



普通高等教育“十一五”国家级规划教材 (高职高专教育)
PUTONG GAODENG JIAOYU SHIYIWU GUOJIAJI GUIHUA JIAOCAI

CHENGXU SHEJI JICHU JIAOCHENG

程序设计基础教程

(C语言与数据结构) 配套教材

——学习指导与习题集 (第二版)

董杨琴 许秀林 主编



中国电力出版社

<http://jc.cepp.com.cn>



普通高等教育“十一五”国家级规划教材 (高职高专教育)
PUTONG GAODENG JIAOYU SHIYIWU GUOJIAJI GUIHUA JIAOCAI

CHENGXU SHEJI JICHU JIAOCHENG

程序设计基础教程

(C语言与数据结构) 配套教材

——学习指导与习题集 (第二版)

主 编 董杨琴 许秀林

副主编 黄晓亚

主 审 曹洪其 岳国英



中国电力出版社

<http://jc.cepp.com.cn>

内 容 提 要

本书是《程序设计基础教程（C语言与数据结构）》的配套教材，由基础知识及典型习题、各类计算机考试中与“C语言与数据结构”有关的考试样卷和附录三部分组成。

“基础知识及典型习题”部分作为《程序设计基础教程（C语言与数据机构）》的补充，分为理论知识、典型习题讲解和习题三部分；“各类计算机考试样卷”部分提供了与“C语言与数据结构”有关的考试样卷，包括全国计算机软件水平考试程序员下午部分试题、全国计算机等级考试二级C语言笔试和上机试卷、江苏省计算机等级考试二级C语言笔试和上机试卷；“附录”部分提供了 Turbo C 2.0 和 Visual C++ 6.0 集成编译环境，供读者在上机时参考使用。此外，还提供了“第1部分”中习题的参考答案、“各类计算机考试样卷”的参考答案，供读者参考使用。

本书重点突出，结构严谨，语言通俗易懂，讲解详细，特别适合高职高专计算机及相关专业的学生使用。

图书在版编目（CIP）数据

程序设计基础教程（C语言与数据结构）配套教材：学习指导与习题集 / 董杨琴，许秀林主编. —2版. —北京：中国电力出版社，2009

普通高等教育“十一五”国家级规划教材. 高职高专教育
ISBN 978-7-5083-9471-8

I. 程… II. ①董…②许… III. ①C语言—程序设计—高等学校：技术学校—教学参考资料②数据结构—高等学校：技术学校—教学参考资料 IV. TP312 TP311.12

中国版本图书馆 CIP 数据核字（2009）第 173988 号

中国电力出版社出版、发行

（北京三里河路 6 号 100044 <http://jc.cepp.com.cn>）

航远印刷有限公司印刷

各地新华书店经售

*

2005 年 9 月第 1 版

2009 年 12 月第二版 2009 年 12 月北京第二次印刷

787 毫米×1092 毫米 16 开本 20 印张 486 千字

印数 5001—8000 册 定价 32.00 元

敬告读者

本书封面贴有防伪标签，加热后中心图案消失
本书如有印装质量问题，我社发行部负责退换

版权专有 翻印必究

前 言

本书是《程序设计基础教程（C语言与数据结构）》的配套教材，由基础知识及典型习题、各类计算机考试中与“C语言与数据结构”有关的考试样卷、附录三部分组成。

“基础知识及典型习题”部分：作为《程序设计基础教程（C语言与数据结构）》的补充，分为理论知识、典型习题讲解、习题三部分。

“各类计算机考试样卷”部分提供了与“C语言与数据结构”有关的考试样卷，包括全国计算机软件水平考试程序员下午部分试题、全国计算机等级考试二级C语言笔试和上机试卷、江苏省计算机等级考试二级C语言笔试和上机试卷，为读者参加各类计算机考试提供学习练习或训练。

“附录”部分提供了 Turbo C 2.0 和 Visual C++ 6.0 集成编译环境，供读者在上机时参考使用。此外，还提供了“第1部分”中习题的参考答案、“各类计算机考试样卷”的参考答案，供读者参考使用。

本书由董杨琴、许秀林主编，各部分编写分工如下：理论部分和样卷部分由董杨琴编写，习题部分由黄晓亚编写，全书由董杨琴、许秀林提出编写大纲，董杨琴统稿，王琼瑶、吴素全参加了部分文字整理工作。全书由曹洪其和岳国英审稿。

