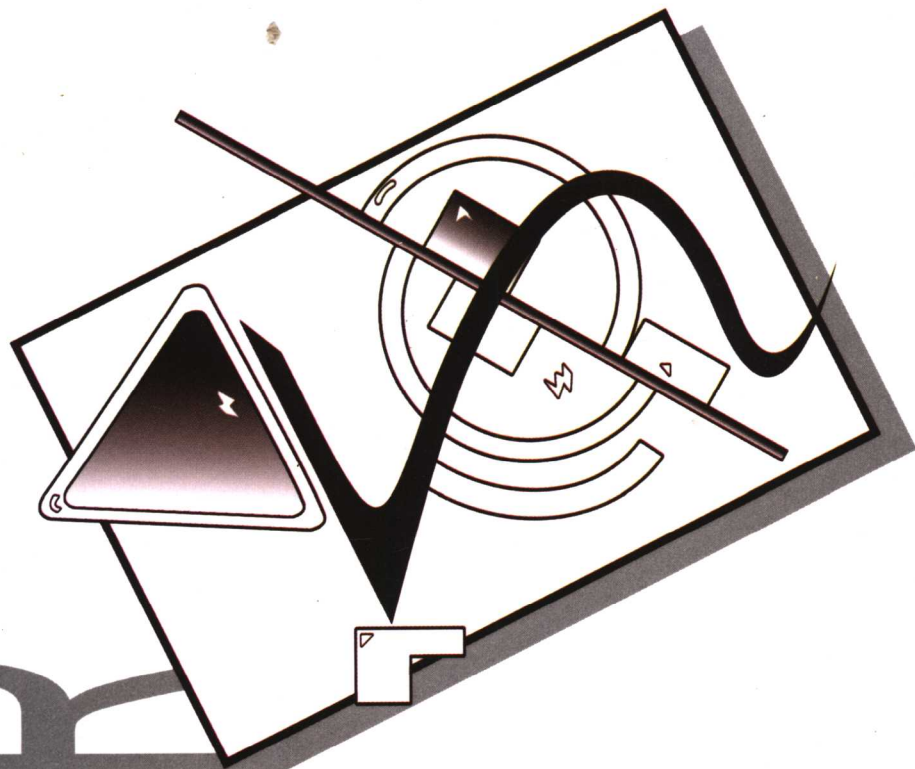


高等职业教育计算机专业推荐教材



C语言 程序设计与实践

高 屹 程自强 编著

 机械工业出版社
CHINA MACHINE PRESS



高等职业教育计算机专业推荐教材

C 语言程序设计与实践

高 屹 程自强 编著



机械工业出版社

C 语言是一种广泛应用的程序设计语言, 它既适合于编写应用软件, 又适合于编写系统软件。本书较全面系统地讲述了 C 语言及其程序设计方法。全书分三部分: 第 1 部分系统地介绍了 C 语言的基本语法和基本程序设计技巧, 共 11 章, 包括 C 语言格式、基本控制结构、函数、数组、指针、结构、文件、编译预处理等内容; 第 2 部分为上机指导部分, 用于对第 1 部分内容学习的巩固与提高; 第 3 部分为习题解答, 供读者参考。

本书从 C 语言程序实例开始, 由浅入深地进行介绍, 通过典型例题对重点部分进行加强, 符合程序设计学习的规律。本书的全部例题都在 Turbo C 2.0 上通过。

本书主要用作高等职业教育教材, 也可作为各类大专院校学生和广大电脑爱好者学习 C 语言的参考书。

图书在版编目 (CIP) 数据

C 语言程序设计与实践 / 高屹, 程自强编著. —北京: 机械工业出版社, 2005.6

(高等职业教育计算机专业推荐教材)

ISBN 7-111-16452-0

I. C... II. ①高...②程... 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 (2005) 第 031554 号

机械工业出版社 (北京市百万庄大街 22 号 邮政编码 100037)

