

普通高校本科计算机专业特色教材精选·算法与程序设计

程序设计基础（C语言版）

徐正芹 等 编著

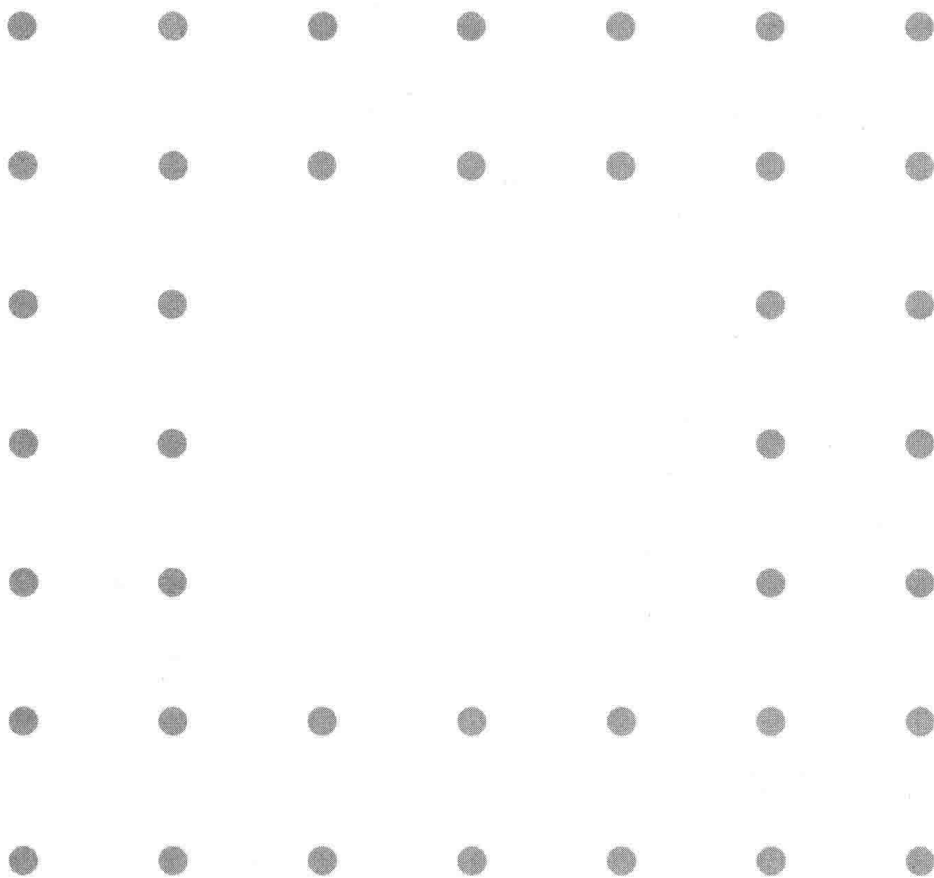
清华大学出版社



普通高校本科计算机专业特色教材精选·算法与程序设计

程序设计基础 (C语言版)

徐正芹 赵文栋 彭来献 李艾静 王海 编著



清华大学出版社
北京

内 容 简 介

本书根据初学者的认知特点,紧贴教学、循序渐进、由浅入深地讲解了C语言的知识。

全书共分为两个部分:第一部分是C语言程序设计,共11章,系统地对C语言的基本语法和基本技巧进行介绍;第二部分是上机实验,内容包括编程思路、设计方法、程序代码、调试过程等,用于提高学生的实际程序设计能力。本书以C程序实例作为引导,通过典型例题对重点知识进行强化,符合程序设计的规律。本书的例题和上机综合实验全部在Visual C++ 6.0环境下调试通过。

本书适合作为高等学校理工类专业程序设计基础课程教材,也可供程序设计初学者自学参考。

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

版权所有,侵权必究。侵权举报电话:010-62782989 13701121933

图书在版编目(CIP)数据

程序设计基础(C语言版)/徐正芹等编著. —北京:清华大学出版社,2019

(普通高校本科计算机专业特色教材精选·算法与程序设计)

