



普通高等教育计算机应用技术实训规划教材

C 语言

程序设计

C YUYAN
CHENGXU SHEJI

卢 杉 刘金魁 ◎主 编
韩秀娟 郝文玲 孟洪兵 ◎副主编
侯继红 ◎主 审



北京邮电大学出版社
www.buptpress.com

普通高等教育计算机应用技术实训规划教材

C 语言程序设计

卢 杉	刘金魁	主 编
韩秀娟	郝文玲	副主编
	侯继红	主 审

北京邮电大学出版社

· 北 京 ·

内 容 简 介

本书是“普通高等教育计算机应用技术实训规划教材”系列丛书之一,有较强的可读性和适用性。全书共 11 章,内容包括:C 语言概述,C 语言的数据类型、运算符与表达式,程序控制结构,数组,函数,编译预处理,结构体与共用体,指针,位运算,文件,以及 C 语言实验指导等。每章后面均附有一定数量的习题和上机实训内容。

本书定位准确,内容新颖,概念清晰,例题丰富,文字流畅,通俗易懂。符合初学者的认识规律,是初学者学习 C 语言的理想教材。本书可以作为高等学校计算机及相关专业学生学习 C 语言程序设计的教材,也可作为自学及参加相关考试的参考书。

图书在版编目(CIP)数据

C 语言程序设计/卢杉,刘金魁主编. —北京:北京邮电大学出版社,2009.12

ISBN 978-7-5635-2119-7

I. C… II. ①卢…②刘… III. C 语言—程序设计 IV. TP312

中国版本图书馆 CIP 数据核字(2009)第 179793 号

书 名: C 语言程序设计

主 编: 卢 杉 刘金魁

责任编辑: 张珊珊

出版发行: 北京邮电大学出版社

社 址: 北京市海淀区西土城路 10 号(邮编:100876)

发 行 部: 电话: 010-62282185 传真: 010-62283578

E-mail: publish@bupt.edu.cn

经 销: 各地新华书店

印 刷: 北京源海印刷有限责任公司

开 本: 787 mm×1 092 mm 1/16

印 张: 17

字 数: 420 千字

印 数: 1—3 000 册

版 次: 2009 年 12 月第 1 版 2009 年 12 月第 1 次印刷

ISBN 978-7-5635-2119-7

定 价: 29.00 元

• 如有印装质量问题,请与北京邮电大学出版社发行部联系 •

前 言

C 语言是一种通用的程序设计语言,具有丰富的数据类型。由于其具有简洁灵活、目标程序效率高、可移植性好以及兼有高级语言和低级语言的优点等特点,C 语言成为各工科专业理想的程序设计语言之一。

本书是按照国家教育部高校基础教育的基本要求,由长期从事计算机技术研究、计算机教学和计算机培训辅导的资深教师编写,它凝聚了作者多年的教学经验和智慧。本书概念准确、结构完整、深入浅出、通俗易懂,具有较强的针对性和操作性。

本书由卢杉、刘金魁任主编,韩秀娟、郝文玲、孟洪兵任副主编。焦作大学机电工程学院卢杉编写第 3 章,河南理工大学高职学院刘金魁编写第 5 章和附录,河南理工大学计算机科学与技术学院韩秀娟编写第 6 章和第 7 章,焦作大学机电工程学院郝文玲编写第 1 章和第 4 章,塔里木大学信息工程学院孟洪兵编写第 10 章和第 11 章,河南理工大学高职学院张兰编写第 2 章,河南理工大学高职学院康超编写第 9 章,塔里木大学信息工程学院张小刚编写第 8 章。全书由焦作大学侯继红教授主审。

由于编者的水平有限,书中难免存在疏漏与不足之处,欢迎广大读者和同行提出宝贵意见,作者联系 E-mail:bycyyjc@126.com。为便于教学,本书免费提供书中习题答案和电子教案,可向北京邮电大学出版社索取。