本书是对江苏省高校特色专业——南通职业大学计算机应用技术专业进行专业教学改革成果，是该校江苏省精品课程“C语言与数据结构”指定教材，得到了该专业全体教师、校领导及教务处的大力支持，在此表示衷心的感谢。同时，也非常感谢中国电力出版社的全体人员，他们对本书的出版给予了很好的指导和有力的支持。

由于编写时间仓促，水平有限，书中不当之处，恳请专家们及广大读者批评指正。

编 者

2009年9月

目 录

前言

第 1 部分 基础知识及典型习题

绪论	1
第 1 章 程序设计数据类型及其运算	4
1.1 理论知识	4
1.2 典型例题讲解	12
习题 1	14
第 2 章 结构化程序设计的三种结构	23
2.1 理论知识	23
2.2 典型例题讲解	27
习题 2	32
第 3 章 函数	47
3.1 理论知识	47
3.2 典型例题讲解	53
习题 3	56
第 4 章 数组及结构体等构造类型	68
4.1 理论知识	68
4.2 典型例题讲解	79
习题 4	84
第 5 章 指针	100
5.1 理论知识	100
5.2 典型例题讲解	109
习题 5	113
第 6 章 文件	135
6.1 理论知识	135
6.2 典型例题讲解	139
习题 6	143
第 7 章 线性表	154
7.1 理论知识	154
7.2 典型例题讲解	166
习题 7	169
第 8 章 栈和队列	181
8.1 理论知识	181

8.2 典型例题讲解	192
习题 8	193
第 9 章 二叉树及其应用	197
9.1 理论知识	197
9.2 典型例题讲解	204
习题 9	206
第 10 章 图及其应用	211
10.1 理论知识	211
10.2 典型例题讲解	220
习题 10	221

第 2 部分 各类计算机考试样卷

2008 年下半年全国计算机软件水平考试程序员下午试题	224
2009 年上半年全国计算机软件水平考试程序员下午试题	229
2008 年 9 月全国计算机等级考试二级 C 语言笔试试卷	233
2009 年 3 月全国计算机等级考试二级 C 语言笔试试卷	242
2009 年 3 月全国计算机等级考试二级 C 语言上机试卷	252
2008 年春江苏省计算机等级考试二级 C 语言笔试试卷	254
2008 年秋江苏省计算机等级考试二级 C 语言上机试卷	261
2009 年春江苏省计算机等级考试二级 C 语言笔试试卷	263

第 3 部分 附 录

附录 A Turbo C 2.0 开发环境	270
附录 B Visual C++ 6.0 开发环境	288
附录 C 运算符及其优先级和结合性	298
附录 D 常用字符与 ASCII 编码对照表	299
附录 E 第 1 部分习题参考答案	300
附录 F 第 2 部分各类计算机考试样卷参考答案	307
参考文献	311

第 1 部分 基础知识及典型习题

绪 论

1. 几个与程序设计有关的概念

(1) 程序及其基本结构

程序是计算机为完成一个任务编制的指令集。在程序中,根据指令语句执行的次序不同,程序有三种基本结构:顺序结构、条件结构(也称为分支结构)和循环结构。顺序结构是指令语句按照语句顺序执行。分支结构是按照条件判断结构选择执行部分语句,即条件判断为真执行一个分支,条件判断为假,则执行另一个分支,如果存在多个分支,则称为多分支结构。循环结构是按循环次数或条件反复执行某一段语句,反复执行的语句称为循环体。循环结构中如果先判断条件,再执行循环体,则称为 while 循环;如果先执行循环体,再判断条件,则称为 until 循环。

一个程序由多个过程或函数组成,过程之间或函数之间存在调用和被调用的关系。对于两个具有调用关系的过程或函数,常称调用的过程或函数为主程序,被调用的过程或函数为子程序。主程序和子程序之间,通过过程名或函数名来调用,数据是通过参数来传递的,主程序的参数称为实在参数,子程序的参数称为形式参数。每个子程序尽量做到单入口、单出口,尽量不用全局变量。一个完整的程序只有唯一的系统入口主程序,在 C 语言中是 main 函数,它是所有函数的主函数。

(2) 算法以及描述算法的工具

算法是解决问题的一种方法或过程,它是计算机解决既定问题的一种描述,是程序的灵魂。从某种程度上说,程序设计就是算法的设计。一个算法必须具有五个基本特性:

- ① 输入。一个算法有 n ($n \geq 0$) 个初始数据的输入。
- ② 输出。一个算法必有一个或多个输出信息,并且输入信息与输出信息存在某种对应关系。
- ③ 有穷性。一个算法必须在有限操作步骤之后结束。
- ④ 确定性。一个相同的算法在不同环境中输入相同的数据,应该有相同的输出。
- ⑤ 可行性。一个算法的指令必须能在现有的计算机环境中正确运行,并且在预计可接受的时间内正常结束。

从计算的角度看,算法是数据的加工和计算的方法;从事务处理的角度看,算法是事务处理流程。算法设计必须要首先分析问题,提出解决问题的思路,然后再根据思路设计算法,将思路进一步明确和细化,并用形式化的工具来描述。描述算法的形式化工具有很多,常用的有自然语言、伪代码、流程图等。由于流程图较为直观,因而是算法描述的主要工具。流程图有传统流程图(FC)、盒图(NS)、问题分析图(PAD)等种类。

(3) 数据结构

数据(Data)是自然界中事物属性的数量描述。在计算机科学中,数据是对客观事物符

号表示的集合，是计算机程序加工的对象，其含义极为广泛，如数字、文字、声音、图像、图形等都属于数据范畴。数据的基本单位称为数据元素（Data Element），数据元素也称为结点、顶点、记录，它描述的是现实世界中客观存在的独立实体。通常数据元素又可分解为若干个数据项（Data Item），数据项是数据具有独立含义的最小单位。识别数据元素的数据项称为关键字（Keyword）。

数据结构是数据元素之间的相互关系。数据结构分为数据的物理结构和数据的逻辑结构。数据的物理结构是数据在存储器中的位置关系。数据的逻辑结构是根据运算规则来描述的数据间的相互关系。数据的物理结构与逻辑结构可能是一致的，也可以是不同的。在计算机科学中，数据结构主要研究数据的逻辑结构。

数据的逻辑结构有四种基本类型：集合结构、线性结构、树状结构（非线性结构）、网状结构（非线性结构）。

数据通常通过顺序、链接、索引、散列等存储方式存储在存储器中，即数据的物理结构有四种：顺序存储结构、链接存储结构、索引存储结构、散列存储结构。

2. 高级语言程序的编程过程

一般用高级语言编写的程序要经过编辑、编译、连接，生成可执行文件后才能运行。用C语言编写的程序也不例外。利用 Turbo C 2.0 或 Visual C++ 6.0 提供的集成开发环境，我们能完成C语言源程序的输入、修改、保存、编译、连接、执行的完整过程。在上述过程中，如果某一步出现错误，便不能执行下一步操作，必须改正了相关的错误以后才能继续，C语言程序的编程过程如图 0-1 所示。

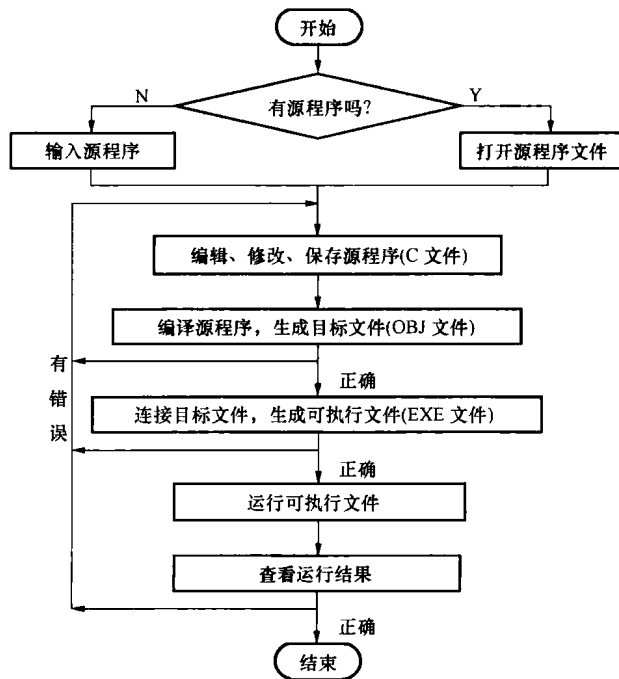


图 0-1 高级语言程序的编程过程

3. 良好的编程风格

程序设计风格是指一个人编制程序时所表现出来的特点、习惯、逻辑思路等。要形成良

好的程序设计风格，应从源程序的文档化、数据说明的方法、语句的构造、输入/输出等方面加以注意。结构化程序设计对文本格式的具体要求有：

- ① 对齐和缩进原则，即内层循环相对于外层循环要有缩进，形成锯齿结构。
- ② 程序的主要语句要有适当的注释。
- ③ 符号名的命名要“见名知意”。
- ④ 一行只写一条语句。
- ⑤ 输入数据前要有适当的提示，输出结果时要有说明。
- ⑥ 对所有的输入数据都要检验数据的合法性。