策 划: 胡毓坚

责任编辑: 陈振虹

责任印制: 杨 曦

北京机工印刷厂印刷 · 新华书店北京发行所发行

2005 年 6 月第 1 版 · 第 1 次印刷

787mm×1092mm 1/16 · 16.75 印张 · 413 千字

0001—5000 册

定价: 24.00 元

凡购本图书, 如有缺页、倒页、脱页, 由本社发行部调换

本社购书热线电话 (010) 68326294

封面无防伪标均为盗版

高等职业教育计算机专业推荐教材

编委会成员名单

主 任 王元元

编 委	丁跃潮	黄陈蓉	黄国兴
	李咏梅	逯燕玲	王爱梅
	奚李峰	杨世平	张桂芸

编者的话

根据有关部门对我国信息产业发展的客观需求及劳动力市场现状的调查,在计算机应用和软件专业领域培养技能型紧缺人才,是当务之急。近年来,不仅高等职业技术类院校,而且相当一部分本科类工程技术院校(特别是相当数量高等学校的二级学院、民办院校),都把招收和培养计算机专业技能型紧缺人才列为教育改革的重要举措。为一些院校提供“适时、适度、优选、优质”的计算机专业的高等职业教育系列教材,正是我们组织编写这套“高等职业教育计算机专业推荐教材”(以下简称“推荐教材”)的目标。“推荐教材”由四个模块的30多本教材组成。这些模块是:基础知识模块、程序设计模块、实用技术模块、实践模块。

这套“推荐教材”是“适时”的,因为它努力适应我国信息产业发展和劳动力市场的客观需求,适应计算机行业技术的现状,强调教学内容的先进性和实用性。这套教材十分关注信息技术的最新发展,突出本专业领域的新知识、新技术、新流程和新方法。其中程序设计模块和实用技术模块充分体现了这一特色,所涉及的19本教材既有基础的平台、语言,如《Linux 操作系统》、《C 语言程序设计与实践》,也有最新的《Visual C#.NET 面向对象程序设计教程》、《XML 实用教程》、《JSP 应用教程》等工具,还有十分接近实际工作需要的《Oracle 数据库应用教程》、《计算机网络管理》、《电子商务概论》等实用教材。

这套“推荐教材”是“适度”的,因为它不是简单地摒弃基础理论,而是注意强调理论联系实际,努力做到专业技能型人才能从中学学习到必要和相对系统的基础理论知识,把各种能力的培养和全面素质的提高放在首要的位置。“推荐教材”中基础知识模块的设置,充分体现了这一特色,它囊括了从数学基础、电子基础、硬件技术基础到系统软件基础、应用技术基础、网络技术基础、信息安全基础等10本教材。

这套“推荐教材”是“优选”的,因为充分考虑了现有高中毕业生的认知水平和已有知识,为学生提供适应劳动力市场需要和有职业发展前景的、模块化的教材体系。在学习内容、教学组织等方面留给教师和学生选择和创新的空間,便于教师组织和构建开放式的课程体系,适应学生个性化发展的需要,在灵活的模块化课程结构中自由发展。“推荐教材”的四个模块对重要内容都安排了看似重复的多种教材,供教师和学生去选择。例如,可以在《C 语言程序设计与实践》、《Visual Basic.NET 可视化程序设计教程》、《Visual C#.NET 面向对象程序设计教程》、《Java 程序设计教程》中任意选择一到两门;也可以在《ASP 基础及应用教程》、《JSP 应用教程》中任选一门。

这套“推荐教材”是“优质”的,因为它们作者多数是从事高等职业教育的计算机专业教师,具有长期的计算机实际工作和教育工作经验。这套教材的优质,还体现在它的改革和创新精神上。其中《计算机电路基础》对传统的电路、模拟电路和数字电路课程教材作了重大的改变;《计算机组装与维修教程》则是一门纯实践的课程教材。我们欢迎使用这套教材的师生,指出教材中存在的问题并提出修改意见。

