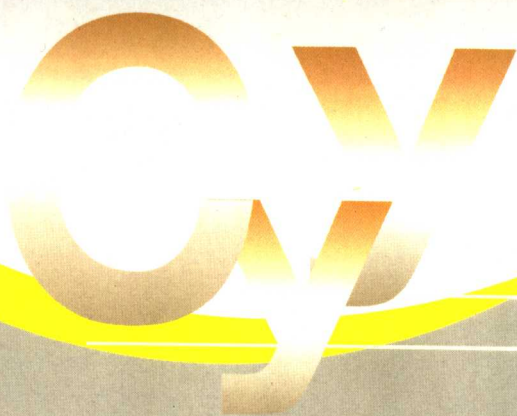


21世纪高职高专规划教材·计算机公共课系列

C 语言

程序设计基础教程

本书主编 潘志安 李学锋



华中师范大学出版社

21 世纪高职高专规划教材·计算机公共课系列

C 语言程序设计基础教程

本书主编	潘志安	李学锋
副主编	张月红	赵世均

华中师范大学出版社

内 容 提 要

C语言是一种生命力很强的程序设计语言,它不仅可以编写应用软件,而且特别适合于编写系统软件。

本书共有10章,包括基本知识、数据类型、运算符与表达式、语句和控制结构、数组、预处理命令、函数、指针、结构体和共用体、文件。

本书是作者根据多年的教学经验编写而成的,内容充实、文字流畅、逻辑性强、例题丰富,并对每个例题中的关键地方配以注释,增加了程序的可读性,便于学习。

本书可作为高职高专计算机专业的教材使用,也可作为一般技术人员的自学教材。

新出图证(鄂)字10号

图书在版编目(CIP)数据

C语言程序设计基础教程/潘志安,李学锋主编.

—武汉:华中师范大学出版社,2006.8

ISBN 7-5622-2952-X/TP·36

I. C… II. ①潘… ②李… III. C语言—程序设计—高等学校:技术学校—教材
IV. TP312

中国版本图书馆CIP数据核字(2004)第050166号

书 名: C语言程序设计基础教程

本书主编: 潘志安 李学锋

选题策划: 华中师范大学出版社第二编辑室 电话: 027-67867362

出版发行: 华中师范大学出版社©

地 址: 武汉市武昌珞瑜路100号 邮编: 430079

发行部电话: 027-67863040 67867371 67861549 67867076

邮购部电话: 027-67861321 传真: 027-67869329

网址: <http://www.ccnp.com.cn> 电子信箱: hscbs@public.wh.hb.cn

经 销: 新华书店湖北发行所

印 刷 者: 华中师范大学印刷厂印刷

责任编辑: 裴媛媛

封面设计: 新视点

责任校对: 张 钟

督 印: 姜勇华

开本/规格: 787mm×1092mm 1/16

字 数: 348千字 印张: 14

版次/印次: 2004年8月第1版 2006年8月第2次印刷

印 数: 10 001—13 500

定 价: 23.00元

敬告读者: 欢迎举报盗版, 请打举报电话 027-67861321。

本书如有印装质量问题, 可向承印厂调换。

21 世纪高职高专规划教材·计算机公共课系列

编审委员会

主 任	孙俊逸			
副主任	程时兴	郭江平	张怀中	
委 员	刘 义	项喜章	张业明	彭 萍
	管火明	张心宇	柴雅静	熊仕涛
	张 谨	朱爱群	胡新和	熊 林
	龚伏廷	李学锋	潘志安	林锦章
	汪 洋	金晓初	赵世均	张月红
	王青云	雷晓云	王广胜	

21 世纪高职高专规划教材·计算机公共课系列

C 语言程序设计基础教程

本书主编 潘志安 李学锋

副主编 张月红 赵世均

编者 (以姓氏笔画为序)

刘 娟 (深圳市技师学院)

李 军 (湖北工业大学)

汪 洋 (武汉船舶职业技术学院)

周 丰 (武汉交通职业学院)

林锦章 (湖北职业技术学院)

曾 志 (咸宁职业技术学院)

蔡国辉 (湖北生物科技职业学院)

潘志安 (湖北职业技术学院)

