

C 语言程序设计

C YUYAN CHENGXU SHEJI

主 编 赵 颖

副主编 金 丹 刘 靖

哈尔滨地图出版社



C 语言程序设计

C YUYAN CHENGXU SHEJI

主 编 赵 颖

副主编 金 丹 刘 靖

哈尔滨地图出版社

• 哈尔滨 •

图书在版编目 (CIP) 数据

C 语言程序设计 / 赵颖主编. — 哈尔滨: 哈尔滨地图出版社, 2008. 12

ISBN 978-7-5465-0010-2

I. C… II. 赵… III. C 语言—程序设计 IV. TP312

中国版本图书馆 CIP 数据核字 (2009) 第 000511 号

内 容 提 要

本书系统地介绍了 C 语言的基本语法知识和程序的基本结构, 包括 C 语言中的数据类型、运算符、表达式、基本语句、数组、函数、指针、文件、结构等内容。全书内容安排循序渐进, 叙述条理清楚, 程序分析透彻, 对建立正确的 C 语言概念, 学会基本的编程方法有指导作用, 适合初学者阅读和自学。

本书可作为高职高专院校 C 语言程序设计课程的教材, 也可作为各类高校计算机专业或非计算机专业的教材或参考书, 亦适合初学者自学使用。

哈尔滨地图出版社出版发行

(地址: 哈尔滨市南岗区测绘路 2 号 邮政编码: 150086)

哈尔滨翰翔印务有限公司印刷

开本: 787 mm×1 092 mm 1/16 印张: 10.125 字数: 250 千字

ISBN 978-7-5465-0010-2

2008 年 12 月第 1 版 2009 年 1 月第 1 次印刷

印数: 1~100 册 定价: 20.00 元

前 言

C 语言是一种应用得非常广泛的编程语言,是最早出现和流行的计算机高级程序设计语言,它既具有功能丰富、语句简洁、实用方便、语法灵活、数据结构多样、能对硬件进行操作、高移植性和通用性等诸多高级语言的优点,又带有庞大的函数库,支持过程化和结构化的程序设计,是系统软件和应用软件开发人员的主要开发工具之一。C 语言的独特魅力使它成为计算机及相关专业人员学习计算机程序设计的首选语言。

现在,C 语言不仅为计算机专业工作者所使用,而且为广大计算机应用人员所喜爱和使用。在我国各高校的计算机和非计算机专业都开设了 C 语言课程,在高职院校的计算机课程设置中,C 语言也是程序设计语言的必修课程。在编写本书的过程中,作者结合自己从事 C 语言教学的实际经验,理论联系实际,力求通俗易懂。本书在体系结构安排上尽可能将概念、知识点与例题结合起来,并且所选题目典型、针对性强,通过一些典型程序将前后知识点联系起来,使读者在对比中理解各种实现方式的特点和异同,能够融会贯通、举一反三。

本书是根据教育部最新高职高专教育教学大纲要求编写的,因此本书可以作为高职高专院校计算机专业和非计算机专业的 C 语言程序设计教学教材,同时也适合 C 语言程序设计初学者。

本书共有 11 章:第 1 章介绍 C 语言概论;第 2 章介绍基本数据类型;第 3 章介绍运算符与表达式;第 4 章介绍 C 语言程序设计;第 5 章介绍数组;第 6 章介绍函数;第 7 章介绍指针;第 8 章介绍文件;第 9 章介绍结构体与共用体;第 10 章介绍位运算;第 11 章介绍预处理。本书从 C 语言的基本数据元素、基本语句和结构控制语句、构造数据类型的定义和使用、函数的定义和调用,到指针的灵活运用等方面都进行了由浅入深的讲解。该书的特点是结构合理、层次分明、例题丰富、通俗易懂、实用性强。

本书由赵颖担任主编,金丹、刘靖担任副主编。赵颖编写了其中的第 1, 4, 5, 9 章;金丹编写了其中的第 2, 3, 6, 10 章;刘靖编写了其中的第 7, 8, 11 章。