高等职业教育计算机专业推荐教材
编委会

前 言

C 语言目前仍然是世界上最流行的计算机程序设计语言之一，它既适合于编写应用程序，又适合于编写系统程序。

C 语言是在 1972~1973 年间由美国贝尔实验室的 D.M.Ritchie 为描述和实现 UNIX 操作系统而设计实现的。随着 UNIX 系统的巨大成功，C 语言也随之推广开来，此后又几经改进而被广泛移植到各种机器上。1978 年，B.W. Kernighan 和 D.M. Ritchie 合著了一本影响深远的名著《The C Programming Language》，奠定了 C 语言的基础，形成了 K&R 的 C 标准。1983 年，美国国家标准协会(ANSI)根据各种 C 语言版本对 C 语言的发展和扩充，制定了新的标准，称为 ANSI C 标准，1987 年又公布了更新的标准，称为 87ANSI C。之后，国际标准化组织(ISO)又于 1990 年通过了 ISO 的 C 标准，其规定与前者完全相同。本书基本上是按 87ANSI C 标准编写的。

20 世纪 80 年代以来，C 语言已经能够在多种操作系统，如 UNIX、DOS、Windows 等环境下运行，一些软件公司相继推出了 C 语言程序开发系统，如 Microsoft C、Turbo C 等。

20 世纪 90 年代以来，面向对象程序设计方法的发展，使 C++得到广泛应用。C++的基础是 C，C 和 C++交融发展，C++对于“C 的部分”进行了不少扩充和改进。现在，实际上在 C++系统下开发 C 程序比在 C 系统下更方便、更有效。本书希望能为读者打好 C 语言的基础，为以后继续学习 C++做好准备。

C 语言应用如此广泛，是由其自身特点决定的：

(1) C 语言具有高级语言的特点，利用它能编写不依赖于机器硬件设备的应用程序；它又具有低级语言的特点（如位处理、地址操作、寄存器访问等），特别适合编写系统程序，用 C 语言编写的 UNIX 操作系统是 C 语言应用最成功的实例。

(2) C 语言是结构化程序设计语言。C 语言的函数机制，为程序的模块化、结构化提供了强有力的支持。使用 C 语言编程具有易于分工合作、分别编译、便于修改和调试、容易扩充等优点。

(3) C 语言是程序员非常喜欢的语言。利用 C 语言编程，书写简明、紧凑，使用灵活、自由，具有编译效率高、表达和运算能力强的特点，此外 C 开发系统还有丰富的库函数可以利用。

相信读者在学习本书以后，将会对 C 语言的特点有深刻的体会。

但 C 语言也有不足，除了过程式语言共有的一些局限性之外，C 本身还有一些缺陷，例如：对数据类型的约束不严；数组不作越界检查；运算符多，优先级多，又规定了结合性；指针运用复杂等。这些虽然便于专业程序员灵活、方便地编程，但对于初学者来说却增加了困难。

要学习一种程序设计语言，最有效的途径就是用它编写程序，只有通过编程和上机才能更好地理解和掌握教学内容。比如一些语法规则，它们对编程很重要，但仅靠课堂讲授既枯燥又记不住；但是如果通过几次编程和上机练习，就会自然、熟练地掌握了。

学会上机调试程序也是程序设计教学的重要内容和基本要求，可以说不会调试程序就不

可能实际地进行编程工作。因此，我们不仅要学会编程的理论和方法，还应当学会通过上机调试发现程序中的错误和排除这些错误，使之能正确地运行。

为了使读者能够全面、系统地掌握 C 语言的编程技巧，特别是为了加强对实践环节的指导，我们将本书分成三部分：

第 1 部分是 C 语言程序设计，系统地对 C 语言的基本语法和基本技巧加以介绍，是本书的主体部分。

第 2 部分是上机实验。每次实验内容的安排，基本与第 1 部分的章相对应，一般每次实验安排 3 个左右的题目，读者可以根据上机时间安排，选择其中部分题目上机实践。