我们可以用欣赏的眼光来分析具有良好风格的程序。

```
/* **** */
/* 程序名: sampl_1.c */
/* 作者: dyq */
/* 编制时间: 2009年1月10日 */
/* 程序功能: 计算n的阶乘(用循环结构实现) */
/* **** */
#include "stdio.h"
void main()
{   int i,n;
    long factorial;
    printf("请输入n: ");
    do {
        scanf("%d",&n);
        if (n<0)
            printf("输入的数据小于0, 请重新输入.\n" );
    } while (n<0);          /*输入n, 直到大于等于0为止*/
    factorial=1;
    for (i=1;i<=n;i++)
        factorial*=i;
    printf("%d!=%ld\n", n, factorial );    /*以3!=6的形式输入结果*/
}
```

第 1 章 程序设计数据类型及其运算

1.1 理论知识

1.1.1 计算机中的数据及其类型

现实问题中的各种各样的“数据”在计算机中的存在形式主要有两种：常量和变量。计算机中的所有数据均是以常量或变量的形式存储在计算机的内存中。

各种数据在计算机内存中存储所需的空间各不相同，为了节约内存空间，提高内存空间的利用率，高级语言在处理不同类型的数据时，根据数据的实际应用，将数据划分成多种类型，每种数据类型在内存中占用不同的存储空间。

1. 标识符

标识符是指以字母或下划线开始的，由字母、数字、下划线组成的有效字符序列。在使用标识符时，请注意以下几点：

- ① 在 C 语言中，标识符是区分大小写的。
- ② 标识符的长度随着不同的编译环境有所差异，大部分编译系统默认标识符的长度为 8 个字符。
- ③ 在使用标识符命名变量时应使标识符能较好地表达变量的含义。
- ④ 使用标识符表示用户自定义的变量、函数等语法单位时，应注意不能使用系统已定义的关键字。

ANSI C 标准定义的关键字有 32 个。

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

这些关键字在 C 语言中起到标识数据类型、构成控制语句等功能，已为系统所定义，不能再用作用户自定义的标识符。

2. 类型的总体划分

根据实际编程的需要，C 语言中类型的划分总体上可分为：

- ① 基本类型。也可以称为原子类型，它不能再分解为其他类型。常见的有整型、实型（又称浮点型）、字符型、枚举型四种。
- ② 构造类型。由一种或多种基本类型的数据按照实际需要组合而成的类型。常见的有数组、结构体、共用体三种。
- ③ 指针类型。
- ④ 空类型。

基本类型是编程的基础，本章我们首先学习基本类型中的整型、实型（又称浮点型）、字符型，其他类型在后面的章节中将陆续学习。

3. 常量及其类型

所谓常量就是指在程序运行过程中内容不会发生变化的数据。对于常量，在程序运行过

程中,只能使用该常量的值而不能对它做任何修改。常量具有不需要事先定义、在需要的地方直接书写数值的特征。常量的表示方法很直接,根据数据的不同类型,其表示方法也不同。C语言中的常量有:整型常量、实型常量、字符型常量、字符串常量和符号常量等。

(1) 整型常量

C语言整型常量有三种表示方法:十进制整型、八进制整型和十六进制整型。其表示形式如表1-1所示。

表 1-1 整型常量的表示形式

进 制	表 示 方 法	样 例
十进制整型	以数字1、2、…、9中的一个数开头	123
八进制整型	以数字0开头	0123
十六进制整型	以0x或0X开头	0x123

(2) 实型常量

C语言中实型常量的表示方法有两种:

- ① 小数形式。由整数部分、小数点和小数部分所组成。例如123.0。
- ② 指数形式。采用科学计数法表示数据,由尾数、e或E和指数3个部分组成。例如 1.23×10^3 ,在C语言中可以写成1.23E3或1.23e3。