尽管本书三易其稿,但由于作者水平有限,加上时间仓促,书中缺点和错误在所难免,恳请读者批评指正。

作 者
2008 年 12 月

目 录

第 1 章 C 语言概论	1
1.1 C 语言概述.....	1
1.1.1 C 语言的发展过程.....	1
1.1.2 C 语言的特点.....	1
1.2 C 语言程序的结构.....	2
第 2 章 基本数据类型	7
2.1 C 语言的数据类型.....	7
2.2 整数型数据.....	7
2.2.1 整型常量.....	8
2.2.2 整型变量.....	9
2.3 实型数据.....	10
2.3.1 实型常量.....	10
2.3.2 实型变量.....	10
2.4 字符型数据.....	11
2.4.1 字符型常量.....	11
2.4.2 字符型变量.....	12
2.4.3 字符串常量.....	13
2.4.4 符号常量.....	13
2.5 变量的初值和类型转换.....	14
2.5.1 变量赋初值.....	14
2.5.2 变量类型的转换.....	14
第 3 章 运算符与表达式	16
3.1 运算符的种类.....	16
3.1.1 C 语言的运算符种类.....	16
3.1.2 优先级和结合性.....	17
3.2 算术运算符与算术表达式.....	17
3.2.1 基本的算术运算符.....	17
3.2.2 算术表达式.....	18
3.3 关系运算符与关系表达式.....	18
3.3.1 关系运算符.....	18
3.3.2 关系表达式.....	19
3.4 逻辑运算符与逻辑表达式.....	19
3.4.1 逻辑运算符.....	19
3.4.2 逻辑表达式.....	20
3.5 其他运算符与表达式.....	20
3.5.1 简单赋值运算符.....	20
3.5.2 复合赋值符及表达式.....	21

3.5.3	逗号运算符和逗号表达式	21
第4章	C语言程序设计	23
4.1	C语言程序的语句	23
4.2	数据输入输出语句	24
4.2.1	输出语句	24
4.2.2	输入语句	26
4.3	分支结构程序	30
4.3.1	if语句	30
4.3.2	if语句的嵌套	32
4.3.3	switch语句	34
4.4	循环结构程序	36
4.4.1	while语句	36
4.4.2	do-while语句	37
4.4.3	for语句	38
4.4.4	转移语句	41
4.4.5	例题	42
第5章	数组	44
5.1	一维数组	44
5.1.1	数组类型的说明	44
5.1.2	数组元素的表示方法	45
5.1.3	数组的赋值	46
5.2	二维数组	47
5.2.1	二维数组类型说明	47
5.2.2	二维数组元素的表示方法	48
5.2.3	二维数组的初始化	49
5.3	字符数组	50
5.3.1	字符数组类型说明	50
5.3.2	字符数组的初始化	50
5.3.3	字符串常用函数	51
5.4	程序举例	54
第6章	函数	57
6.1	函数的分类和定义	57
6.1.1	函数的分类	57
6.1.2	函数定义的一般形式	59
6.2	函数调用	61
6.2.1	函数调用的一般形式	61
6.2.2	函数说明	62
6.3	函数的参数和函数的值	63
6.3.1	函数的参数	63
6.3.2	函数的值	64
6.3.3	数组作为函数参数	64