第 3 部分是题解。应该说明的是，由于程序设计的特点，我们给出的答案并非是惟一的，也不一定是最佳的，仅供读者参考。

本书中的全部例题均在 Turbo C 2.0 上调试通过。

编 者

目 录

编者的话
前言

第 1 部分 C 语言程序设计

第 1 章 C 程序实例	1
1.1 几个例子	1
1.2 C 的基本字符与词汇	3
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
1.4 习题	6
第 2 章 数据类型、运算符及表达式	8
2.1 数据和数据类型	8
2.2 常量	8
2.2.1 整型常量	9
2.2.2 实型常量	9
2.2.3 字符常量	9
2.2.4 字符串常量	10
2.2.5 符号常量	10
2.3 变量	11
2.3.1 整型变量	11
2.3.2 实型变量	12
2.3.3 字符型变量	12
2.4 赋值与算术运算	13
2.4.1 赋值运算	13
2.4.2 二元算术运算	13
2.4.3 自加、自减运算	14
2.4.4 复合赋值运算	15
2.5 变量赋初值	15
2.6 类型转换	16
2.6.1 自动类型转换	16
2.6.2 强制类型转换	16
2.7 习题	17
第 3 章 数据的输入输出	19

3.1	字符输入输出函数	19
3.1.1	字符输入函数 <code>getchar</code>	19
3.1.2	字符输出函数 <code>putchar</code>	20
3.2	格式输入输出函数	21
3.2.1	格式输出函数 <code>printf</code>	21
3.2.2	格式输入函数 <code>scanf</code>	25
3.3	应用举例	26
3.4	习题	28
第 4 章	C 语言程序的基本控制结构	30
4.1	程序的三种基本控制结构	30
4.2	顺序结构	31
4.2.1	空语句和表达式语句	31
4.2.2	复合语句	31
4.2.3	顺序结构程序设计	31
4.3	选择结构	33
4.3.1	关系运算和逻辑运算	33
4.3.2	<code>if</code> 语句	35
4.3.3	<code>switch</code> 语句	39
4.4	循环结构	40
4.4.1	<code>while</code> 循环语句	40
4.4.2	<code>do-while</code> 循环语句	42
4.4.3	<code>for</code> 循环语句	43
4.4.4	循环的退出	46
4.5	应用举例	48
4.6	习题	51
第 5 章	数组	54
5.1	一维数组	54
5.1.1	一维数组的定义	54
5.1.2	一维数组的使用	55
5.1.3	一维数组的初始化	56
5.2	二维数组	57
5.2.1	二维数组的定义	57
5.2.2	二维数组的初始化	58
5.2.3	二维数组的引用	59
5.3	字符数组和字符串	60
5.3.1	字符数组的定义	60
5.3.2	字符串	61
5.4	应用举例	64
5.5	习题	67

第 6 章 函数	69
6.1 C 程序与函数	69
6.2 函数的定义和说明	70
6.2.1 函数的定义	70
6.2.2 函数的说明	73
6.3 函数的调用	74
6.3.1 函数调用的格式	74
6.3.2 参数的传递	76
6.4 应用举例 1	78
6.5 函数的嵌套调用与递归调用	80
6.5.1 函数的嵌套调用	80
6.5.2 函数的递归调用	82
6.6 作用域与存储类	84
6.6.1 作用域	84
6.6.2 存储类	86
6.6.3 内部函数与外部函数	89
6.7 应用举例 2	90
6.8 习题	93
第 7 章 指针	96
7.1 指针的概念	96
7.1.1 什么是指针	96
7.1.2 指针变量的定义	97
7.1.3 指针的运算	97
7.2 指针与函数参数	99
7.3 指针与数组	101
7.3.1 一维数组的指针	101
7.3.2 多维数组的指针	106
7.3.3 字符指针与字符串	107
7.4 指针数组	111
7.4.1 指针数组的定义	111
7.4.2 用指针数组处理多维数组	112
7.4.3 用字符指针数组处理字符串组	113
7.4.4 用作 main 函数的形参	115
7.5 指针与函数	116
7.5.1 返回指针值的函数	116
7.5.2 指向函数的指针	118
7.6 指向指针的指针	119
7.7 应用举例	120
7.8 习题	122