前 言

近几年来,高等职业技术教育飞速发展,预计到2005年,高等职业技术学院的在校人数将占整个高等院校在校人数的50%,其规模与影响越来越大。但随之而来的师资与教材问题也凸现出来,特别是教材问题,主要表现为与实际教学脱节,大多借用大学本科教材,与职业技术学院的性质与培养目标不适应。因此,编写与高等职业技术学院培养目标的认知水平相适应的教材,是高等职业技术学院教育改革的重要内容。

为了促进和规范高等职业技术学院的教材建设,构建符合自身特点和规律的教材体系,我们多次召开由省内外职业技术教育的专家和多所示范性高等职业技术学院的领导、第一线的教师参加的教材建设与编写研讨会,提出了高职高专教材编写的指导思想和整体规划。在总结教学经验,分析教材现状的基础上,确立了“**基础够用,重在实践,应用与应试相结合**”的编写原则,并先行推出高职高专计算机公共课系列教材,在取得经验的基础上,再逐步将高职高专各科教材系列化。

本系列教材具有以下特点:

1. 紧扣教学大纲与培养目标,准确定位,突出课程特点。针对高等职业技术教学的特点,既注重基本概念、基础理论、基本技术与方法的讲解,适用于学生在教师指导下自主学习的教学模式,又能反映计算机技术的新发展,并将主讲教材与上机实践教材配套,内容布局分工合理。

2. 体系得当,注重实际应用与动手能力的培养。在教材体系的构建、内容的取舍方面特别强调实践教学环节,注重培养学生的上机操作能力、解决问题的能力 and 知识综合应用的能力。

3. 精雕细琢,强调实用性,实现理论与实践的紧密结合。本系列教材的作者都是省内外示范性高等职业院校的“双行型”骨干教师,他们长期工作在教学第一线,具有丰富的教学、研究和实践经验,参加过多种相关教材的编写工作,使本系列教材的针对性和实用性较强,理论与实践结合紧密。

4. 较好地把握了应试与应用的相互关系。通过考试,取得学科等级证书,获得某种职业的从业资格,已成为合格大学毕业生的条件之一,同时也是用人单位录用人才的标准之一。该系列教材在注重理论与实践相结合,把握高职高专教材特色的基础上,较好地解决了应用与应试的关系,将通过计算机二级考试作为教材编写的重要目标之一,配备了大量的模拟习题和实例,学完本系列教材,通过计算机二级考试应不是难事。

总之,本系列教材内容丰富,针对性强,是为高职高专度身定制的、符合高职高专培养目标和教学实际的优秀教材。

《C 语言程序设计基础教程》第 1、2 章由蔡国辉（湖北生物科技职业学院）编写，第 3、6 章由汪洋（武汉船舶职业技术学院）编写，第 4 章、附录由潘志安（湖北职业技术学院）编写，第 5 章由刘娟（深圳市技师学院）编写，第 7 章由周丰（武汉交通职业学院）编写，第 8 章由李军（湖北工业大学）编写，第 9 章由曾志（咸宁职业技术学院）编写，第 10 章由林锦章（湖北职业技术学院）编写。襄樊职业技术学院张月红、赵世均参加了全书编写及书稿整理工作。全书由潘志安、李学锋任主编，负责总体设计及最后的修改定稿。

教材的编写是一项复杂的系统工程，它与教学体制、培养目标、学生的认知水平、师资的强弱密切相关，同时也与社会的政治、经济、文化等各方面密切相关，因而一部好的教材必须经过实践的检验，在使用过程中不断改进，逐步完善。高等职业技术教育是近几年快速发展起来的办学模式，相关教材的编写处于探索阶段，因此，本系列教材不可避免地存在着不足之处，恳切希望广大读者给予指正。