ISBN 978-7-302-52194-5

I. ①程… II. ①徐… III. ①C语言—程序设计—高等学校—教材 IV. ①TP312.8

中国版本图书馆CIP数据核字(2019)第013072号

责任编辑:袁勤勇 战晓雷

封面设计:傅瑞学

责任校对:胡伟民

责任印制:沈 露

出版发行:清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址:北京清华大学学研大厦A座 邮 编:100084

社总机:010-62770175 邮 购:010-62786544

投稿与读者服务:010-62776969, c-service@tup.tsinghua.edu.cn

质量反馈:010-62772015, zhiliang@tup.tsinghua.edu.cn

课件下载: <http://www.tup.com.cn>, 010-62795954

印 装 者:三河市龙大印装有限公司

经 销:全国新华书店

开 本:185mm×260mm

印 张:17

字 数:420千字

版 次:2019年5月第1版

印 次:2019年5月第1次印刷

定 价:39.00元

产品编号:081358-01

出版说明

INTRODUCTION

在我国高等教育逐步实现大众化后，越来越多的高等学校将会面向国民经济发展的第一线，为行业、企业培养各级各类高级应用型专门人才。为此，教育部已经启动了“高等学校教学质量和教学改革工程”，强调要以信息技术为手段，深化教学改革和人才培养模式改革。如何根据社会的实际需要，根据各行各业的具体人才需求，培养具有特色显著的人才，是我们共同面临的重大问题。具体地，培养具有一定专业特色的和特定能力强的计算机专业应用型人才则是计算机教育要解决的问题。

为了适应 21 世纪人才培养的需要，培养具有特色的计算机人才，急需一批适合各种人才培养特点的计算机专业教材。目前，一些高校在计算机专业教学和教材改革方面已经做了大量工作，许多教师在计算机专业教学和科研方面已经积累了许多宝贵经验。将他们的教研成果转化为教材的形式，向全国其他学校推广，对于深化我国高等学校的教学改革是一件十分有意义的事。

清华大学出版社在经过大量调查研究的基础上，决定编写出版一套“普通高校本科计算机专业特色教材精选”。本套教材是针对当前高等教育改革的新形势，以社会对人才的需求为导向，主要以培养应用型计算机人才为目标，立足课程改革和教材创新，广泛吸纳全国各地的高等院校计算机优秀教师参与编写，从中精选出版确实反映计算机专业教学方向的特色教材，供普通高等院校计算机专业学生使用。

本套教材具有以下特点：

1. 编写目的明确

本套教材是在深入研究各地各学校办学特色的基础上，面向普通高校的计算机专业学生编写的。学生通过本套教材，主要学习计算机科学与技术专业的基本理论和基本知识，接受利用计算机解决实际问题的基本训练，培养研究和开发计算机系统，特别是应用系统的基本能力。

2. 理论知识与实践训练相结合

根据计算学科的三个学科形态及其关系,本套教材力求突出学科理论与实践紧密结合的特征,结合实例讲解理论,使理论来源于实践,又进一步指导实践得到自然的体现,使学生通过实践深化对理论的理解,更重要的是使学生学会理论方法的实际运用。

3. 注意培养学生的动手能力

每种教材都增加了能力训练部分的内容,学生通过学习和练习,能比较熟练地应用计算机知识解决实际问题。既注意培养学生分析问题的能力,也注重培养学生解决问题的能力,以适应新经济时代对人才的需要,满足就业要求。

4. 注重教材的立体化配套

大多数教材都将陆续配套教师用课件、习题及其解答提示,学生上机实验指导等辅助教学资源,有些教材还提供能用于网上下载的文件,以方便教学。

由于各地区各学校的培养目标、教学要求和办学特色均有所不同,所以对特色教学的理解也不尽一致,我们恳切希望大家在使用教材的过程中,及时地给我们提出批评和改进意见,以便我们做好教材的修订改版工作,使其日趋完善。