6.4	函数的嵌套调用和递归调用	69
6.4.1	函数的嵌套调用	69
6.4.2	函数的递归调用	71
6.5	变量的作用域	74
6.5.1	局部变量	74
6.5.2	全局变量	76
6.6	变量的存储类型	78
6.6.1	动态存储方式与静态存储方式	78
6.6.2	存储类型说明	79
6.7	内部函数和外部函数	84
6.7.1	内部函数	84
6.7.2	外部函数	84
第7章	指针	85
7.1	指针的基本概念	85
7.1.1	指针	85
7.1.2	指针变量	85
7.2	指针变量的使用	86
7.2.1	指针变量的类型说明	86
7.2.2	指针变量的赋值	86
7.2.3	指针变量的运算	87
7.3	数组指针变量	89
7.3.1	指向一维数组的指针	89
7.3.2	指向多维数组的指针	91
7.3.3	使用字符串指针变量与字符数组的区别	94
7.4	函数指针变量	95
7.4.1	函数指针变量的调用	95
7.4.2	指针型函数	96
7.5	指针数组及带参 main 函数	97
7.5.1	指针数组的定义和使用	97
7.5.2	main 函数的参数	100
7.5.3	指向指针的指针变量	102
第8章	文件	103
8.1	文件概述	103
8.1.1	文件的分类	103
8.1.2	文件指针	104
8.2	文件的打开与关闭	104
8.2.1	文件打开函数 fopen	104
8.2.2	文件关闭函数 fclose	106
8.3	文件的读写	106
8.3.1	字符读写函数 fgetc 和 fputc	106
8.3.2	字符串读写函数 fgets 和 fputs	109

8.3.3	数据块读写函数 fread 和 fwrite	111
8.3.4	格式化读写函数 fscanf 和 fprintf	112
8.4	文件的定位	114
8.5	文件检测函数	115
8.5.1	文件检测函数类型	115
8.5.2	C 语言库文件	116
第 9 章	结构体与共用体	117
9.1	结构体类型定义和结构变量说明	117
9.1.1	结构体的定义	117
9.1.2	结构类型变量的说明	118
9.1.3	结构变量的赋值	119
9.1.4	结构变量的初始化	120
9.2	结构体数组	121
9.3	结构体指针变量	123
9.3.1	结构体指针变量的说明和使用	123
9.3.2	结构数组指针变量	124
9.3.3	结构指针变量作函数参数	125
9.4	链表	126
9.4.1	存储空间的动态分配	126
9.4.2	链表的概念	128
9.4.3	链表的基本操作	128
9.5	共用体	135
9.5.1	共用体的概念	135
9.5.2	共用体类型的定义和共用体变量的说明	135
9.5.3	共用体变量的赋值和使用	136
9.6	枚举类型	138
9.6.1	枚举的定义	138
9.6.2	枚举变量的说明	138
9.6.3	枚举类型变量的赋值和使用	139
9.7	自定义类型	140
第 10 章	位运算	141
10.1	位运算符和位运算	141
10.2	位域	143
10.2.1	位域的定义和位域变量的说明	143
10.2.2	位域的使用	144
第 11 章	预处理	146
11.1	宏定义	146
11.1.1	无参宏定义	146
11.1.2	带参宏定义	148
11.2	文件包含	151
11.3	条件编译	152

第 1 章 C 语言概论

C 语言自出现到现在已经经历了三十多年的时间, 是使用最广泛的程序设计语言之一, 本章主要介绍了 C 语言的发展过程及主要特点、C 语言程序的编制环境和步骤, 使大家在简单程序的编制过程中初步掌握 C 语言程序的构成、特点和编写规范。

1.1 C 语言概述

1.1.1 C 语言的发展过程

C 语言是在 20 世纪 70 年代初问世的。1978 年由美国电话电报公司(AT&T) 贝尔实验室正式发表了 C 语言。同时由 B.W.Kernighan 和 D.M.Ritchie 合著了著名的《THE C PROGRAMMING LANGUAGE》一书。通常简称为《K&R》, 也有人称之为《K&R》标准。但是, 在《K&R》中并没有定义一个完整的标准 C 语言, 后来由美国国家标准学会在此基础上制定了一个 C 语言标准, 于 1983 年发表。通常称之为 ANSI C。

