.3.1 课后习题解答	151		
7.3.2 过程性自测及解答	153		
第 8 章 自定义数据类型与链表	171		
8.1 本章导学	171		
8.2 重点难点释疑	171		
8.2.1 结构体类型	171		
8.2.2 单向链表	175		
8.2.3 共用体类型	178		
8.2.4 枚举类型	179		
8.3 习题解答	180		
8.3.1 课后习题解答	180		
8.3.2 过程性自测及解答	185		
第 9 章 文件	212		
9.1 本章导学	212		
9.2 重点难点释疑	212		
9.2.1 文件的概念	212		
9.2.2 文件的常用操作	212		
9.3 习题解答	215		
9.3.1 课后习题解答	215		
9.3.2 过程性自测及解答	219		
第 10 章 编译预处理与多文件组织	224		
10.1 本章导学	224		
10.2 重点难点释疑	224		
10.2.1 常用预处理命令	224		
10.2.2 宏定义	224		
10.2.3 条件编译	225		
10.3 习题解答	226		
10.3.1 课后习题解答	226		
10.3.2 过程性自测及解答	228		
第 11 章 位操作	235		
11.1 本章导学	235		
		上机实践指导篇	
		实验一 Visual C++ 6.0 集成开发环境的 使用	241
		【实验 1.1 指导与参考程序】	241
		【实验 1.2 指导与参考程序】	241
		【实验 1.3 指导与参考程序】	241
		实验二 数据的基本类型与基本运算	242
		【实验 2.1 指导与参考程序】	242
		【实验 2.2 指导与参考程序】	242
		【实验拓展与参考程序】	243
		实验三 数据的输入与输出	244
		【实验 3.1 指导与参考程序】	244
		【实验 3.2 指导与参考程序】	245
		【实验拓展与参考程序】	246
		实验四 数据的基本语句与基本结构	248
		【实验 4.1 指导与参考程序】	248
		【实验 4.2 指导与参考程序】	249
		【实验 4.3 指导与参考程序】	251
		【实验拓展与参考程序】	255
		实验五 函数	256
		【实验 5.1 指导与参考程序】	256
		【实验 5.2 指导与参考程序】	259
		【实验拓展与参考程序】	262
		实验六 数组	264
		【实验 6.1 指导与参考程序】	264
		【实验 6.2 指导与参考程序】	266
		【实验 6.3 指导与参考程序】	271
		【实验 6.4 指导与参考程序】	274
		【实验拓展与参考程序】	278
		实验七 指针	282
		【实验 7.1 指导与参考程序】	282

【实验 7.2 指导与参考程序】	283	实验十 综合实训	303
【实验拓展与参考程序】	285	【综合实训 1 指导与参考程序】	303
实验八 自定义数据类型与链表	287	【综合实训 2 指导与参考程序】	307
【实验 8.1 指导与参考程序】	287	【实验拓展与参考程序】	318
【实验 8.2 指导与参考程序】	288	模拟试卷参考解答	321
【实验拓展与参考程序】	291	期末试卷 (A 卷) 参考解答	321
实验九 文件	293	期末试卷 (B 卷) 参考解答	324
【实验 9.1 指导与参考程序】	293	期末试卷 (C 卷) 参考解答	327
【实验 9.2 指导与参考程序】	297	期末试卷 (D 卷) 参考解答	332
【实验拓展与参考程序】	300		



理论知识辅导篇

第 1 章 绪 论

1.1 本章导学

- (1) 程序、程序设计以及程序设计语言的基本概念。
- (2) C 语言的特点以及 C 语言的标准。
- (3) C 语言程序基本结构与书写规则。

1.2 重点难点释疑

1.2.1 C 语言的特点

C 语言是国际上广泛流行的一种计算机程序设计语言，具有强大的生命力，其特点如下。

- (1) 简洁、紧凑、灵活、书写自由。
- (2) 处理能力强，运算符和数据类型丰富，有位运算。
- (3) 程序设计结构化、模块化。
- (4) 生成目标代码质量高。
- (5) 可移植性好。
- (6) 可以直接操作硬件，能实现汇编语言的大部分功能，有人把它称为中级语言。