我们相信经过大家的共同努力,这套教材一定能成为特色鲜明、质量上乘的优秀教材,同时,我们也希望通过本套教材的编写出版,为“高等学校教学质量和教学改革工程”做出贡献。

清华大学出版社

前言

PREFACE

“程序设计基础”是高等学校理工类专业学生的基础课程。本书以 C/C++ 语言作为基本工具，以程序设计思想与方法作为核心内容，以动手编程解决实际问题能力的培养作为最终目标。通过本书，不仅要使学生掌握程序设计语言的语法与结构，更重要的是逐步培养学生用计算机解决问题的思维、习惯与方法。

本书的读者对象是没有编程基础的初学者，他们通过本书所能接触到的也只是一些简单的程序，但程序结构的设计和编程习惯的培养却正是从初学阶段开始形成的，因此本书对概念、算法、语法以及例题的讲解都强调规范化、结构化，引导读者适当地模仿，从开始就养成规范编程的习惯。

本书有以下特色：

(1) 本书以程序设计为核心，将 C 语言的有关语法有机结合到程序设计实例中，避免了生硬枯燥的语法介绍。

(2) 突出针对性。这主要是指“继续学习”的针对性。C 语言的后续课程主要有两类：一类是其他编程语言课程；另一类是后续专业课程，包括“数据结构”“数据库”“算法分析”等。在教学实践中经常出现这种情况：作为这些课程基础知识应该提前掌握的知识，学生在 C 语言的学习中并未认真掌握，如参数的传递方式、指针的使用和内存的动态申请等，需要重新学，给教学带来了很大的不便。本书引入了数据结构中的简单算法和基础内容作为讲解实例或应用实例，使计算机相关专业学生为将来的专业课程学习打好基础，也使非计算机专业的学生有能力进一步拓展知识。

(3) 注重实用性。本书不仅介绍编程的理论和方法，还通过上机实践来提高学生的编程能力。本书以 Visual C++ 6.0 为基本开发环境，用一定的篇幅对 Visual C++ 6.0 的编程和调试环境作了较深入的介绍，既能激发学生的学习兴趣，更能为学生今后学习高级开发技术打下良好的基础。本书的例题和上机实验全部在 Visual C++ 6.0 环境下调试通过。

(4) 强调编程技巧和方法。本书强调结构化程序设计的概念、方法和编程技巧。10个上机实验内容包括问题的提出、编程思路、设计方法、程序代码、调试过程等,以帮助学生提高程序设计能力。

(5) 本书将编程学习过程中容易出错的地方用黑框标出,重点加以说明,使初学者避免了在学习过程中“踩雷”。

本书分为两部分。

第一部分是C语言程序设计,系统地介绍了程序设计的方法及C语言的基本语法和基本技巧,是本书的主体部分,共11章。其中带*的内容为扩展内容,在今后的学习中应用较少,跳过这些部分,不会对其他内容的学习带来影响,可以等将来用到时再回来查阅。各章内容如下:第1章通过几个简单的示例程序,对C语言程序作初步介绍;第2章介绍程序设计方法、程序设计过程、程序设计语言的概念,并简要介绍结构化程序的几种基本控制结构;第3章介绍数据类型、常量与变量、运算符与表达式的概念;第4章介绍常用的输入输出函数和顺序结构程序设计;第5章介绍关系运算及选择结构程序设计;第6章介绍循环结构程序设计;第7章介绍数组的定义、引用、初始化及数组的应用;第8章介绍函数的使用、参数的传递及变量的作用域等内容;第9章介绍指针的概念和运算、指针在函数调用中的作用及指针与数组的关系等内容;第10章介绍结构、联合和枚举等类型;第11章介绍与文件操作相关的内容。

第二部分是上机实验,包括10次实验的内容。实验内容的安排基本与第一部分的各章相对应。一般每次实验安排两三个不同难度的题目,可以根据上机时间有选择地安排其中的部分题目。