早期的 C 语言主要是用于 UNIX 系统。由于 C 语言的强大功能和各方面的优点逐渐为人们所认识, 到了 20 世纪 80 年代, C 语言开始进入其他操作系统, 并很快在各类大、中、小和微型计算机上得到了广泛的使用, 成为当代最优秀的程序设计语言之一。

目前比较流行的 C 语言有 Turbo C, Microsoft C, Borland C 等, 这些 C 语言版本不仅实现了 ANSI C 标准, 而且在此基础上各自作了一些扩充, 使之更加方便、完美。

1.1.2 C 语言的特点

C 语言发展如此迅速, 并且成为最受欢迎的语言之一, 主要是因为它具有强大的功能。C 语言的特点:

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

① C 语言中的 32 个关键字, 如表 1-1 所示。

注意: 在 C 语言中, 关键字都是小写的。

② C 语言中的 9 种控制语句如下:

if(...)...else...	switch(...)...break	for(...)...	while(...)...
do ... while(...)	continue	break	goto return

表 1-1 C 语言的关键字

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

(2) 运算符丰富, 共有 34 种。C 语言把括号、赋值、逗号等都作为运算符处理, 从而使 C 语言的运算类型极为丰富, 可以实现其他高级语言难以实现的运算。

(3) 数据结构类型丰富, 有整型、实型、字符型、枚举类型等基本类型, 有数值、结构体、共用体等构造类型以及指针类型, 还为用户提供了自定义数据类型, 特别是引入了指针概念, 能够实现复杂的数据结构。

(4) C 语言的控制语句形式多样、使用方便, 具有结构化的控制语句, 便于结构化模块的实现和控制, 便于程序的编制和维护。

(5) 语法限制不太严格, 程序设计自由度大。

(6) C 语言允许直接访问物理地址, 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, 有人把它称为中级语言。

(7) 生成目标代码质量高, 程序执行效率高。

(8) 与汇编语言相比, 用 C 语言写的程序可移植性好。

但是, C 语言对程序员要求也高, 程序员用 C 语言写程序会感到限制少、灵活性大、功能强, 但较其他高级语言在学习上要困难一些。

1.2 C 语言程序的结构

为了说明 C 语言源程序结构的特点, 先看以下几个程序。这几个程序由简到难, 表现了 C 语言源程序在组成结构上的特点。虽然有关内容还未介绍, 但可从这些例子中了解到组成一个 C 源程序的基本部分和书写格式。

【例 1.1】第一个 C 程序。

```
main()
{printf("欢迎进入 C 语言世界!\n");
}
```

main 是主函数的函数名, 表示这是一个主函数。每一个 C 源程序都必须有, 且只能有一个主函数 (main 函数)。函数调用语句 printf 函数的功能是把要输出的内容送到显示器去显示。printf 函数是一个由系统定义的标准函数, 可在程序中直接调用。

【例 1.2】求输入任意数的正弦值。

```
#include <math.h>
#include <stdio.h>
main()
{
double x, s; /*定义两个实数变量, 以被后面程序使用*/
printf("input number:\n"); /*显示提示信息*/
```

```
scanf("%lf",&x);          /*从键盘获得一个实数 x*/
s=sin(x);                /* 求 x 的正弦, 并把它赋给变量 s*/
printf("sine of %lf is %lf\n",x,s); /*显示程序运算结果*/
}
```

程序的功能是从键盘输入一个数 x , 求 x 的正弦值, 然后输出结果。在 `main()` 之前的两行称为预处理命令(详见后面)。预处理命令还有其他几种, 这里的 `include` 称为文件包含命令, 其意义是把引号“”或尖括号“<”内指定的文件包含到本程序中来, 成为本程序的一部分。被包含的文件通常是由系统提供的, 其扩展名为 `.h`。因此也称为头文件或首部文件。C 语言的头文件中包括了各个标准库函数的函数原型。因此, 凡是在程序中调用一个库函数时, 都必须包含该函数原型所在的头文件。在本例中, 使用了三个库函数: 输入函数 `scanf`, 正弦函数 `sin`, 输出函数 `printf`。`sin` 函数是数学函数, 其头文件为 `math.h` 文件, 因此在程序的主函数前用 `include` 命令包含了 `math.h`。`scanf` 和 `printf` 是标准输入输出函数, 其头文件为 `stdio.h`, 在主函数前也用 `include` 命令包含了 `stdio.h` 文件。