1.2.2 程序的基本结构

1. 注意事项

C 程序由函数构成，要注意以下几点。

- (1) 一个 C 语言源程序由一个或多个函数组成，但有且只能有一个 `main` 函数。
- (2) `main` 函数也叫主函数，是 C 语言编译系统提供的特殊函数，可以放在程序中的任何位置，但不能定义在其他函数体内。
- (3) 一个 C 程序总是从 `main` 函数开始执行，在 `main` 函数中结束。
- (4) `main` 函数可以调用其他任何一个函数，其他函数之间也可以相互调用，但不能调用 `main` 函数。
- (5) 函数可以嵌套调用，也可以递归调用，但不能嵌套定义。

嵌套调用指在一个函数定义的函数体中允许调用另一个函数，另一个函数还可以调用其他函数；递归调用是指在一个函数定义的函数体中调用函数自身。不能嵌套定义是指在定义一个函数时不能再定义另外的函数。有关函数的调用将在第 5 章详述。

2. 函数类型

除了 main 函数外，C 程序还有库函数和用户自定义的函数，要注意区别以下几个概念。

(1) main 函数：是必不可少的函数，在一个 C 程序中有且只能有一个 main 函数。

(2) 库函数：是系统提供的函数，如 printf()、sqrt() 等，系统已经赋予这些函数具体的功能，用户只要包含在头文件中就可直接调用，如使用数学函数 sqrt()，必须在函数开头写上编译预处理命令 #include <math.h>。

(3) 自定义函数：是用户根据需要自己设计的函数，能实现一定的功能，如求最大值函数 max()、排序函数 sort()。

3. 函数构成

一个函数由两部分组成：函数首部和函数体。

(1) 函数首部：包括函数返回值类型、函数名、函数参数类型和函数参数名，函数名后必须有一对圆括号，有的函数没有参数但圆括号必须有。

(2) 函数体：放在一对花括号({})内，包含声明部分和执行部分，声明部分主要为变量定义和函数声明，执行部分由若干语句组成。

1.2.3 书写格式

编写 C 语言程序时，要按照一定的格式书写，其格式特点如下。

(1) C 语言区分大小写，主要用小写字母书写。

(2) C 程序书写格式自由，一行内可以写几个语句，一个语句可以分写在多行上，可使用空行和空格，C 程序没有行号。

(3) 每个语句和数据声明的最后必须有一个分号“;”，分号是 C 语句必不可少的组成部分，但复合语句除外。

(4) C 语言本身没有输入/输出语句，输入和输出操作通过调用库函数完成，库函数定义在头文件“stdio.h”中。

(5) 可以用 /*...*/ 对 C 程序中的任何部分作注释，增加程序的可读性。注释不产生编译代码，不能嵌套。

(6) 为了使程序易读，书写程序常采用锯齿形书写格式，形成层次，应做到以下几点：使用空格或 Tab 缩进，花括号({})尽量对齐，有足够的注释，有合适的空行。

1.3 习题解答

1.3.1 课后习题解答

1. 简要说明程序、程序设计、程序设计语言的概念。

【参考解答】 程序就是为完成某项任务而编写的一组计算机指令序列。编写程序的过程称为程序设计。

程序设计语言是用来实现人与计算机之间交流的，它经历了从机器语言、汇编语言到高

级语言的发展历程。

2. 简要说明高级语言程序的开发过程。

【参考解答】 通常,高级语言程序的开发遵循以下步骤。

- (1) 明确问题。
- (2) 算法设计。
- (3) 选择某种语言进行编程。
- (4) 测试与调试。
- (5) 运行与维护。

3. 简述 C 语言的主要特点。

【参考解答】 与其他高级语言相比,C 语言之所以发展迅速,成为最受欢迎的语言之一,主要原因是它具有强大的功能。归纳起来,C 语言具有以下一些特点。

- (1) C 语言是中级语言。
- (2) C 语言是结构化程序设计语言。
- (3) 语言简洁、紧凑,使用方便、灵活。
- (4) 运算符和数据结构丰富,表达式多样。
- (5) 语法限制不太严格,程序设计自由度大。
- (6) 生成的目标代码质量高。
- (7) C 程序的可移植性好。