编 者

目 录

第 1 章 C 语言概述	1
1.1 程序设计与计算机语言	1
1.1.1 程序设计	1
1.1.2 计算机语言	1
1.2 C 语言的发展过程和特点	3
1.2.1 C 语言的发展过程	3
1.2.2 C 语言的特点	3
1.3 C 程序的基本结构和组成	4
1.3.1 C 程序的基本结构	4
1.3.2 C 程序的基本组成	5
1.4 C 语言的基本标识符	6
1.5 运行 C 程序的步骤实例	7
1.5.1 运行 C 程序的一般过程	7
1.5.2 运行 C 程序的一个实例	7
本章小结	10
习题	10
实训	11
第 2 章 C 语言的数据类型、运算符与表达式	13
2.1 C 语言的数据类型	13
2.2 常量	14
2.2.1 整型常量	14
2.2.2 实型常量	15
2.2.3 字符常量	16
2.2.4 字符串常量	17
2.2.5 符号常量	18
2.3 变量	18
2.3.1 变量的定义与初始化	18
2.3.2 整型变量	19
2.3.3 实型变量	21

2.3.4 字符型变量	22
2.4 运算符及表达式	24
2.4.1 算术运算符和算术表达式	24
2.4.2 关系运算符和关系表达式	25
2.4.3 逻辑运算符和逻辑表达式	26
2.4.4 赋值运算符和赋值表达式	27
2.4.5 逗号运算符和逗号表达式	27
2.4.6 变量的自增(++)、自减(--)运算符	28
2.4.7 混合运算和类型转换	29
本章小结	30
习题	30
实训	33
第3章 程序控制结构	36
3.1 顺序结构	36
3.1.1 赋值语句	36
3.1.2 数据输出	37
3.1.3 数据输入	41
3.1.4 应用实例	44
3.2 选择结构	46
3.2.1 单分支选择结构	46
3.2.2 双分支选择结构	47
3.2.3 多分支选择结构	52
3.3 循环结构	54
3.3.1 循环的基本概念	55
3.3.2 用 while 语句设计循环结构程序	55
3.3.3 用 do-while 语句设计循环结构程序	57
3.3.4 用 for 语句设计循环结构程序	58
3.3.5 break 语句与 continue 语句	61
3.3.6 几种循环语句的比较	62
3.3.7 循环的嵌套	63
3.3.8 应用实例	64
3.4 实际应用	67
本章小结	71
习题	72
实训	78
第4章 数组	83
4.1 数组的基本概念	83

4.2 一维数组	84
4.2.1 一维数组的定义	84
4.2.2 一维数组的初始化	84
4.2.3 一维数组的引用	85
4.2.4 一维数组程序举例	86
4.3 二维数组	87
4.3.1 二维数组的定义	88
4.3.2 二维数组的初始化	88
4.3.3 二维数组的引用	89
4.3.4 二维数组程序举例	90
4.4 字符数组	91
4.4.1 字符数组的定义、初始化及引用	91
4.4.2 字符串与字符数组	92
4.5 字符串处理函数	94
4.5.1 字符串输出函数 puts	94
4.5.2 字符串输入函数 gets	94
4.5.3 字符串连接函数 strcat	95
4.5.4 字符串拷贝函数 strcpy	95
4.5.5 字符串比较函数 strcmp	96
4.5.6 测字符串长度函数 strlen	96
4.5.7 字符串小写函数 strlwr	96
4.5.8 字符串大写函数 strupr	96
4.6 应用实例	97
4.7 实际应用	100
本章小结	103
习题	103
实训	108
第5章 函数	111
5.1 模块化设计与函数	111
5.1.1 模块化设计	111
5.1.2 函数	112
5.2 函数定义与分类	114
5.2.1 函数的定义	114
5.2.2 函数的分类	115
5.3 函数的调用	117
5.3.1 函数调用的方式	117
5.3.2 函数的参数传递	117
5.4 函数的声明与返回	120

5.4.1 函数的声明	120
5.4.2 函数的返回值	121
5.5 函数的嵌套和递归调用	121
5.5.1 函数的嵌套调用	121
5.5.2 函数的递归调用	122
5.6 变量的存储类型和作用域	126
5.6.1 变量的存储类型	126
5.6.2 变量的作用域	129
5.7 外部函数和内部函数	132
5.7.1 外部函数	132
5.7.2 内部函数	133
5.8 实际应用	134
本章小结.....	136
习题.....	136
实训.....	141
第 6 章 编译预处理.....	145
6.1 宏定义	145
6.1.1 不带参数的宏定义	145
6.1.2 带参数的宏定义	147
6.2 文件包含处理	149
6.3 条件编译	150
6.3.1 #ifdef...#endif 和 #ifndef...#endif 命令	150
6.3.2 #if...#endif	151
本章小结.....	151
习题.....	151
实训.....	153
第 7 章 结构体与共用体.....	156
7.1 结构体的定义及其变量的初始化	156
7.1.1 结构体的定义	156
7.1.2 结构类型变量的说明	157
7.1.3 结构体变量的初始化	158
7.2 结构体类型变量的引用	160
7.3 结构体数组	160
7.3.1 定义	161
7.3.2 结构体数组初始化	161
7.4 指针和结构体	163
7.4.1 指向结构体变量的指针	163