需要说明的是, C 语言规定对 `scanf` 和 `printf` 这两个函数可以省去对其头文件的包含命令。所以在本例中也可以删去第二行的包含命令 `#include`。同样, 在例 1.1 中使用了 `printf` 函数, 也省略了包含命令。

在例题中的主函数体中又分为两部分: 一部分为说明部分, 另一部分为执行部分。说明是指变量的类型说明。例题中未使用任何变量, 因此无说明部分。C 语言规定, 源程序中所有用到的变量都必须先说明后使用, 否则将会出错。这一点是编译型高级程序设计语言的一个特点, 与解释型的 BASIC 语言是不同的。说明部分是 C 源程序结构中很重要的组成部分。本例中使用了两个变量 x , s , 用来表示输入的自变量和 `sin` 函数值。由于 `sin` 函数要求这两个量必须是双精度浮点型, 故用类型说明符 `double` 来说明这两个变量。说明部分后的四行为执行部分或称为执行语句部分, 用以完成程序的功能。执行部分的第一行是输出语句, 调用 `printf` 函数在显示器上输出提示字符串, 请操作人员输入自变量 x 的值。第二行为输入语句, 调用 `scanf` 函数, 接受键盘上输入的数并存入变量 x 中。第三行是调用 `sin` 函数并把函数值送到变量 s 中。第四行是用 `printf` 函数输出变量 s 的值, 即 x 的正弦值。程序结束。

在前面两个例子中用到了输入和输出函数 `scanf` 和 `printf`, 在第四章中我们要详细介绍, 这里我们先简单介绍一下它们的格式以便下面使用。`scanf` 和 `printf` 这两个函数分别称为格式输入函数和格式输出函数。其意义是按指定的格式输入输出值。因此, 这两个函数在括号中的参数表都由以下两部分组成:

“格式控制串”, 参数表。

格式控制串是一个字符串, 必须用双引号括起来, 它表示了输入输出量的数据类型。各种类型的格式表示法可参阅第二章。在 `printf` 函数中还可以在格式控制串内出现非格式控制字符, 这时在显示屏幕上将原文照应。参数表中给出了输入或输出的量。当有多个量时, 用逗号间隔。例如:

```
printf("sine of %lf is %lf\n",x,s);
```

其中 `%lf` 为格式字符, 表示按双精度浮点数处理。它在格式串中两次出现, 对应了 x 和 s 两个变量。其余字符为非格式字符, 则照原样输出在屏幕上。

```

【例 1.3】 main() {
int x, y, z;
printf("input two numbers:\n"); scanf("%d%d", &x, &y);
z=max(x, y);
printf("maxmum=%d", z);
}
int max(int a, int b) {
if(a>b) return a; else return b;
}

```

上例程序的功能是由用户输入两个整数，程序执行后输出其中较大的数。本程序由两个函数组成：主函数和 max 函数。函数之间是并列关系。可从主函数中调用其他函数。max 函数的功能是比较两个数，然后把较大的数返回给主函数。max 函数是一个用户自定义函数，因此，在主函数中要给出说明（程序第三行）。可见，在程序的说明部分中，不仅可以有变量说明，还可以有函数说明。在程序的每行后用“/*”和“*/”括起来的内容为注释部分，程序不执行注释部分。

上例中程序的执行过程是，首先在屏幕上显示提示串，请用户输入两个数，回车后由 scanf 函数语句接收这两个数并送入变量 x, y 中，然后调用 max 函数，并把 x, y 的值传送给 max 函数的参数 a, b。在 max 函数中比较 a, b 的大小，把大者返回给主函数的变量 z，最后在屏幕上输出 z 的值。