4. 简述 C 程序的结构。

【参考解答】 一个 C 语言源程序是由一个或多个函数定义顺序组成的,其中必须有一个函数名为 main 的函数,main()函数又称为主函数。C 语言源程序的次要构成成分有:编译预处理命令、注释和声明。

1.3.2 过程性自测及解答

一、填空题

1. C 语言对大小写字母是_____ (敏感/不敏感) 的。

【答案】 敏感

2. 每条 C 语句都以_____ 符号结尾。

【答案】 ;

3. C 语言中用_____ 符号进行语句块的注释,使用_____ 符号对一行语句进行注释。

【答案】 第一空: /* */ 第二空: //

4. C 语言没有输入/输出语句,输入/输出操作是通过_____ 完成的。

【答案】 函数

5. C 程序必须经过_____、_____ 后生成可执行文件,才能运行。

【答案】 第一空: 编译 第二空: 链接

6. 程序写好后,可能会出现错误,其中_____错误是指程序没有按照语言的语法规则来书写,这类错误可以由_____来发现。

【答案】 第一空:编译 第二空:编译器

7. 如果程序没有完成预期的功能,这种类型的错误称为_____错误,可以通过_____工具来查找。

【答案】 第一空:逻辑 第二空:调试

8. 由于对程序运行环境的非正常情况考虑不周而导致的程序异常终止,我们称这类错误为_____错误。

【答案】 运行

二、选择题

1. C 语言规定,在一个源程序中 main 函数的位置()。

- A. 必须在开头
- B. 必须在最后
- C. 必须在预处理命令的后面
- D. 可以在其他函数之前或之后

【解析】 main 函数也叫主函数,是 C 语言编译系统提供的特殊函数,可以放在程序中的任何位置,但不能定义在其他函数体内。

【答案】 D

2. 下列关于 main 函数的叙述正确的是()。

- A. main 函数必须出现在其他函数之前
- B. main 函数中至少必须有一条语句
- C. C 程序总是从 main 函数的第一条语句开始执行
- D. C 程序总是在执行完 main 函数的最后一条语句后结束

【解析】 main 函数可以放在其他函数之前或之后,也可以为空函数。C 程序总是从 main 函数的第一条语句开始执行,但不一定在执行完 main 函数的最后一条语句后结束,因为有函数调用。

【答案】 C

3. 以下关于 C 语言函数的叙述中正确的是()。

- A. C 语言系统以函数为单位编译源程序
- B. main 函数必须放在程序开始
- C. 用户定义的函数可以被一个或多个函数调用任意多次
- D. 在一个函数体内可以定义另外一个函数

【解析】 C 语言系统以文件为单位编译源程序,程序由函数组成。除了 main 函数外,函数可以被其他函数多次调用。函数不能定义在其他函数体内,即不能嵌套定义。

【答案】 C

4. 以下关于 C 语言程序的叙述中正确的是()。

- A. 在编译时可以发现注释中的拼写错误
- B. C 语言程序的每一行只能写一条语句
- C. main()必须位于程序的开始

D. C 语言程序可以由一个或多个函数组成

【解析】 C 程序的注释不产生编译代码，编译时跳过去，发现不了其中的错误。C 程序书写格式自由，一行内可以写几个语句，一个语句可以分写有多行上。main 函数可以放在程序中的任何位置。C 程序可以由一个或多个函数组成，函数可以为空函数，包括 main 函数。

【答案】 D

5. 以下关于 C 语言函数的叙述中错误的是 ()。

- A. 一个 C 源程序由若干函数定义组成，其中必须有且仅有一个名为 main 的函数定义
- B. 函数定义由函数头部和函数体两部分组成
- C. 在一个函数定义的函数体中允许定义另一个函数
- D. 在一个函数定义的函数体中允许调用另一个函数或调用函数自身

【解析】 函数可以嵌套调用，也可以递归调用，但不能嵌套定义。C 的意思是嵌套定义，故错误，D 的意思是嵌套调用和递归调用，正确。

【答案】 C

6. 一个用 C 语言编写的源程序中，() 是必不可少的。

- A. 取名为 main 的函数定义
- B. #include <stdio.h>
- C. 变量声明
- D. 注释

【答案】 A

7. 以下叙述中正确的是 ()。

- A. 在编译时可以发现注释中的拼写错误
- B. C 语言程序的每一行只能写一条语句
- C. 构成 C 程序的最小单位是语句
- D. C 语言程序可以由一个或多个函数组成