本书主要作为高等学校理工类专业的程序设计基础课程教材,也可作为计算机爱好者学习C语言的参考书。本书电子教案可从清华大学出版社网站 www.tup.com.cn 下载。限于作者水平,书中难免会存在一些不足之处,敬请读者批评指正。

作者

2018年9月

目 录

CONTENTS

第 1 章 C 语言初步介绍	1
1.1 几个例子	1
1.2 C 语言的字符集与词汇	4
1.2.1 C 语言的字符集	4
1.2.2 词汇	4
1.3 C 程序的上机操作	5
1.3.1 启动 Visual C++ 6.0	6
1.3.2 集成开发环境介绍	7
1.3.3 一个程序的上机操作过程	8
1.4 习题	14
第 2 章 程序设计过程	17
2.1 程序设计概述	17
2.1.1 结构化程序设计	17
2.1.2 模块化程序设计的方法	18
2.1.3 面向对象的程序设计	18
2.2 程序设计语言	19
2.2.1 程序设计的基本步骤	19
2.2.2 算法及其表示	20
2.2.3 程序设计语言	21
2.3 程序的基本控制结构	23
2.3.1 顺序结构	23
2.3.2 选择结构	23
2.3.3 循环结构	23
2.3.4 子程序调用	23

2.4	应用举例	24
2.5	习题	25
第3章	数据类型、运算符及表达式	27
3.1	数据和数据类型	27
3.2	常量	28
3.2.1	整型常量	28
3.2.2	实型常量	28
3.2.3	字符常量	29
3.2.4	字符串常量	29
3.2.5	符号常量	30
3.3	变量	30
3.3.1	整型变量	30
3.3.2	实型变量	31
3.3.3	字符型变量	31
3.4	赋值运算与算术运算	32
3.4.1	赋值运算	32
3.4.2	二元算术运算	33
3.4.3	自加、自减运算	34
3.4.4	复合赋值运算	35
3.5	变量赋初始值	35
3.6	类型转换	36
3.6.1	自动类型转换	36
3.6.2	强制类型转换	37
3.7	表达式	37
3.7.1	表达式求值	37
3.7.2	表达式语句	39
3.8	位运算	39
3.8.1	按位与运算	40
3.8.2	按位或运算	40
3.8.3	按位异或运算	41
3.8.4	按位取反运算	41
3.8.5	左移运算	42
3.8.6	右移运算	42
3.9	习题	42
第4章	顺序结构程序设计	45
4.1	编译预处理	45

4.1.1	宏定义	45
4.1.2	文件包含	46
4.2	字符输入输出函数	47
4.2.1	字符输入函数 getchar	48
4.2.2	字符输出函数 putchar	48
4.3	格式输入输出函数	49
4.3.1	格式输入函数 scanf	49
4.3.2	格式输出函数 printf	51
4.4	顺序结构程序设计	56
4.4.1	空语句和表达式语句	56
4.4.2	复合语句	56
4.5	应用举例	57
4.6	习题	59
第 5 章	选择结构程序设计	61
5.1	关系运算和逻辑运算	61
5.1.1	关系运算	61
5.1.2	逻辑运算	62
5.2	if 语句	63
5.2.1	if 语句的 3 种形式	63
5.2.2	条件运算	67
5.3	switch 语句	68
5.4	应用举例	70
5.5	习题	72
第 6 章	循环程序设计	75
6.1	while 循环语句	75
6.2	do-while 循环语句	76
6.3	for 循环语句	77
6.4	循环的退出	81
6.4.1	break 语句	81
6.4.2	continue 语句	82
6.4.3	goto 语句	82
6.5	应用举例	84
6.6	习题	87
第 7 章	数组	91
7.1	一维数组	91