1. C 源程序的结构特点：

- ①一个 C 语言源程序可以由一个或多个源文件组成。
- ②每个源文件可由一个或多个函数组成。
- ③一个源程序不论由多少个文件组成，都有一个且只能有一个 main 函数，即主函数。
- ④源程序中可以有预处理命令(include 命令仅为其中的一种)，预处理命令通常应放在源文件或源程序的最前面。
- ⑤每一个说明、每一个语句都必须以分号结尾。但预处理命令、函数头和花括号“}”之后不能加分号。

⑥标志符、关键字之间必须至少加一个空格以示间隔。若已有明显的间隔符，也可不再加空格来间隔。

2. 书写程序时应遵循的规则

从书写清晰，便于阅读、理解、维护的角度出发，在书写程序时，应遵循以下规则：

- ①一个说明或一个语句占一行。
- ②用“{}”括起来的部分，通常表示了程序的某一层结构。“{}”一般与该结构语句的第一个字母对齐，并单独占一行。

③低一层次的语句或说明可比高一层次的语句或说明缩进若干格后书写。以便看起来更加清晰，增加程序的可读性。在编程时应力求遵循这些规则，以养成良好的编程风格。

3. C 语言的字符集

字符是组成语言的最基本的元素。C 语言字符集由字母、数字、空格、标点和特殊字符组成。在字符常量、字符串常量和注释中还可以使用汉字或其他可表示的图形符号。

- ①字母。小写字母 a~z 共 26 个，大写字母 A~Z 共 26 个。

②数字。0~9 共 10 个。

③空白符。空格符、制表符、换行符等统称为空白符。空白符只在字符常量和字符串常量中起作用。在其他地方出现时，只起间隔作用，编译程序对它们忽略。因此，在程序中使用空白符与否，对程序的编译不发生影响，但在程序中适当的地方使用空白符将增加程序的清晰性和可读性。

④标点和特殊字符。

4. C 语言词汇

在 C 语言中使用的词汇分为六类：标志符，关键字，运算符，分隔符，常量，注释符等。

(1) 标志符

在程序中使用的变量名、函数名、标号等统称为标志符。除库函数的函数名由系统定义外，其余都由用户自定义。C 语言规定，标志符只能是字母(A~Z, a~z)、数字(0~9)、下划线组成的字符串，并且其第一个字符必须是字母或下划线。

以下标志符是合法的：

a, x, 3x, BOOK 1, sum5。

以下标志符是非法的：

3s, 以数字开头；

s*T, 出现非法字符*；

-3x, 以减号开头；

bowy-1, 出现非法字符-(减号)。

在使用标志符时还必须注意以下几点：

①标准 C 语言不限制标志符的长度，但它受各种版本的 C 语言编译系统限制，同时也受到具体机器的限制。例如，在某版本 C 语言中规定标志符前八位有效，当两个标志符前八位相同时，则被认为是同一个标志符。

②在标志符中，大小写是有区别的。例如 BOOK 和 book 是两个不同的标志符。

③标志符虽然可由程序员随意定义，但标志符是用于标志某个量的符号。因此，命名应尽量有相应的意义，以便阅读理解，做到“顾名思义”。

(2) 关键字

关键字是由 C 语言规定的具有特定意义的字符串，通常也称为保留字。用户定义的标志符不应与关键字相同。C 语言的关键字分为以下几类：

①类型说明符

用于定义、说明变量、函数或其他数据结构的类型。如前面例题中用到的 int, double 等。

②语句定义符

用于表示一个语句的功能。如例 1.3 中用到的 if else 就是条件语句的语句定义符。

③预处理命令字

用于表示一个预处理命令。如前面各例中用到的 include。

(3) 运算符

C 语言中含有相当丰富的运算符。运算符与变量、函数一起组成表达式，表示各种运算功能。运算符由一个或多个字符组成。

(4) 分隔符

在C语言中采用的分隔符有逗号和空格两种。逗号主要用在类型说明和函数参数表中，分隔各个变量。空格多用于语句各单词之间，作间隔符。在关键字、标志符之间必须要有一个以上的空格符作间隔，否则将会出现语法错误，例如把 `int a;` 写成 `inta;` C语言编译器会把 `inta` 当成一个标志符处理，其结果必然出错。