注意:在十进制小数形式表示中,小数点不可少;用指数形式表示时,e(E)的前后必须有数字,且e(E)的后面必须为整数。

(3) 字符型常量

字符型常量是用一对单引号括起来的单一字符,在计算机的存储中占据一个字节。单引号是定界符,它并不是字符型常量的一部分。例如,'A'、'9'、'#'等都是字符型常量。字符型常量在计算机中以它的ASCII码值形式进行存储。

由于字符型常量中的单引号已为定界符使用,另外还有一些控制字符(如制表符、回车换行字符等)不能直接表示,所以为了表达方便,C语言提供了转义字符表示法。转义字符表示法以反斜杠(\)开头,后面跟上相关的字符来表示特殊的字符。常用的转义字符如表1-2所示。

表 1-2 常用转义字符

字符形式	含 义	ASCII 码
\n	换行,将光标移到下一行开头	10
\t	水平制表(光标右移8列)	9
\b	退格,光标前移一列	8
\r	回车,光标移到本行首列	13
\f	换页,光标移到下页开头	12
\\	反斜杠字符	92
\'	单引号字符	39
\"	双引号字符	34
\ddd	3位八进制数所代表的字符	ddd ₈
\xhh	2位十六进制数所代表的字符	hh ₁₆

(4) 字符串常量

字符串是指由多个字符构成的一串字符。字符串常量的表示比较简单，使用一对双引号将字符串括起，即可构造字符串常量，双引号是字符串常量的定界符。

字符串常量简称字符串。C 语言规定：在每一个字符串的结尾加一个“字符串结束标志”（'\0'），以便系统据此判断字符串是否结束。字符串的长度就是字符的个数，因此，长度为 n 的字符串，在计算机的存储中占用 $n+1$ 个字节。

在字符串中，除了可以使用一般字符外，还可以使用转义字符。

(5) 符号常量

在 C 语言中，可对常量进行命名，即用符号代替常量。该符号叫做符号常量。符号常量通常由预处理命令 `#define` 来定义，而且一般用大写字母表示，以便与其他标识符相区别。符号常量要先定义后使用，定义的一般格式为：

```
#define 符号常量名 常量
```

例如：

```
#define PI 3.14159
```

4. 变量类型及其定义

所谓变量就是指在程序的运行过程中值可以改变的数据。在 C 语言中，要求对所有用到的变量作强制定义，也就是“先定义，后使用”。变量定义的一般格式是：

```
类型说明符 变量名表；
```

其中，类型说明符指定变量的数据类型。

(1) 整型变量

整型变量的基本类型符为 `int`。可根据数值的范围将整型变量定义为基本整型（以 `int` 表示）、短整型（以 `short int` 表示，或以 `short` 表示）和长整型（以 `long int` 表示，或以 `long` 表示）。在实际应用中，有些变量的值常常是正的（如学号、年龄等）。为了充分利用变量的表示范围，可以将变量定义为无符号类型。对上面三类都可以加上修饰符 `unsigned`，以指定是无符号数。如果加上修饰符 `signed`，则指定是有符号数。如果默认的话，则隐含为有符号数。因此，一共有六种整型变量。

- | | | |
|-----------|-----------------------|----------------------------|
| ① 有符号基本整型 | <code>[signed]</code> | <code>int</code> 。 |
| ② 无符号基本整型 | <code>unsigned</code> | <code>int</code> 。 |
| ③ 有符号短整型 | <code>[signed]</code> | <code>short [int]</code> 。 |
| ④ 无符号短整型 | <code>unsigned</code> | <code>short [int]</code> 。 |
| ⑤ 有符号长整型 | <code>[signed]</code> | <code>long [int]</code> 。 |
| ⑥ 无符号长整型 | <code>unsigned</code> | <code>long [int]</code> 。 |

利用这些整型类型，我们就可以定义整型变量了。例如：

```
int math, chinese, english;      /* 定义了 3 个基本整型变量 */
unsigned short c, d;             /* 定义 c、d 为无符号短整型 */
long e, f;                       /* 定义 e、f 为长整型 */
```

(2) 实型变量

在 C 语言中，实型变量分为单精度（`float` 型）、双精度（`double` 型）和长双精度（`long`

double 型)三类。对每一个实型变量都应在使用前加以定义。例如:

```
float   a, b;           /* 定义 a、b 为单精度实型 */
double  c, d;           /* 定义 c、d 为双精度实型 */
long double e, f;      /* 定义 e、f 为长双精度实型, 一般用得很少 */
```