丛书编委会

2004 年 8 月

目 录

第 1 章 C 语言基本知识	1
1.1 C 语言概述	1
1.2 C 语言程序的基本结构	2
1.3 C 语言的基本语法单位	5
1.3.1 标识符	5
1.3.2 关键字	5
1.3.3 分隔符	6
1.4 C 语言程序的上机步骤	6
1.4.1 运行一个 C 语言程序的基本过程	6
1.4.2 在 Turbo C 运行 C 语言程序的步骤	6
1.4.3 在 Visual C++6.0 运行 C 语言程序的步骤	11
习题	15
第 2 章 数据类型	16
2.1 C 语言的数据类型	16
2.1.1 C 语言的数据类型分类	16
2.1.2 基本数据类型的名字及长度	16
2.2 常量和变量	18
2.2.1 常量	18
2.2.2 变量	19
2.3 整型数据	20
2.3.1 整型数据的内存表示	20
2.3.2 整型常量	21
2.3.3 整型变量	22
2.4 实型数据	23
2.4.1 实型数据的内存表示	23
2.4.2 实型常量	23
2.4.3 实型变量	24
2.5 字符型数据	25
2.5.1 字符常量	25
2.5.2 字符变量	27
2.5.3 字符数据的内存表示	27
2.5.4 字符串常量	29
习题	30
第 3 章 运算符与表达式	32
3.1 算术运算符与算术表达式、强制类型转换运算符	32

3.1.1	C 语言运算符的介绍	32
3.1.2	基本的算术运算符	32
3.1.3	运算符的优先级别、结合性	33
3.1.4	算术表达式	33
3.1.5	数值型数据混合运算的类型转换规则	33
3.1.6	强制类型转换运算符	34
3.1.7	自加运算符和自减运算符	35
3.2	赋值运算符与赋值表达式	36
3.2.1	赋值运算符和赋值表达式	36
3.2.2	复合的赋值运算符	37
3.2.3	赋值运算的类型转换及规则	37
3.3	逗号运算符与逗号表达式	38
3.4	关系运算和逻辑运算	39
3.4.1	C 语言中的逻辑值	39
3.4.2	关系运算符和关系表达式	39
3.4.3	逻辑运算符和逻辑表达式	40
3.4.4	条件运算符和条件表达式	41
3.5	长度运算符和长度运算表达式	43
3.6	位运算	43
3.6.1	位运算符	43
3.6.2	位运算符的运算功能	44
习题		46
第 4 章	语句和程序结构	48
4.1	C 语言语句	48
4.2	顺序结构	50
4.2.1	赋值语句	50
4.2.2	数据的输入输出	50
4.2.3	putchar 函数 (字符输出函数)	51
4.2.4	getchar 函数 (字符输入函数)	51
4.2.5	printf 函数 (格式输出函数)	51
4.2.6	scanf 函数 (格式输入函数)	54
4.3	选择结构	56
4.3.1	if 语句	56
4.3.2	switch 语句	59
4.4	循环结构	62
4.4.1	while 语句和 do-while 语句	62
4.4.2	for 语句	64
4.4.3	循环的嵌套	66
4.4.4	几种循环的比较	66
4.5	其他控制语句	66
4.5.1	break 语句	66

4.5.2	continue 语句	67
4.5.3	return 语句	67
4.5.4	goto 语句	68
4.6	程序举例	68
	习题	70
第 5 章	数组	72
5.1	一维数组	72
5.1.1	一维数组的定义和初始化	72
5.1.2	一维数组的引用	74
5.1.3	一维数组应用举例	74
5.2	二维数组	77
5.2.1	二维数组的定义和初始化	77
5.2.2	二维数组的引用	79
5.2.3	二维数组应用举例	80
5.3	字符数组	81
5.3.1	字符数组的定义和初始化	81
5.3.2	字符数组的引用	83
5.3.3	字符串及字符串结束标志	83
5.3.4	字符数组的输入输出	84
5.3.5	字符数组应用举例	86
5.4	字符串处理函数	86
	习题	90
第 6 章	编译预处理	92
6.1	预处理概述	92
6.2	宏定义	92
6.2.1	不带参数的宏定义	92
6.2.2	带参数的宏定义	94
6.2.3	终止宏定义	95
6.3	文件包含	96
6.4	条件编译	97
	习题	99
第 7 章	函数	101
7.1	函数概述	101
7.2	函数的定义和声明	102
7.2.1	函数的定义	103
7.2.2	函数声明和函数原型	104
7.3	函数的调用、函数参数和返回值	105
7.3.1	函数的形参和实参	106
7.3.2	函数的返回值	108
7.3.3	函数的调用	110