(5) 常量

C语言中使用的常量可分为数字常量、字符常量、字符串常量、符号常量、转义字符等多种。在第二章中将专门给予介绍。

(6) 注释符

C语言的注释符是以“`/*`”开头并以“`*/`”结尾的串。在“`/*`”和“`*/`”之间的即为注释。程序编译时，不对注释作任何处理。注释可出现在程序中的任何位置。注释用来向用户提示或解释程序的意义。在调试程序中对暂不使用的语句也可用注释符括起来，使翻译跳过不作处理，待调试结束后再去掉注释符。

第 2 章 基本数据类型

信息要以数据的形式进行表示,因此数据是组成程序的重要组成部分。如何表示数据,如何将不同特点的数据分门别类地进行表示和处理,这是编制程序的重要基础。本章将学习数据的基本表示方式,主要包括常量和变量的特点、种类和定义形式,以及 C 语言各种数据类型的基本特点及其定义方法等。

2.1 C 语言的数据类型

在第一章中,我们已经看到程序中使用的各种变量都应预先加以说明,即先说明,后使用。所谓数据类型,是按被说明量的性质、表示形式、占据存储空间的大小、构造特点来划分的。在 C 语言中,数据类型可分为基本数据类型、构造数据类型、指针类型、空类型四大类。

1. 基本数据类型

基本数据类型最主要的特点是,其值不可以再分解为其他类型。也就是说,基本数据类型是自我说明的。

2. 构造数据类型

是根据已定义的一个或多个数据类型用构造的方法来定义的。也就是说,一个构造类型的值可以分解成若干个“成员”或“元素”。每个“成员”都是一个基本数据类型或又是一个构造类型。在 C 语言中,构造类型有以下几种:数组类型、结构类型、联合类型。

3. 指针类型

指针是一种特殊的,同时又是具有重要作用的数据类型。其值用来表示某个量在内存存储器中的地址。虽然指针变量的取值类似于整型量,但这是两个类型完全不同的量,因此不能混为一谈。

4. 空类型

在调用函数值时,通常应向调用者返回一个函数值。这个返回的函数值是具有一定的数据类型的,应在函数定义及函数说明中给以说明,例如在例题中给出的 max 函数定义中,函数头为: `int max(int a, int b);` 其中“int”类型说明符即表示该函数的返回值为整型量。又如在例题中,使用了库函数 `sin`, 由于系统规定其函数返回值为双精度浮点型,因此在赋值语句 `s=sin(x);` 中, `s` 也必须是双精度浮点型,以便与 `sin` 函数的返回值一致。所以在说明部分,把 `s` 说明为双精度浮点型。但是,也有一类函数,调用后并不需要向调用者返回函数值,这种函数可以定义为“空类型”。其类型说明符为 `void`。

2.2 整数型数据

对于基本数据类型量,按其取值是否可改变又分为常量和变量两种。在程序执行过程

中，其值不发生改变的量称为常量，其值可变的量称为变量。它们可与数据类型结合起来分类。例如，可分为整型常量、整型变量、浮点常量、浮点变量、字符常量、字符变量、枚举常量、枚举变量。在程序中，常量是可以不经说明而直接引用的，而变量则必须先说明后使用。

整型量包括整型常量、整型变量。整型常量就是整常数。在C语言中，使用的整常数有八进制、十六进制和十进制三种。

2.2.1 整型常量

1. 十进制整常数

十进制整常数没有前缀。其数码为0~9。

以下各数是合法的十进制整常数：

237, -568, 65535, 1627;