(3) 字符型变量

字符型变量用来存放字符数据, 只能存放一个字符, 而不能存放一个字符串。在 C 语言中, 字符型变量只有一种, 用关键字 `char` 表示。字符型变量在内存中只占一个字节。字符型变量的定义形式如下:

```
char  ch1, ch2;        /* 定义 ch1、ch2 为字符型 */
```

5. 变量赋初值

程序中常需要对一些变量预先设置初值。C 语言允许在定义变量的同时对变量进行初始化。例如:

```
int    math=90;        /* 定义 math 为基本整型变量, 初值为 90 */
char  ch1= 'A';        /* 定义 ch1 为字符型变量, 初值为 'A' */
```

也可以对被定义的多个变量的一部分赋初值。例如:

```
int    a, b, c=5;      /* 定义 a、b、c 为基本整型变量, c 初值为 5 */
```

1.1.2 数据的输入和输出

在 C 语言中, 没有提供输入/输出的语句, 所有输入/输出操作均使用 C 语言提供的标准输入/输出函数来完成。为了使用方便, 已将使用标准输入/输出函数所必需的信息归类到一个名为 `stdio.h` 的头文件中, 使用这些 I/O 标准函数时, 只需在程序前面用编译预处理的包含命令将 `stdio.h` 头文件包含进来即可 (`#include <stdio.h>`)。

1. 格式输出函数 printf

`printf` 函数主要用于通过标准输出设备 (如显示器) 按照指定的格式输出数据。该函数不仅能根据需要输出多个数据项, 而且可以根据需要设置每个输出项的显示格式, 在编程中使用频率较高。下面我们重点来学习该函数的使用。

格式: `printf(格式控制字符串, 输出项表);`

功能: 按格式控制字符串指定的格式将输出项表中的内容输出到输出设备。

说明:

1) 格式控制字符串是由双引号括起的字符串, 包括:

① 格式说明符: 由 % 和格式字符组成, 将要输出的数据转换为指定的格式。在输出时由输出项表中相应的输出项代替。格式字符的使用方法请参见表 1-3。

表 1-3 格式说明符的使用

格式字符	说 明
d, i	以带符号的十进制形式输出整数 (正数不输出符号)
u	以无符号十进制形式输出整数
o	以八进制无符号形式输出整数 (不输出前导符 0)
X, x	以十六进制无符号形式输出整数 (不输出前导符 0x)
c	以字符形式输出单个字符

续表

格式字符	说 明
s	输出字符串
f	以小数形式输出单、双精度实数，默认输出 6 位小数
E/e	以标准指数形式输出单、双精度实数，小数位数 6 位
G/g	选用%f或%e格式中宽度较短的格式输出，不输出无意义的 0

② 附加格式说明符：用来对输出项的宽度和对齐方式等进行说明，参见表 1-4。

表 1-4 附加格式说明符的使用

格式字符	说 明
字母 l	用于长整型，可使用在格式符 d、o、x、u 前面
^m (代表一个正整数)	指域宽，表示数据的最小宽度
ⁿ (代表一个正整数)	指精度，对于实数表示输出 n 位小数；对字符串表示截取的字符个数
-	在域宽前加一，表示输出的数字或字符在输出时左对齐；否则为右对齐
0	在域宽前加 0，表示输出数字前的空位用 0 填补；否则用空格填补

③ 普通字符：为原样输出的字符，使程序的输出更加直观。

2) 输出项表是由逗号分隔的多个输出项组成。输出项可以是常量、变量或表达式。输出项和格式控制字符串中的格式说明类型要一致。

2. 格式输入函数 scanf

scanf 函数主要用于从标准输入设备（如键盘）按照指定的格式读取数据，并给指定的变量赋值。该函数基本上能完成各种类型数据的输入。

格式：scanf(格式控制字符串，输入变量地址列表)；

功能：按格式控制字符串指定的格式从标准输入设备读取数据给指定的变量。

说明：

① 格式控制字符串。标识本次输入过程中读取数据的个数和类型，使用"%C"的方式来构造格式字符串，C 称为格式字符，如 int 型数据是 d 或 i，具体如表 1-5 所示。多个输入变量就构成"%C%C%C..."的格式字符串，例如要输入一个字符、一个整数、一个单精度实数，则格式字符串可以表示成"%c%d%f"。