7.4 嵌套和递归调用	113
7.4.1 函数的嵌套调用	113
7.4.2 函数的递归调用	114
7.5 数组作函数参数	117
7.5.1 数组元素作函数参数	117
7.5.2 数组名作函数参数	118
7.6 变量的作用域	120
7.6.1 变量的作用域规则	120
7.6.2 局部变量和全局变量	121
7.7 变量的存储类别	124
7.7.1 自动存储变量	124
7.7.2 寄存器存储变量	125
7.7.3 外部存储变量	126
7.7.4 静态存储变量	127
7.8 内部函数和外部函数	130
7.8.1 内部函数	130
7.8.2 外部函数	131
7.8.3 多文件程序	133
习题	133
第8章 指针	135
8.1 地址与指针	135
8.1.1 指针的概念	135
8.1.2 指针变量的定义	136
8.1.3 指针变量的初始化和赋值	136
8.1.4 指针变量的引用	137
8.1.5 指针变量的运算	138
8.1.6 指针变量作函数参数	141
8.2 指针与数组	144
8.2.1 指针与一维数组	144
8.2.2 指针与二维数组	147
8.2.3 指向数组的指针作函数参数	151
8.3 指针与字符串	154
8.3.1 字符串的表示和使用	154
8.3.2 指向字符串的指针作函数的参数	156
8.4 指针与函数	157
8.4.1 指向函数的指针变量的定义与引用	157
8.4.2 指向函数的指针变量作函数参数	159
8.4.3 返回指针值的函数	159
8.5 指针数组和指向指针的指针	160
8.5.1 指针数组的概念、定义和用途	160
8.5.2 指针数组作 main 函数的参数	161
8.5.3 指针的指针	163

8.5.4 void 指针类型	165
习题	166
第9章 结构体与共用体	167
9.1 结构体类型声明、结构体类型变量定义和引用	167
9.1.1 结构体类型的定义	167
9.1.2 结构体类型变量的定义、引用和初始化	168
9.2 结构体数组和指向结构体类型数据的指针	172
9.2.1 结构体数组	172
9.2.2 指向结构体类型数据的指针	174
9.3 链表	177
9.3.1 链表概述	177
9.3.2 动态内存分配处理函数	178
9.3.3 动态链表	179
9.4 共用体	186
9.4.1 共用体的概念和共用体类型的定义	186
9.4.2 共用体变量定义和引用	187
9.4.3 共用体类型数据的特点	188
9.5 枚举类型	191
9.6 用 typedef 定义类型	192
9.6.1 用 typedef 规定新类型的方法	193
9.6.2 用 typedef 规定新类型的作用	193
9.6.3 使用 typedef 的说明	194
习题	194
第10章 文件	195
10.1 文件概述	195
10.2 文件类型指针和文件号	196
10.3 文件操作	196
10.3.1 文件的打开与关闭	196
10.3.2 文件的读写	197
10.3.3 文件的定位	200
习题	202
附录	
附录1 常用字符与 ASCII 代码对照表	203
附录2 C 语言中的关键字	204
附录3 运算符和结合性	204
附录4 C 语言库函数	206
参考文献	212

第 1 章 C 语言基本知识

C 语言是一种通用计算机程序设计语言。它适用于编写各种系统软件,也适用于编写各种应用软件。随着计算机的迅速发展和广泛应用,在计算机上推出的各种版本的 C 语言系统和工具(例如 Turbo C、C、C++、Visual C 等版本),受到越来越多用户的青睐。

本章主要介绍 C 语言的发展、特点、C 语言程序的基本结构、C 语言的基本语法单位以及如何实现 C 语言程序的运行等内容。

1.1 C 语言概述

1. C 语言的发展历史

C 语言是 1972 年由美国 AT&T 公司贝尔实验室的 Dennis. M. Ritchie 设计发明的,当时的 C 语言附属于 UNIX 操作系统。1978 年, Brian W. Kernighan 和 Dennis. M. Ritchie 以 UNIX 第 7 版中的 C 语言编译程序为基础写了经典著作《The C Programming Language》,这本书是以后介绍 C 语言的各种书籍的蓝本。1978 年以后, C 语言先后移植到各种计算机系统上,且不再依赖于 UNIX 操作系统而独立存在。1988 年,美国国家标准协会(ANSI)在综合各种 C 语言版本的基础上制定了 C 语言文本标准,称为 ANSI C 标准。ANSI C 实现了 C 语言的规范化和统一化, Brian W. Kernighan 和 Dennis. M. Ritchie 按照 ANSI C 标准重写了《The C Programming Language》一书,于 1990 年正式发表了《The C Programming Language Second Edition》。1990 年国际标准化组织(ISO)公布了以 ANSI C 为基础制定的 C 语言的国际标准 ISO C,人们通常称之为标准 C。1988 年以后推出的各种 C 语言版本对标准 C 是兼容的。