7.4.2 指向结构体数组的指针	164
7.4.3 结构指针参数	164
7.5 用指针处理链表	165
7.5.1 链表	165
7.5.2 建立链表	166
7.5.3 链表输出	168
7.5.4 对链表中的元素进行删除	168
7.5.5 对链表插入结点	170
7.5.6 主函数	170
7.6 共用体	172
7.6.1 共用体的概念	172
7.6.2 共用体的引用方式	172
7.6.3 共用体的特点	173
7.7 枚举	173
7.8 用 typedef 定义类型	174
7.9 实际应用	175
本章小结	178
习题	179
实训	184
第 8 章 指针	189
8.1 指针概述	189
8.1.1 指针与指针变量	189
8.1.2 指针变量的定义和应用	190
8.1.3 指针变量的运算	192
8.1.4 指向指针的指针	195
8.2 指针与数组	195
8.2.1 指针与一维数组	196
8.2.2 指针与二维数组	197
8.2.3 指针与字符串	198
8.2.4 指针数组	200
8.2.5 指向指针的指针	201
本章小结	203
习题	203
实训	206
第 9 章 位运算	208
9.1 位运算和位运算符	208
9.1.1 按位与运算	209

9.1.2 按位或运算	209
9.1.3 按位异或运算	210
9.1.4 求反运算	211
9.1.5 左移运算	211
9.1.6 右移运算	211
9.1.7 位复合赋值运算符	212
9.2 位域	212
9.2.1 位域的定义和位域变量的说明	213
9.2.2 关于位域的定义和引用的说明	213
9.2.3 位域的使用	214
本章小结	215
习题	215
实训	217
第 10 章 文件	220
10.1 文件概述	220
10.1.1 文件的概念及分类	220
10.1.2 C 文件结构及其指针	220
10.1.3 文件系统的缓冲性	221
10.2 文件的打开与关闭	222
10.2.1 文件的打开函数 fopen	222
10.2.2 文件的关闭函数 fclose	223
10.3 文件的读写操作	224
10.3.1 读/写字符函数 fgetc 和 fputc	224
10.3.2 读/写字符串函数 fgets 和 fputs	224
10.3.3 读/写数据块函数 fread 和 fwrite	225
10.3.4 读/写格式化函数 fscanf 和 fprintf	226
10.4 文件的定位	226
10.5 文件检测函数	227
10.6 实际应用	228
本章小结	228
习题	229
实训	230
第 11 章 C 语言实验指导	232
11.1 实验概述	232
11.1.1 实验的性质和任务	232
11.1.2 实验的目的	232

11.1.3 实验步骤.....	232
11.2 集成开发环境.....	233
11.2.1 Turbo C启动和主界面介绍.....	233
11.2.2 各菜单项功能.....	234
11.2.3 C语言程序的调试过程.....	241
11.3 常见错误分析.....	242
11.3.1 错误类型.....	242
11.3.2 常见错误举例.....	243
本章小结.....	248
附录一 ASCII 代码表.....	249
附录二 C语言常用关键字及说明.....	250
附录三 C语言运算符及优先级.....	251
附录四 常用库函数.....	252
参考文献.....	259

 本章考点和学习目标

1. 掌握程序的构成、main 函数和其他函数。
2. 掌握头文件、数据说明、函数的开始和结束标志以及程序中的注释。
3. 掌握源程序的书写格式以及基本的编程环境。
4. 了解 C 语言的风格。

1.1 程序设计与计算机语言

1.1.1 程序设计

计算机是一种机器,通常只能接受简单的指令,如将数据 1 放在 A 处,将数据 2 放在 B 处,再将两者之和放在 C 处等,并且指令都是用 0 和 1 组成的二进制码来编写的。程序是一系列有序指令组成的集合,应当包括数据描述和操作步骤两方面内容,其中数据是操作的对象。它们之间的关系可以这样来理解:

- (1) 数据的描述称为数据结构;
- (2) 操作步骤的集合称为算法;
- (3) 程序=数据结构+算法。

1.1.2 计算机语言

编写计算机程序、进行软件开发所用的计算机语言叫做“程序设计语言”,它大致的发展历程是机器语言、汇编语言、高级语言。

1. 机器语言

计算机只能识别二进制码,而不能识别人的自然语言。因此最早的计算机语言是二进制码形式的,称为机器语言。指令是计算机能够直接识别与执行的命令,它在计算机内部以二进制码表示,一台计算机的全部指令的二进制码,构成了这台计算机的机器语言。计算机唯一能直接识别和执行的程序设计语言就是机器语言。并且不同型号的计算机有着各自不同的机器语言,所以用机器语言编写的程序通用性很差。

例如:应用 8086 CPU 完成运算 $768+12\ 288-1\ 280$,机器码如下:

101100000000000000000011

```
000001010000000000110000
```

```
001011010000000000000101
```

可见虽然用机器语言编写的程序占用的存储单元较少,执行速度也较快,但是它与人类自然语言毕竟相差太大,理解和记忆都有困难。所以现在已经很少有人用机器语言来编写程序了。

2. 汇编语言

针对机器语言的缺陷,人们用简短的英文单词或其缩写,作为“助记符”来替代一串串冗长难记的机器代码。例如:输入操作 IN、输出操作 OUT、加法操作 ADD、停止操作 END 等。这种用助记符构成的计算机程序设计语言,称为汇编语言。它虽然比机器语言易于理解,但是计算机不能直接识别和执行,必须由“翻译”系统翻译成机器语言后才能执行。

例如在 80C51 系统中计算 $100+50-80$,指令如下:

```
MOV A, #100           ;100 放到寄存器 A 中
```

```
ADD A, #50            ;50 加上 A 中原来的数,结果放在 A 中
```

```
SUBB A, #80           ;A 中的数减去 80,结果放在 A 中
```

汇编语言仅仅是机器语言的符号化。不同型号的计算机,其汇编语言仍然不能通用。但是,由于汇编语言节省内存空间,运行速度快,至今人们还经常使用它。

3. 高级语言

随着计算机应用的日益普及,越来越多的人希望能学会编写程序,这时需要一种接近人类自然语言且能通用于各种计算机的语言,计算机高级语言由此应运而生。高级语言接近人的自然语言和数学语言,用高级语言编写的程序易读、易记、易修改。

用高级语言编写程序,编程者只要将数据赋给变量,由高级语言翻译系统将变量的值存放到相应的内存单元,这样编程者就无须了解变量分配使用内存储器具体情况了。因此,不仅对编程者的专业要求降低了,而且编写的程序在不同型号的计算机上能够方便地被移植使用。

在微型计算机上,比较常见的高级语言有:适合初学者入门的 BASIC 语言;适合科学计算的 FORTRAN 语言;用于儿童趣味作图的 LOGO 语言;用于商业数据处理的 COBOL 语言;适宜进行结构化设计思想教学的 PASCAL 语言;广泛用于软件、数据库技术等领域的 C 语言;人工智能化的 PROLOG 语言等。

用高级语言编写的程序称为“源程序”。源程序是不能在计算机上直接运行的,必须将其翻译成二进制程序后才能执行。翻译过程有两种方式:一种是翻译一句执行一句,称为“解释执行”方式,完成翻译工作的程序就称为“解释程序”;另一种是全都翻译成二进制程序后再执行,承担翻译工作的程序就称为“编译程序”,编译后的二进制程序称为“目标程序”。两种方式的工作示意图如图 1-1 所示。

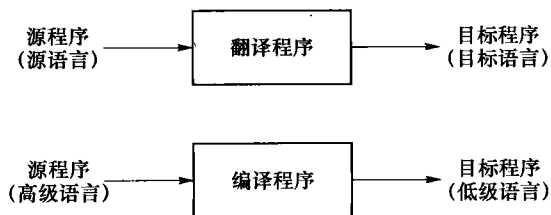


图 1-1 解释和编译两种翻译方式示意图

计算机程序设计语言的发展趋势是越来越向人们所用的自然语言靠拢。特别是近几年来,随着图形用户界面、面向对象程序设计方法以及可视化软件开发工具的兴起,软件开发者的编程工作量大为减少,而最终用户也越来越不需要有编程要求。

1.2 C语言的发展过程和特点

1.2.1 C语言的发展过程

C语言的前身是ALGOL语言。1960年ALGOL60版本推出后,很受程序设计人员的欢迎。用ALGOL60来描述算法很方便,但是它离计算机硬件系统很远,不宜用来编写系统程序。