表 1-5 格式字符的使用

格式字符	说 明	格式字符	说 明
d、i	用来输入有符号的十进制整数	c	用来输入单个字符
u	用来输入无符号的十进制整数	s	用来输入字符串，输入的字符串保存在字符数组中
o	用来输入无符号的八进制数	f	用来输入实数，可以是小数或指数形式
x、X	用来输入无符号的十六进制整数	e、E、g、G	与 f 相同

② 输入变量地址列表。由逗号分隔的一个或多个接收数据的变量的地址构成的地址列表，在编程时应使地址列表中所含变量的类型和个数与格式字符串相一致。

在使用键盘输入数据时，每个输入项输入结束时可以使用 Enter 或 Tab 键结束当前项的输入并开始输入下一项。如果在格式字符串中含有除格式字符以外的非格式字符，则在输入

完相关变量的值以后输入该非格式字符。

1.1.3 数据的处理与运算

在计算机中，对数据的各种处理均可以认为是数据运算。数据运算的方法有多种，既可以使用各种运算符构建表达式来进行比较简单的运算，又可以利用系统函数或自定义函数来完成比较复杂的运算。

所谓运算符是指C语言中用来表示各种运算的规定符号，特定的符号表示了特定的运算类型以及特定的运算规则。C语言规定了多种运算符，每个运算符都有优先级和结合性。优先级是指在参加运算时运算的先后次序，优先级高的先参加运算。结合性是指运算符与运算数据的结合方向，一般分为“自左至右”和“自右至左”两种。

所谓表达式是指由常量、变量、函数和运算符构成的，能进行计算并有唯一确定值的式子。

1. 各种运算符

(1) 算术运算符

C语言中的算术运算符比较多，数量超过了我们数学概念中的算术运算，其算术运算符及其运算规则如表1-6所示（表中箭头指示优先级的方向是由高到低，下同）。

表1-6 算术运算符的使用

运算符	运算方法	运算结果	优先级	结合性	说明
++	a++或++a	a的值增1	相同	自右至左	a++(a-): 称为后置运算符, 表达式的值为a原来的值, 然后a的值增加(减少)1 ++a(-a): 称为前置运算符, 先增加(减少)a的值, 并将新值作为表达式的值
--	a--或--a	a的值减1			
*	a*b	计算a、b的积	相同	自左至右	a、b只能是整数
/	a/b	计算a、b的商			
%	a%b	计算a对b的余数			
+	a+b	计算a、b的和	相同		
-	a-b	计算a、b的差			

(2) 关系运算符

关系运算相当于数学中的不等式，用来对数据进行比较，以确定相互之间的关系。关系表达式的运算结果只有两种：“成立”与“不成立”，如果“成立”则计算结果为逻辑真“1”，否则就为逻辑假“0”。关系运算的运算符及运算规则如表1-7所示。

表1-7 关系运算符的使用

运算符	运算方法	运算结果	优先级	结合性	说明
<	a<b	a小于b则为1	相同	自左至右	参加运算数据a、b应为可比较类型数据
<=	a<=b	a小于或等于b则为1			
>	a>b	a大于b则为1			
>=	a>=b	a大于或等于b则为1	相同		
==	a==b	a等于b则为1			
!=	a!=b	a不等于b则为1			

(3) 逻辑运算符

逻辑运算是计算机中的一种基本运算，参加运算的数据是表示逻辑值的“真”和“假”，运算结果也是逻辑值，常用于表达比较复杂的条件，在后面讲述的条件语句和循环语句中也被大量使用。逻辑运算的类型有多种，但任何复杂的逻辑运算均由三种基本逻辑运算组合而成，其运算符和运算规则如表 1-8 所示。

在 C 语言中，逻辑值的表示有约定俗成的规则：参加逻辑运算的数据只要不是 0，就视为逻辑“真”，0 为逻辑假，即“非 0 即真”；逻辑运算的结果“1”表示逻辑“真”，“0”表示逻辑“假”。

在计算逻辑表达式的值的过程中，为了提高计算的效率，C 程序在编译时，如果该表达式的值已能确定，则中止该表达式的后继计算，直接得到该表达式的值。

表 1-8 逻辑运算符的使用

运算符	运算方法	运算结果	优先级	结合性	说明
!	!a	取 a 的反	↓	自右至左	参加运算数据 a、b 可为任何类型数据
&&	a&& b	a 和 b 均为真则为 1, 否则为 0		自左至右	
	a b	a 和 b 均为假则为 0, 否则为 1			