7.1.1	一维数组的定义	91
7.1.2	一维数组的使用	92
7.1.3	一维数组的初始化	93
7.2	二维数组	95
7.2.1	二维数组的定义	95
7.2.2	二维数组的初始化	95
7.2.3	二维数组的引用	96
7.3	字符数组和字符串	98
7.3.1	字符数组的定义	98
7.3.2	字符串	99
7.4	应用举例	102
7.5	习题	105
第8章	函数	107
8.1	C程序与函数	107
8.2	函数的定义和说明	109
8.2.1	函数的定义	109
8.2.2	函数的说明	112
8.3	函数的调用	113
8.3.1	函数调用的格式	113
8.3.2	参数的传递	114
8.4	应用举例1	117
8.5	函数的嵌套调用与递归调用	119
8.5.1	函数的嵌套调用	119
8.5.2	函数的递归调用	121
8.6	作用域与存储类	123
8.6.1	作用域	123
8.6.2	存储类	125
8.6.3	内部函数与外部函数	128
8.7	应用举例2	129
8.8	习题	132
第9章	指针	135
9.1	指针概述	135
9.1.1	什么是指针	135
9.1.2	指针变量的定义	136
9.1.3	指针的运算	136
9.2	指针与函数参数	139

9.3	指针与数组	141
9.3.1	一维数组的指针	141
9.3.2	多维数组的指针	146
9.3.3	字符指针与字符串	148
9.4	应用举例	151
9.5	习题	153
第 10 章	结构、联合和枚举	155
10.1	结构的概念	155
10.1.1	结构类型的定义	155
10.1.2	结构变量的说明	156
10.1.3	结构变量的引用	157
10.1.4	结构变量的初始化	158
10.2	结构数组	160
10.3	指向结构的指针	161
10.3.1	指向结构的指针	161
10.3.2	结构指针作函数参数	164
10.4	动态数据结构	165
10.4.1	内存的动态分配和释放函数	166
10.4.2	链表	166
10.5	联合	174
10.5.1	联合的定义	174
10.5.2	联合变量的说明与引用	175
10.6	枚举	176
10.7	类型定义	178
10.7.1	基本数据类型的重新命名	179
10.7.2	用新类型名代替构造类型名	179
10.7.3	用新类型名定义数组和字符串	180
10.8	应用举例	180
10.9	习题	184
第 11 章	文件	187
11.1	C 文件的概念	187
11.1.1	C 语言文件的特点	187
11.1.2	二进制流和字符流	188
11.1.3	缓冲文件系统和非缓冲文件系统	188
11.2	文件指针	189
11.3	文件的打开与关闭	190

11.3.1	文件的打开——fopen 函数	190
11.3.2	文件的关闭——fclose 函数	191
11.4	文件的读写	192
11.4.1	字符读写函数——fputc 和 fgetc	192
11.4.2	字符串读写函数——fputs 和 fgets	194
11.4.3	数据块读写函数——fread 和 fwrite	196
11.4.4	格式化读写函数——fscanf 和 fprintf	198
11.5	文件的定位	199
11.5.1	rewind 函数	199
11.5.2	ftell 函数	200
11.5.3	fseek 函数	200
11.6	输入输出重定向	201
11.7	应用举例	203
11.8	习题	204
第 12 章	上机实验	205
实验 1	Visual C++ 系统与顺序结构程序设计	205
实验 2	选择结构程序设计	206
实验 3	循环结构程序设计	207
实验 4	程序调试	208
实验 5	数组	215
实验 6	函数	216
实验 7	指针	218
实验 8	结构和枚举	219
实验 9	链表	220
实验 10	文件操作	220
附录 A	习题参考答案	223
附录 B	运算符的优先级及其结合性	251
附录 C	标准 ASCII 字符集	253
附录 D	常用的 C 语言库函数	257
	参考文献	260

C 语言初步介绍

学习一种程序设计语言,最有效的途径就是用它来实际编写程序。本章首先通过几个简单的 C 语言程序使学习者对 C 语言有初步的了解;然后介绍 C 语言的字符集和词汇;最后对 C 语言上机操作的相关知识进行简要介绍。

本章重点:掌握 C 语言程序的结构形式和书写规则;正确区分和使用标识符、保留字、运算符和分隔符等。

1.1 几个例子

例 1.1 首先给出一个最简单的 C 语言程序。

```
#include <stdio.h>
void main()          /* 主函数,每个程序都必须有 */
{
    printf("这是一个最简单的 C 程序!\n"); /* 输出语句,以分号结束 */
}
```

运行结果:

这是一个最简单的 c 程序!