2. C 语言的特点

C 语言之所以成为目前世界上使用最广泛的程序设计语言,并被选作适应软件工程需要而发展起来的面向对象的程序设计语言 C++ 的基础语言,是由于 C 语言的诸多突出优点所决定的。

(1) 语言简洁、紧凑,使用方便、灵活。C 语言共有 32 个关键字,9 种控制语句。程序书写形式自由,主要用小写字母表示,压缩了一切不必要的成分。

(2) 运算符丰富。C 语言的运算符包含的范围很广泛,共有 34 种运算符。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) 数据结构丰富,具有现代化语言的各种数据结构。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。能用来实现各种复杂的数据结构(如链表、树、栈等)的运算,尤其是指针类型数据,使用起来更为灵活、多样。

(4) 具有结构化的控制语句(如 if... else 语句、while 语句、do... while 语句、switch 语句、for 语句)。用函数作为程序的模块单位,便于实现程序的模块化。C 语言是良好的结构化语言,符合现代编程风格的要求。

(5) 语法限制不太严格,程序设计自由度大。例如对数组下标越界不做检查,由程序编写者自己保证程序的正确。对变量的类型使用比较灵活,例如整型数据与字符型数据可以通用。

一般的高级语言语法检查比较严,能检查出几乎所有的语法错误。而C语言允许程序编写者有较大的自由度,因此放宽了语法检查。

(6) C语言能进行位(bit)操作,能实现汇编语言的大部分功能,可以直接对硬件进行操作。因此,C语言既具有高级语言的功能,又具有低级语言的许多功能,可用来编写系统软件。C语言的这种双重性,使它既是成功的系统描述语言,又是通用的程序设计语言。

(7) 生成目标代码质量高,程序执行效率高。一般只比汇编程序生成的目标代码效率低10%~20%。

(8) 用C语言编写的程序可移植性好(与汇编语言比)。基本上不做修改就能用于各种型号的计算机和各种操作系统。

C语言的以上特点,读者现在也许还不能深刻理解,待学完C语言以后再回顾一下,就会有比较深的体会。

1.2 C语言程序的基本结构

1. 一个具有代表性的简单C语言程序

【例1.1】 假设已知两个正整数m和n的值,求m与n之和是多少?

```
#include<stdio. h>
void main()
{   int m,n,o;
    scanf("%d%d",&m,&n);
    o=m+n;
    printf("%d",o);
}
```

当运行该程序后,在屏幕上出现一个闪烁的光标,要求用户从键盘上输入m和n的值,当键入如下信息:

123 118 ↵

其中,123和118是赋给m和n的值;空格表示,作为123和118的分隔符;↵表示回车键。(当然,也可以输入一个123按一个回车键,再输入一个118再按一个回车键。)

屏幕上接着显示如下信息:

241

这就是m与n之和,是该程序运行后的结果。

2. 程序的简单分析

为了分析方便,我们将例1.1的程序增加了几个注释,请看经过注释后的例1.1。

```
#include <stdio. h>
void main()                /* 第1行:主函数 */
{                           /* 第2行:主函数体的开始,用花括号表示{ */
    int m,n,o;              /* 第3行:定义变量类型语句 */
    scanf("%d%d",&m,&n); /* 第4行:数据输入语句 */
    o=m+n;                  /* 第5行:赋值语句 */
    printf("%d",o);         /* 第6行:数据输出语句 */
}                           /* 第7行:主函数体的结束,用花括号表示} */
```

下面就来分析该程序。

(1) 该程序作为一个文件存放在磁盘上,其扩展名为.C(系统默认)。

(2) 程序的第1行中 main()表示主函数,每一个C语言程序都必须有一个 main()函数。函数体由一对花括号{ }括起来。

(3) 程序的第3行是定义变量 m,n 和 o 类型的语句。“int m,n,o”表示定义 m、n、o 这3个变量为整型类型。语句与语句之间用分号“;”分隔。

(4) 程序的第4行是数据输入语句,其中“%d”表示输入的数据类型为十进制整数类型。“&m,&n”中的“&”表示“取地址”。整个语句的作用就是,接受用户从键盘上输入的2个整数类型数值,分别输入到变量 m 和 n 的地址所标志的存储单元中。假设,从键盘上输入 123 和 118 两个整型数值,存储单元可以看成是一个盒子,盒子的名称分别为 m 和 n,盒子中分别存放 123 和 118 两个数值。