(4) 位运算符

位运算是指进行二进制位的运算。在系统软件中，常要处理二进制位的问题。位运算符和运算规则如表 1-9 所示。

表 1-9 位运算符的使用

运算符	运算方法	运算结果	优先级	结合性	说明	
~	~a	将 a 按位取反	↓	自右至左	a、b 应为整型或字符型数据	
<<	a<<b	将 a 左移 b 位		相同		自左至右
>>	a>>b	将 a 右移 b 位				
&	a&b	将 a 和 b 按位进行与运算				
^	a^b	将 a 和 b 按位进行异或运算				
	a b	将 a 和 b 按位进行或运算				

(5) 赋值运算符

通过上面的学习，我们可以通过各种运算对数据进行运算构成表达式，从而得到表达式的值。如果我们希望将表达式的值保存到相应的变量中，则可以使用赋值运算符来实现赋值运算，从而构造赋值表达式。C 语言中赋值运算符的运算规则如表 1-10 所示。

表 1-10 赋值运算符的使用

运算符	运算方法	运算结果	优先级	结合性	说明
=	a=b	将 b 的值存在变量 a 中	相同	自右至左	a 必须为变量, b 为表达式
+=	a+=b	将 a+b 的值存在变量 a 中			
-=	a-=b	将 a-b 的值存在变量 a 中			
=	a=b	将 a*b 的值存在变量 a 中			
/=	a/=b	将 a/b 的值存在变量 a 中			

续表

运算符	运算方法	运算结果	优先级	结合性	说明
%=	a%=b	将 a%b 的值存在变量 a 中	相同	自右至左	a 必须为变量, b 为表达式
<<=	a<<=b	将 a<<b 的值存在变量 a 中			
>>=	a>>=b	将 a>>b 的值存在变量 a 中			
&=	a&=b	将 a&b 的值存在变量 a 中			
=	a =b	将 a b 的值存在变量 a 中			
^=	a^=b	将 a^b 的值存在变量 a 中			

由赋值运算符将一个变量和一个表达式连接起来的式子称为赋值表达式。其一般格式为：

<变量><赋值运算符><表达式>

对赋值表达式求解的过程是：先计算赋值运算符右侧的表达式值，然后将计算结果赋给左侧的变量。赋值表达式的值就是被赋值的变量的值。

(6) 条件运算符

条件运算符的使用格式为：

表达式 1 ? 表达式 2 : 表达式 3

上述格式构成一个条件表达式，当表达式 1 的值为“真”时，表达式 2 的值为表达式的值，否则表达式 3 的值为表达式的值。

(7) 逗号运算符

逗号运算符通过逗号将多个子表达式加以分隔，构成一个逗号表达式。逗号表达式的值为各子表达式中最右边表达式的值。

2. 运算符的复合运算

在一个表达式中，可以含有多种类型的运算符，从而导致一个表达式可以由多种运算符所描述的多种类型子表达式构成的，其运算过程中不仅需要考虑相同类型运算符的优先级和结合性，而且还要考虑不同类型运算符的优先级以及不同类型数据间运算的数据转换。

首先，在掌握了同类型间运算符的优先级和结合性的基础上，要进一步掌握不同类型运算符的优先级关系。常见的各类运算符的优先级参见附录 A。

其次，对于表达式中参加运算的数据而言，只要满足运算符的要求就可以参与运算，但不同类型的数据在参与运算的过程中，为了保证运算的精度，要先将不同类型的数据转换成同一类型的数据再进行运算。转换规则如图 1-1 所示。

在图 1-1 中，横向左箭头表示必定的转换，纵向箭头表示当运算对象为不同类型时转换的方向，其类型转换由系统自动完成。

最后，对于赋值运算，当将赋值表达式右侧表达式的值赋予左侧变量时，按照左侧变量的类型进行类型转换，但右侧表达式的求值按照图 1-1 的转换规则进行。

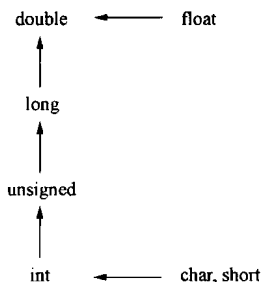


图 1-1 不同类型数据转换规则

3. 强制类型转换

可以利用强制类型转换运算符将一个表达式转换为所需类型。其一般形式为：

(类型名)(表达式)