上述程序中,/* 和 */中间的内容是注释。注释对 C 程序的执行没有影响。在程序中增加注释是好的编程习惯,可以提高程序的可读性。

#include <stdio.h>是一条编译预处理命令,此句包含了一个标准输入输出的头文件,有了此句,在程序中就可以使用 scanf、printf 等输入输出标准库函数了。

void main()表示主函数,是 C 语言程序必须有的一个函数。由{ }括起来的部分是函数体,它用来规定该函数所要完成的工作。函数体由一系列语句组成,每个语句都以分号结束。

本例主函数的函数体只包含一条语句——printf("这是一个最简单的 C 程序!\n"),它调用系统标准输出函数 printf。双引号内的字符串照原样输出。\\n 表示换行符,即在输出字符串后换行。

例 1.2 用 C 语言编写程序,要求输入圆的半径值,计算圆面积并输出。

```
#include <stdio.h>
#define PI 3.14159
void main()                                /* 计算圆面积 */
{
    int r;
    float s;
    printf("请输入圆的半径值:");
    scanf("%d",&r);                        /* 输入圆的半径值 */
    s=PI * r * r;
    printf("该圆面积=%f\n",s);           /* 输出圆面积值 */
}
```

本例第 2 行也是编译预处理命令,作用是在编译前将程序中所有的 PI 都用 3.14159 替换。在 C 语言程序中,凡是前面带 # 号的行都是编译预处理命令。

本例主函数的函数体由几条语句组成。函数体中第 1、2 行为变量说明,分别说明 r 为整形变量,s 为实型变量。第 3 行为 printf 函数调用,在屏幕上显示“请输入圆的半径值:”,用来提示用户输入数据。第 4 行为 scanf 函数调用,它的作用是等待用户从键盘上输入数据,并将输入的数据存放到变量 r 中。第 5 行是赋值语句,用来计算=右边表达式的值,即圆面积的值,并赋予变量 s。最后一行还是 printf 函数调用,但在输出时会用变量 s 的值替换字符串中的%f。

程序的运行结果如下:

```
请输入圆的半径值:1
该圆面积=3.141590
```

例 1.3 编写一个程序,要求输入两个数,求其中较大的数。

```
#include <stdio.h>
int max(int x,int y)                        /* 被调用函数 */
{
    int z;
    if(x>y) z=x;
    else z=y;
    return(z);                             /* 将 z 的值返回调用处 */
}

void main()                                /* 主函数 */
{
    int a,b,c;
    scanf("%d,%d",&a,&b);                  /* 输入两个整数 */
    c=max(a,b);                             /* 调用 max 函数 */
    printf("max=%d\n",c);
}
```

本程序包含两个函数：主函数（也是主调函数）main 和被调用函数 max。max 函数的作用是将 x、y 中较大的数的值赋予 z，并通过 return 语句将 z 的值返回到 main 中的调用处。x、y 是两个形式参数，它们的值由主调函数中对应的实际参数 a、b 的值传过来。

main 函数体中第 3 行的功能是：调用 max 函数。调用时将实际参数 a 和 b 的值分别传给 max 函数中对应的形式参数 x 和 y，在执行 max 函数之后，得到一个返回值，即 max 函数中变量 z 的值，把这个值赋予变量 c。

程序的最后是输出 c 的值，运行结果如下：

```
4,6          (从键盘上输入 4,6, 然后按回车键)
max=6        (输出 c 的值)
```