1963年,英国剑桥大学在ALGOL语言基础上增添了处理硬件的能力,并命名为“CPL(复合程序设计语言)”。CPL由于规模大、学习和掌握困难,因而没有流行开来。

1967年,剑桥大学的马丁·理查德对CPL语言进行了简化,推出“BCPL(基本复合程序设计语言)”。

1970年,美国贝尔实验室的肯·汤普逊对BCPL进行了进一步的简化,突出了硬件处理能力,并取了“BCPL”的第一个字母“B”作为新语言的名称。同时用B语言编写了UNIX操作系统程序。

1972年,贝尔实验室的布朗·W·卡尼汉和丹尼斯·M·利奇对B语言进行了完善和扩充,在保留B语言强大的硬件处理能力的基础上,扩充了数据类型,恢复了通用性,并取了“BCPL”的第二字母“C”作为新语言的名称。此后,两人合作重写了UNIX操作系统。C语言伴随着UNIX操作系统成为一种很受欢迎的计算机语言。

1917年,为了让C语言脱离UNIX操作系统,成为在任何计算机上都能运行的通用计算机语言,卡尼汉和利奇(K&R)撰写了《C程序设计语言》一书,对C语言的语法进行了规范化的描述,成为了当时的标准。

随着微型机的普及,出现了不同的C语言版本,为了统一标准,美国标准化协会(ANSI)于1987年制定了C语言的标准,称为“ANSI C”。通常将K&R的标准称为旧标准,将“ANSI C”称为新标准。

目前在微型机上使用的C编译程序有:Turbo C、Microsoft C、Quick C。本书将以“ANSI C”为标准,以“Turbo C 2.0”为编译程序介绍C语言的内容、程序设计和调试方法。

1.2.2 C语言的特点

C语言发展如此迅速,而且成为最受欢迎的语言之一,主要因为它具有强大的功能。许多著名的系统软件(如DBASE PLUS、DBASE)都是由C语言编写的。用C语言加上一些汇编语言子程序,就更能显示C语言的优势了,如PC-DOS、WORDSTAR等就是用这种方法编写的。C语言具有以下几个特点。

1. C语言集中了低级语言和高级语言的优点

C语言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。它可以像汇编语言一样对位、字节和地址进行操作,又能像高级语言那样面向用户,容易记忆,便于阅读和

书写。

2. 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,这些函数可方便地调用,并具有多种循环语句、条件语句控制程序流向,从而使程序完全结构化。

3. C 语言功能齐全

C 语言简洁、紧凑,使用方便灵活,运算符丰富。具有各种数据类型,并引入了指针概念,可使程序效率更高。另外 C 语言也具有强大的图形功能,支持多种显示器和驱动器,而且计算功能、逻辑判断功能也比较强大,可以实现决策目的。

4. C 语言适用范围广阔

C 语言适合于多种操作系统,如 DOS、UNIX,同时也适用于多种机型。

1.3 C 程序的基本结构和组成

1.3.1 C 程序的基本结构

C 程序的基本结构是函数,一个 C 程序是由一个或多个 C 函数组成的,C 函数的实质是实现一个特定功能的程序段,一个 C 函数一般由若干条 C 语句组成。C 语句是完成某种程序功能的最小单位。下面通过一些例子来分析和说明 C 语言程序的基本结构。

【例 1-1】 输出一行文字。

```
main( )                                /* 主函数 */
{
    printf("This is a C program.\n");    /* 输出函数调用 */
}
```

运行结果:

This is a C program.

本程序的作用是输出一行信息,其中 main()表示“主函数”,每一个 C 程序都必须有且只有一个主函数。函数体由花括号“{ }”括起来,printf()是输出函数。/* … */表示注释部分,用于解释该程序或该语句的作用。注释对系统编译和运行不起任何作用,可以出现在程序的任何地方。

【例 1-2】 计算两个整数之和。

```
#include "stdio.h"                    /* 命令行,指明本程序包含 stdio.h 头文件 */
main( )
{
    int a,b,s;                        /* 声明 a,b,s 三个整型变量 */
    a = 2;b = 3;                      /* 给 a,b 赋值 */
    s = a + b;                        /* 计算 a+b 的和,并赋给变量 s */
}
```

```
printf("a = %d,b = %d,s = %d\n",a,b,s);    /* 输出 a,b 及 s 值 */
}
```