(5) 程序的第5行是赋值语句,其中“=”表示赋值号,赋值号的左边是变量 o,右边是表达式 m+n。语句的作用就是将右边表达式的值赋给左边的变量 o,也就是执行了赋值语句后,在 o 变量的存储单元中,存放的数值为 241,该数值的类型也是整型类型。

(6) 程序的第6行是输出语句,其中“%d”表示输出数值的类型为十进制整数类型,该语句的作用是输出变量 o 的值。

(7) 程序中的“/* */”表示注释符号,在注释符号中间是注释内容,该内容可以由任何的字符构成,系统不执行注释内容。注释的作用是给程序员阅读程序带来方便。而且程序中做注释(文档),这是优秀程序员必备的良好习惯。

3. C语言程序结构

通过以上的分析,可以看出C语言程序的基本结构。

(1) C语言程序是由函数构成的

C语言程序是由函数构成的,一个C语言源程序至少包含一个 main()主函数,也可以包含一个主函数和若干个其他函数。例 1.1 源程序是由一个主函数构成的。下面再来看一个程序,该程序包含2个函数,一个是主函数,另一个是自定义函数。

【例 1.2】 假设已知2个正整数 m 和 n 的值,比较它们的大小,并输出其中的最大值。

```
#include <stdio.h>
int max(int x,int y);
void main()           /* 主函数 */
{
    int m,n,o;          /* 定义变量 */
    scanf("%d%d",&m,&n); /* 输入变量 m 和 n 的值 */
    o=max(m,n);         /* 调用 max 函数,并将得到的值赋给变量 o */
    printf("max=%d",o); /* 输出 o 的值 */
}
int max(int x,int y)    /* 定义 max 函数,函数值为整型,x,y 为形式参数 */
{
    int z;              /* 定义变量 z 类型 */
    if(x>y)
        z=x;           /* 比较 x,y 的大小,如果 x 大则赋给变量 z */
}
```

```

else z=y; /* 否则 y 赋给变量 z */
return(z); /* 返回 z 的值,通过 max 带回调用处 */
}

```

当运行该程序并输入 m 和 n 数值时,假设分别输入 123 和 118 后,在屏幕上输出以下信息:

```
max=123
```

从以上程序来看,该程序是由两个函数构成,一个是 main() 主函数,另一个是 max(int x, int y) 自定义函数。在主函数中调用自定义函数,并将实型参数 m 和 n 的值分别传递给 x 和 y,并执行比较,结果最大值是 123 并将该值返回去,程序结果是 max=123。

可以说,C 语言是函数式的语言。程序全部工作都是由函数来完成的。C 语言的函数库十分丰富,标准 C 提供了一百多个库函数,Turbo C 和 MS C 4.0 提供了三百多个库函数,Visual C 提供的函数库就更多了。

(2) 函数的组成

函数是一个独立的程序块,相互不能嵌套。main 函数以外的其他任何函数只能由 main 函数或其他函数调用;自己不能单独运行。

一个函数由两部分组成:函数头部和函数体。函数头部包括函数返回值的类型、函数名和参数表(函数头部的末尾不能加分号)。参数表可以为空,参数表为空时用类型名 void 表示(void 可以缺省)。函数体包括说明部分(局部说明)和语句部分,可以没有说明部分,也可以没有语句部分。说明部分和语句部分都为空的函数称为哑函数,例如 int max(int x, int y) { }, max 是一个哑函数。哑函数是一个最小合法函数,调用一个哑函数在功能上不执行任何操作,但在调试由多个函数组成的大程序方面很有用处。

(3) C 语言标准函数

C 语言函数分为两类:标准函数和用户定义函数。用户定义函数是由程序员在自己的源程序中编写的函数,例如例 1.2 中的 max 函数。标准函数是由 C 语言编译程序提供的一些通用函数,这些函数以编译后的目标代码形式集中存放在称为 C 语言标准函数库(简称 C 语言库)的文件中,C 语言标准函数又称为 C 语言库函数。例如,scanf 和 printf 函数都是 C 语言库函数。