第 8 章 结构、联合和枚举	125
8.1 结构的概念	125
8.1.1 结构类型的定义	125
8.1.2 结构变量的说明	125
8.1.3 结构变量的引用	127
8.1.4 结构变量的初始化	128
8.2 结构数组	129
8.3 指向结构的指针	131
8.3.1 指向结构的指针	131
8.3.2 结构指针作函数参数	133
8.4 动态数据结构	134
8.4.1 内存的动态分配和释放函数	135
8.4.2 链表	136
8.5 联合	143
8.5.1 联合的定义	143
8.5.2 联合变量的说明与引用	144
8.6 枚举	146
8.7 类型定义	148
8.8 应用举例	150
8.9 习题	154
第 9 章 位处理	156
9.1 位运算	156
9.2 位段	159
9.3 应用举例	161
9.4 习题	163
第 10 章 编译预处理	165
10.1 宏定义	165
10.1.1 简单的宏定义	165
10.1.2 带参数的宏定义	167
10.2 文件包含	168
10.3 条件编译	170
10.4 应用举例	172
10.5 习题	173
第 11 章 文件	175
11.1 C 文件的概念	175
11.2 文件类型指针	176
11.3 文件的打开与关闭	177
11.3.1 文件的打开——fopen 函数	177
11.3.2 文件的关闭——fclose 函数	179

11.4 文件的读写	179
11.4.1 字符读写函数——fputc 和 fgetc	179
11.4.2 字符串读写函数——fgets 和 fputs	181
11.4.3 数据块读写函数——fread 和 fwrite	183
11.4.4 格式化读写函数——fscanf 和 fprintf	184
11.5 文件的定位	186
11.5.1 rewind 函数	186
11.5.2 ftell 函数	187
11.5.3 fseek 函数	187
11.6 输入输出重定向	188
11.7 应用举例	189
11.8 习题	191

第2部分 上机实验

实验1 熟悉 Turbo C 2.0 系统与顺序结构程序设计	193
实验2 循环结构程序设计	196
实验3 程序调试	198
实验4 利用数组的编程	203
实验5 函数	204
实验6 指针	207
实验7 结构和枚举	208
实验8 链表应用	210
实验9 位处理与编译预处理	211
实验10 文件操作	212

第3部分 习题参考答案

第1章习题参考答案	213
第2章习题参考答案	214
第3章习题参考答案	214
第4章习题参考答案	216
第5章习题参考答案	219
第6章习题参考答案	224
第7章习题参考答案	229
第8章习题参考答案	232
第9章习题参考答案	236
第10章习题参考答案	239
第11章习题参考答案	241
附录	247
附录 A Turbo C 2.0 编译系统使用方法简介	247

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	252
附录 D 常用的 C 库函数	253
参考文献	256

第 1 部分 C 语言程序设计

第 1 章 C 程序实例

学习一种程序设计语言，最有效的途径就是用它来编写程序。下面几个简单的程序，将使读者对 C 语言有一个初步的了解。

1.1 几个例子

【例 1-1】 首先给出一个最简单的 C 语言程序。

```
/* example 1-1 */
#include <stdio.h>
main()
{
    printf("Hello,C!\n");
}
```

运行结果：

Hello,C!

程序中 `main()` 表示主函数，是 C 语言程序必须有的一个函数。由花括号 `{ }` 括起来的部分是函数体，函数体用来规定该函数所要完成的工作，它由一些语句组成，每个语句都以分号 `“;”` 结束。本例中函数体内只有一个输出语句，`printf()` 是输出函数调用，它调用系统标准输入输出函数库中的 `printf` 函数。双引号内的字符串照原样输出，`\n` 是换行符，即在输出字符串后回车换行。语句最后有一个分号 `“;”`。

程序开头出现的 `“#include <stdio.h>”` 是一条编译预处理命令，它的作用是编译之前将尖括号 `<>` 括起来的文件内容放到该命令的位置，有关这方面的知识后面章节要详细介绍。