以下各数不是合法的十进制整常数：

023, 不能有前导0;

23D, 含有非十进制数码。

2. 八进制整常数

八进制整常数必须以0开头，即以0作为八进制数的前缀。数码取值为0~7。八进制数通常是无符号数。

以下各数是合法的八进制数：

015(十进制为13) 0101(十进制为65) 0177777(十进制为65535)

以下各数不是合法的八进制数：

256(无前缀0) 03A2(包含了非八进制数码) -0127(出现了负号)

3. 十六进制整常数

十六进制整常数的前缀为0X或0x。其数码取值为0~9, A~F或a~f。

以下各数是合法的十六进制整常数：

0X2A(十进制为42), 0XA0(十进制为160), 0XFFFF(十进制为65535);

以下各数不是合法的十六进制整常数：

5A, 无前缀0X;

0X3H, 含有非十六进制数码。

在程序中是根据前缀来区分各种进制数的，因此，在书写常数时，不要把前缀弄错，以免造成结果不正确。

4. 整型常数

整型常数的后缀在16位字长的机器上，基本整型的长度也为16位，因此表示的数的范围也是有限定的。十进制无符号整常数的范围为0~65535，有符号数为-32768~+32767。八进制无符号数的表示范围为0~0177777。十六进制无符号数的表示范围为0X0~0XFFFF或0x0~0xFFFF。如果使用的数超过了上述范围，就必须用长整型数来表示。长整型数是用后缀“L”或“l”来表示的。例如：

十进制长整常数：158L(十进制为158), 358000L(十进制为-358000);

八进制长整常数：012L(十进制为 10)，077L(十进制为 63)，020000L (十进制为 65536)；

十六进制长整常数：0X15L(十进制为 21)，0XA5L(十进制为 165)，0X10000L(十进制为 65536)。

长整数 158L 和基本整常数 158 在数值上并无区别。但对于 158L，因为是长整型量，C 编译系统将为它分配 4 个字节存储空间。而对于 158，因为是基本整型，只分配 2 个字节的存储空间，因此，在运算和输出格式上要予以注意，避免出错。无符号数也可用后缀表示，整型常数的无符号数的后缀为“U”或“u”。例如：358u, 0x38Au, 235Lu 均为无符号数。前缀、后缀可同时使用，以表示各种类型的数。如 0XA5Lu 表示十六进制无符号长整数 A5，其十进制为 165。

2.2.2 整型变量

整型变量可分为以下几类：

1. 基本型

类型说明符为 int，在内存中占 2 个字节，其取值为基本整常数。

2. 短整型

类型说明符为 short int，所占字节和取值范围均与基本型相同。

3. 长整型

类型说明符为 long int 或 long，在内存中占 4 个字节，其取值为长整常数。

4. 无符号型

类型说明符为 unsigned。

无符号型又可与上述三种类型匹配而构成：

(1) 无符号基本型。类型说明符为 unsigned int 或 unsigned；

(2) 无符号短整型。类型说明符为 unsigned short；

(3) 无符号长整型。类型说明符为 unsigned long。

各种无符号类型量所占的内存空间字节数与相应的有符号类型量相同。但由于省去了符号位，故不能表示负数。下表列出了 Turbo C 中各类整型量所分配的内存字节数及数的表示范围。

类型说明符	数的范围	分配字节数
int	-32768~32767	■■
short int	-32768~32767	■■
signed int	-32768~32767	■■
unsigned int	0~65535	■■
long int	-2147483648~2147483647	■■■■
unsigned long	0~4294967295	■■■■

整型变量的说明：

变量说明的一般形式为：类型说明符 变量名标志符，变量名标志符，...；例如：

int a, b, c; (a, b, c 为整型变量)

long x, y; (x, y 为长整型变量)