用户程序需要使用标准函数时,只需在使用前用 #include 包含该标准函数所需的系统头文件,例如 scanf 函数和 printf 函数的头文件为 stdio.h,然后按规定的格式调用所需标准函数即可。系统头文件中包含了标准函数的相应说明(函数原型)、有关的类型定义及常量定义。

(4) C 语言程序的书写格式

C 语言本身对书写格式要求很宽松,它的书写格式非常自由。但是,由于 C 语言语句比较简洁精练,它的易读性较差,这就要求在书写格式上按照一定规格,增加其易读性。C 语言的书写格式,读者可以从后面的程序中逐渐学会,在这里我们仅做一般性的简单介绍。

① C 语言中一行内可以写一个语句,也可以写几个语句。一个语句可以在一行内完成,也可以分成几行完成,C 语言程序中没有行号。

② C 语言中花括号“{ }”用得比较多,一般情况下,左右花括号各占一行,并且需上下对齐,这样便于检查花括号的成对性。

③ 在 C 语言程序中运用缩格写法对增加易读性十分重要,有时由于缩格运用得不好,反而将程序理解错了,缩格写法需在实践中学习。例如,一个函数的函数体要比定义函数的花括号缩 2 格;一个循环语句的循环体或条件语句的 if 体、else 体等都要做缩格处理。这样做有利于程序的阅读和查错,特别是在编写实际应用的程序时尤为重要。

④ 一般C语言程序使用小写字母来书写程序,特别是函数名必须使用小写字母来表示,C语言程序中的大写字母一般表示常量。

1.3 C语言的基本语法单位

基本语法单位是指具有独立语法意义的最小语法成分。在C语言中,基本语法单位被称为单词,组成单词的基本符号是字符,标准C语言及大多数C语言编译程序使用的字符集是ASCII字符集。

C语言的单词分为六类:标识符、关键字、常量、字符串、运算符及分隔符。本节介绍标识符、关键符及分隔符。

1.3.1 标识符

1. 标识符的含义

C语言中,标识符是指程序中的常量、变量、数据类型和函数的名字。例如例1.2中的变量名m、n、o、形参x,y以及函数名max都是标识符,主函数名main和标准函数的名字(例如scanf,printf)也是标识符。数据类型名int和void不是标识符而是关键字。

2. 标识符的组成规则

(1) 标识符由字母(A~Z,a~z)和数字(0~9)组成,必须以字母开头,随后可跟0个或多个字母或数字(即标识符是以字母开头的字母、数字串);

(2) 字母要区分大小写,例如A和a,Ab和ab等都是不同的标识符;

(3) 下划线(_)被作为一个字母看待。

在使用标识符时,习惯上变量名和函数名用小写,常量名和用typedef定义的数据类型名用大写。此外,为便于阅读和记忆,应选用能够表达含义的英文单词、英文单词的一部分或缩写作为标识符,例如day, date, year, sum, flag, lines count, HEIGHT_START, HEIGHT_END等。

下面是一些合法的标识符例子:

```
x xl i name _ab a_b ab_ A PI
```

下面的表示均不是标识符:

4ab 不是以字母开头(非法表示)

order.no 小数点(.)不是字母也不是数字

p→x →不是字母也不是数字

3. 标识符的有效长度

在组成标识符的字符中,能够被编译程序识别的那一部分字符的数目称为标识符的有效长度。这就是说,程序员可以写一个很长的标识符,但在有效长度以内的字符才是有意义的字符。标准C语言规定标识符的有效长度为前31个字符。实际应用中为便于记忆和书写,在能够区别于其他标识符及能够表达一定含义的前提下,标识符应尽可能简短一些。

注意:标识符不能与关键字同名。

1.3.2 关键字

关键字由固定的小写字母组成,是由系统预先定义好的名字,用于表示C语言的语句、数据类型、存储类型或运算符。用户不能用它们来作为自己定义的常量、变量、数据类型或函数的名字。关键字又称为保留字,即被系统保留作为专门用途的名字。

标准C语言定义的32个关键字见附录2。