本例中用到函数定义、函数调用、形式参数和实际参数等概念，在后面有关章节中将作详细讨论，这里仅仅是使读者对 C 语言程序的组成和形式有一个初步的了解。

通过上面几个例子，可以看出：

(1) 一个 C 语言程序可以由多个函数组成，但必须包含且只能包含一个主函数 main。函数是组成 C 语言程序的基本单位。

(2) 一个函数由函数头和函数体两部分组成。

函数头定义了函数名、函数类型、函数形式参数名及其类型。例如上面例子中 max 函数的函数头为

```
int max(int x,int y)
```

它定义了函数名为 max，函数类型为 int（整型），函数形式参数为 x 和 y，形式参数的类型为 int。

这里有必要说明一下，早期的 C 语言版本与 ANSI C 在函数定义上有区别。如上例，旧版本写法为

```
int max(x,y)
int x,y;
```

其中，参数在小括号内命名，参数类型在函数体左大括号前说明。本书后面的示例中都采用 ANSI C 的规定。

函数体即函数头下面大括号 {} 内的部分。函数体定义了该函数所要实现的功能。函数体一般包括以下两部分：

① 变量说明。如例 1.3 的 main 函数中的“int a,b,c”。

② 执行部分。由若干条语句组成，每个语句都以分号结束。

(3) 一个 C 程序中函数出现的次序可以是任意的，但总是从 main 函数开始执行。

(4) 前面带有 # 的语句，如 #include、#define 等，都是编译预处理命令。

(5) C 语言程序书写格式比较自由，一行可有多个语句，一个语句也可写成多行（但不能将一个单词分开）。为了提高程序的可读性，往往是一行写一个语句，并以缩进写法来体现语句的层次。

1.2 C语言的字符集与词汇

一个C语言程序好比一篇英语文章,它的各种语言成分,如表达式、语句等,都是由特定字符集中的一些基本字符和词汇按照严密的语法规则构成的。这些基本字符和词汇是语言最基本的语法单位。

1.2.1 C语言的字符集

各种程序设计语言都规定了允许使用的字符(一般称为合法字符),以便处理系统能正确识别它们。C语言中使用的合法字符如下:

(1) 字母和数字,包括小写字母 a~z、大写字母 A~Z、数字 0~9。

(2) 特殊字符,包括以下字符:

+、=、-、_、()、*、#、&、%、\$、!、|、<>、^、.、,、;、:、“ ’、/、\、?、{}、[]、~。

(3) 空白字符,包括空格、换行符和制表符。

以上字符的集合就是C语言的字符集。C语言程序所用的全部字符都在这个字符集中(但在字符串中可使用任意字符)。

1.2.2 词汇

单词是由字符组成的,单词的集合称为词汇。C语言的词汇包括标识符、保留字、运算符和分隔符等。

1. 标识符

C语言的标识符是由字母、数字和下画线“_”组成的字符序列。它的第一个字符必须是字母或下画线。根据上述规则,下面的标识符是合法的:

```
a x1 _fout b4c High _x_y
```

而下面则是不合法的标识符:

```
5a #mod β b+c .obj
```

在C语言的标识符中,大写字母和小写字母是有区别的,例如ABC、Abc和abc是3个不同的标识符。这一点与有些高级语言不同,使用时应注意。

对于标识符的长度,C语言本身未作限制,取决于编译系统。一般编译系统规定前8个字符有效,如果长于8个字符,多余的将不被识别。这样,只要两个标识符前8个字符相同,系统就认为它们是同一个标识符。有的编译系统允许使用长达31个字符甚至更长的标识符。

标识符是用来为常量、变量、数组、函数及类型等命名的。使用标识符时,一方面要使其尽量有意义,做到见名知义,如PI、name、count、max等,以利于阅读和理解;另一方面要避免在书写时引起混淆,如字母O和数字0、字母I和数字1,字母Z和数字2,减号和