运行结果:

a = 2,b = 3,s = 5

本程序的作用是求两个整数 a 和 b 之和。其中,“a = %d,b = %d,s = %d\n”是输出的“格式控制字符串”。

【例 1-3】 求两个数的最大值。

```
#include "stdio.h"
main( )
{
    int a,b,ma;                /* 定义变量 a 和 b */
    scanf("%d,%d",&a,&b);    /* 从键盘输入 a 和 b 的值 */
    /* 调用 max 函数,并将 a 和 b 的值对应传给 x 和 y,将得到的函数结果赋给 ma 变量 */
    ma = max(a,b);
    printf("max = %d\n",ma); /* 输出 ma 的值 */
}

int max(int x,int y)          /* 函数首部 */
/* 定义 max 函数,函数值为 int 型,两个形式参数 x,y 均为 int 型 */
{
    int m;                    /* 定义 max 函数中的变量 m */
    if(x>y) m = x;            /* 条件判断语句,如果 x>y 成立,则将 x 的值赋给变量 m */
                                /* 量 m */
    else m = y;                /* 如果 x>y 不成立,则将 y 的值赋给变量 m */
    return m;                  /* 将 m 值从 max 函数带回到主函数 */
}
```

运行结果:

8,5 ↙ (输入 8 和 5 给 a 和 b)

max = 8

本程序包括两个函数:主函数 main 和被调用的函数 max。max 函数的作用是将变量 x 和 y 中较大者的值赋给变量 m,然后由 return 语句将 m 的值返回给主调函数 main。返回是通过函数名 max 带回到 main 函数的调用处。

1.3.2 C 程序的基本组成

1. C 程序是由函数构成的。

一个 C 程序有且只有一个主函数 main,函数是 C 程序的基本单位。被调用的函数可以是系统提供的库函数(例如 scanf 和 printf),也可以是用户自己编写的函数,如例 1-3 中的 max。

2. 一个函数由两部分组成,即函数的首部和函数体。

函数的首部,即函数的第一行。包括函数名、函数类型、参数(形参)名和参数类型等。如例 1-3 中的 max 函数的首部为: `int max ( int x,int y )`。

一个函数名后面必须跟一对圆括号,函数参数可以没有,如 main,但圆括号不能省,函数体即函数首部下面的花括号“{ }”内的部分。如果一个函数内有多个花括号,则最外层的一对花括号为函数体的范围。

函数体一般包括声明部分和执行部分。

声明部分用于定义变量,如例 1-2 中 main 函数中的“`int a,b,s;`”语句。执行部分是由若干条语句组成的,用以实现该函数的功能。

3. 分号是 C 语句的组成部分。
4. 一个 C 程序总是从 main 开始,再由 main 结束。
5. C 程序中一行内可以写几个语句,一个语句也可以分写在多行上。
6. C 语言的输入和输出的操作是由 C 语言提供的库函数完成的。
7. 可以用 `/ * ... * /(`与 `*` 之间不能有空格)对 C 程序中的任何部分做注释。

1.4 C 语言的基本标识符

标识符实际上是一个字符序列,用来标识变量名、符号名、函数名、数组名和文件名等。C 语言允许用做标识符的字符有:

- (1) 26 个英文字母,包括大小写(共 52 个);
- (2) 数字 0,1,2,...,9;
- (3) 下划线_。

C 语言的标识符有 3 类。

1. 关键字

关键字具有特定用途,不允许用户使用这些关键字做变量名、函数名等。由 ANSI 标准定义的关键字共 32 个。

- (1) 数据类型关键字(12 个): `char, double, enum, float, int, long, short, signed, struct, union, unsigned, void`。
- (2) 控制语句关键字(12 个): `break, case, continue, default, do, else, for, goto, if, return, switch, while`。
- (3) 存储类型关键字(4 个): `auto, extern, register, static`。
- (4) 其他关键字(4 个): `const, sizeof, typedef, volatile`。

2. 特定字

特定字具有特定的含义,一般用于预处理程序中,它们同关键字一样,不允许用做变量名、函数名等。特定字共 7 个: `# define, # endif, # ifdef, # ifndef, # include, # line, # undef`。

3. 一般标识符

通常用户是根据标识符的构成来定义标识符,作为变量名、函数名等。C 语言标识符构造规则为:必须以字母或下划线开头,后面跟随字母,或数字,或下划线,或它们的任意组合,长度一般不超过 8 个字符(较高版本可达到 31 个字符),且不能和关键字重名。