【例 1-2】 用 C 语言编写程序，要求输入圆半径值，计算圆面积并输出。

```
/* example 1-2 */
#include <stdio.h>
#define PI 3.14159
main() /* 计算圆的面积 */
{
    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 */
}

```

这个程序比上例稍长一些，但仍是一个很简单的程序。

程序第 4、9、11 行/*和*/中间的内容是注释，简要说明了该程序或语句在做什么。为便于理解，我们用汉字表示注释，当然也可用英语或拼音作注释。在/*和*/之间的字符，对编译和运行不起作用，只是为了使程序更易理解。第 2、3 行都是编译预处理命令，其中第一条使用同上例，后一条的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，都是编译预处理命令。

该程序的函数体由若干语句组成，每个语句都以“;”结束。函数体中第 1、2 句为变量说明，分别说明 r 为整形变量，s 为实型变量。第 3 条语句为 printf 函数调用，在屏幕上显示“请输入圆的半径值:”，用来提示用户准备输入数据。第 4 条语句为 scanf 函数调用，它的作用是等待用户从键盘上输入数据，&r 中的“&”含义为“取地址”，即将数值输入到变量 r 的地址所标识的内存单元中，也就是输入给 r。&r 前面的“%d”是输入格式符，用来说明输入时数据的类型，“%d”表示输入十进制整数。第 5 条是赋值语句，用来计算赋值号“=”右边表达式的值，即圆面积的值，并赋给变量 s。最后一句为 printf 函数调用，双引号内的字符串“该圆面积=%f\n”，在输出时其中的“%f”表示将变量 s 按规定的实型数据格式输出，“该圆面积=”照原样输出。程序的运行结果如下：

```

请输入圆的半径值:1<CR>
该圆面积=3.141590

```

【例 1-3】 编写一个程序，输入两个数，求其中较大的数。

```

/* example 1-3 */
#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 */
}

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 main()和被调用函数 max()。max 函数的作用是将 x、y 中

较大的数和值赋给 `z`，并通过 `return` 语句将 `z` 的值返回到主调函数 `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` 函数体中第三行为调用 `max` 函数，调用时将实在参数 `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，把这个值赋给变量 `c`，然后输出 `c` 的值。程序运行结果如下：

```
4,6↵      (从键盘上输入 4、6 给 a、b)
max=6      (输出 c 的值)
```

本例中用到了函数定义、函数调用、形参和实参等概念，在后面有关章节中要作详细的讨论。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

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

定义了函数的名字为 `max`，函数的类型为 `int`，函数的形式参数为 `x` 和 `y`，它们的类型为 `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;
```

参数在圆括号内命名，参数类型在函数体左花括号前说明。

函数体，即函数头下面花括号 `{}` 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：

- 变量说明。如例 1-3 中 `main` 函数中的 “`int a,b,c`”。
- 执行部分。由若干语句组成，每个语句都以 “`;`” 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，往往以缩进写法来体现语句的层次。

1.2 C 的基本字符与词汇

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集，以便处理系统能正确识别它们。C 语言中使用的合法字符有：

(1) 字母和数字

小写字母：a b c...z

大写字母：A B C...Z

数字：0 1 2...9

(2) 特殊字符

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

(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 语言的合法字符集。C 语言程序所用的全部字符都在这个字符集中。

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

a, x1, _fout, b4c, High, _x_y

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，如 ABC、Abc 和 abc 分别表示了三个不同的标识符。这一点与其他高级语言有区别，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。但也有的编译系统允许使用长达 31 个字符的标识符。

标识符是用来为常量、变量、数组、函数及类型等取名的。使用标识符时，一方面要取得尽量有意义，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“-”和下划线“_”等都易混淆，使用时要小心。再有，不能把保留字用作一般的标识符，因为保留字在 C 语言中有专门的含义。

(2) 保留字

保留字又称关键字，是语言中具有特定含义的一些单词，对保留字不能重新定义，也不能用作一般的标识符。C 语言有如下保留字：

auto

break

case

char

const