【答案】 D

第 2 章 数据的基本类型与基本运算

2.1 本章导学

- (1) C 语言的基本数据类型（整型、实型、字符型）的概念、详细分类、内存表示及使用方法。
- (2) 常量、变量的使用语法、规范。
- (3) 数据类型的基本运算。
- (4) 数据类型的转换。

2.2 重点难点释疑

2.2.1 基本数据类型

在 C 语言中，不同类型的数据在内存中占用的空间不同，为了节约内存空间和提高数据处理效率以及使用方便，C 语言中将数据分为很多种类型，如图 2.1 所示。

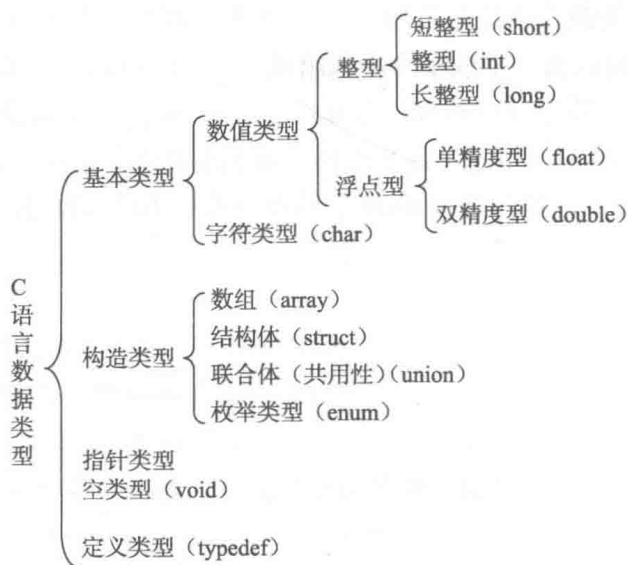


图 2.1 C 语言的基本数据类型

- (1) C 语言的基本数据类型有整型、实型、字符型 3 种。这是本部分详细讨论的内容。
- (2) 构造类型根据已定义的一个或多个数据类型用构造的方法来定义，主要有数组、结构体和联合体 3 种。枚举类型在有的教材中将其归为数值类型。
- (3) 指针类型是一种特殊的类型，也是 C 语言的特色，其值用来表示某个变量在内存中的地址。

(4) 空类型主要用于说明函数返回值, 当调用后不需要向调用者返回函数值时, 可以将函数的返回值类型定义为空类型, 类型说明符为 `void`。

1. 常量和变量、标识符

1) 常量和变量

对于基本数据类型量, 按其值是否可以改变分为常量和变量两种。

程序运行过程中其值不能改变的量称为常量, 其值可以改变的量称为变量。

常量有整型常量、实型常量、字符常量、字符串常量。

变量有整型变量、浮点型变量、字符型变量, 没有字符串变量。变量有变量名、变量值、变量在内存中存放的地址。对变量的操作主要通过变量名进行, 变量名用标识符来表示。

2) 标识符

标识符是用来表示符号常量、变量、数组、函数和文件等对象的名称的有效字符系列, 以字母或下划线开头的字母、数字、下划线序列。例如, `abc`、`_123` 都是合法的标识符。注意, 标识符可以为数字但不能以数字开头, 标识符中不能有空格、标点符号、运算符或除字母、数字、下划线外的其他字符, 标识符区分大小写。

3) 标识符类型

(1) 关键字。

关键字是指由系统内部定义的, 具有特殊含义和用途的标识符, 其代表了固定的含义, 程序中不能另作他用, 只能由小写字母组成。C 语言中的关键字共有以下 32 个:

<code>auto</code>	<code>break</code>	<code>case</code>	<code>char</code>	<code>const</code>
<code>continue</code>	<code>default</code>	<code>do</code>	<code>double</code>	<code>else</code>
<code>enum</code>	<code>extern</code>	<code>float</code>	<code>for</code>	<code>goto</code>
<code>if</code>	<code>int</code>	<code>long</code>	<code>register</code>	<code>return</code>
<code>short</code>	<code>signed</code>	<code>sizeof</code>	<code>static</code>	<code>struct</code>
<code>switch</code>	<code>typedef</code>	<code>union</code>	<code>unsigned</code>	<code>void</code>
<code>volatile</code>	<code>while</code>			

在计算机等级考试中, 经常会出现判断是否是关键字的题目。要注意的是 `main` 不是关键字, 而属于预定义标识符。

(2) 预定义标识符。

在 C 语言中有特定的含义, 可以另作他用但不会失去原有含义, 如库函数名 `printf` 和预编译命令 `define` 等。

(3) 用户自定义标识符。

用户自定义标识符是用户根据需要定义的标识符, 用来表示变量名、数组名、函数名等。在定义标识符时必须遵循其命名规则, 最好做到见名知意, 注意不要将 `l` 与 `I`、`o` 与 `0` 等混淆。

2. 整型数据

整型数据包括整型常量和整型变量, 在内存中以补码的形式存放。正数的补码与原码相同, 为该数的二进制形式, 左边第一位为符号位 `0`, 负数的补码将该数的绝对值的二进制形式按位取反再加 `1`, 左边的第一位为符号位 `1`。例如, `+37` 的 8 位补码为 `00100101`, `-37` 的 8 位

补码为 11011011。

1) 整型常量

整型常量有符号常量和直接常量两种。

(1) 符号常量。

符号常量用标识符来代表一个常量，一般用大写字母表示。符号常量在使用前必须先定义，其定义格式如下：

```
#define 标识符 常量
```

其中，`#define` 是宏定义预处理命令，不是 C 语句，后面不能有分号，一个预处理命令只能写在一行上，一行也只能写一个预处理命令。其功能是将该标识符定义为其后的常量值，一经定义，在其后的程序中出现的该标识符都被替换为该常量，其值在作用域内不再改变，也不能再被赋值。例如，如果有以下定义：

```
#define PI 3.14159
```

则在以下程序中若有 `PI` 则全部用 `3.14159` 来代替。这实际上是一个不带参数的宏定义，在第 10 章中将详细讨论带参数的宏定义。

(2) 直接常量。

直接常量也叫字面常量，此处先讨论整型常量。

整型常量就是整常数，可以用十进制、八进制或者十六进制表示。

十进制整数由正负号及 0~9 十个数字组成，如 23、-456、0。

八进制整数以数字 0 开头，由 0~7 八个数字组成，如 012、0763。

十六进制整数由 0x 开头，由 1~9 十个数字和小写字母 a~f 或者大写字母 A~F 组成，如 0x3d、0XAF2。

说明：

① 如果表示长整型常量，可以在整常量后面加上小写字母 `l` 或者大写字母 `L`。如 `12L`，表示 12 在内存中是以 `long int` 的形式存放的，占 32 位，不是 `int` 型的 16 位。

② 如果表示无符号整型常量，可以在数据后面加上小写字母 `u` 或者大写字母 `U`，如 `5U`，而 `5UL` 表示无符号长整型数 5。

2) 整型变量

使用变量时必须“先定义，后使用”。凡未被事先定义的，系统不把它作为变量名，这样能保证程序中变量名的正确使用。

(1) 整型变量的类型。

每一个变量被指定为一个确定类型，在编译时就能为其分配相应的内存单元。对于整型变量，其基本类型为 `int`，简称整型，除此之外，整型变量还有短整型和长整型，以及有符号整型和无符号整型之分，共 6 种，如表 2.1 所示。

整型变量有 6 种，最常用的是 `int` 型，在内存中占 2 字节，表示范围为 -32768~32767，若某整型变量的值超过该范围，则需定义为 `long` 型，在一些特殊的场合，可能需要将某些变量定义为无符号整型 (`unsigned`)。其他类型很少使用。在表 2.1 中，需要记忆的是每种类型在内存中占用的字节数，以及 `int` 和 `unsigned int` 型数的表示范围。

(2) 变量的定义。

变量定义一般放在函数开头，变量定义的一般格式如下：