



普通高等院校“十三五”规划教材·计算机精品课程主讲教材

C语言程序设计

主编◎王爱平 刘经天

C YUYAN CHENGXUSHEJI



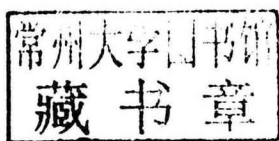
电子科技大学出版社



普通高等院校“十三五”规划教材·计算机精品课程主讲教材

C语言程序设计

主 编 王爱平 刘经天
副主编 刘华敏 殷荣网 周 飞
谭同德 关秀芬



电子科技大学出版社

图书在版编目 (CIP) 数据

C 语言程序设计 / 王爱平, 刘经天主编. -- 成都 :

电子科技大学出版社, 2017.7

ISBN 978-7-5647-4583-7

I. ①C… II. ①王… ②刘… III. ①C 语言—程序设计—教材 IV. ①TP312.8

中国版本图书馆 CIP 数据核字 (2017) 第 127542 号

C 语言程序设计

王爱平 刘经天 主编

策划编辑 杨仪玮

责任编辑 杨仪玮

出版发行 电子科技大学出版社

成都市一环路东一段 159 号电子信息产业大厦九楼 邮编 610051

主 页 www.uestcp.com.cn

服务电话 028-83203399

邮购电话 028-83201495

印 刷 北京荣玉印刷有限公司

成品尺寸 185mm×260mm

印 张 19.75

字 数 478 千字

版 次 2017 年 7 月第一版

印 次 2017 年 7 月第一次印刷

书 号 ISBN 978-7-5647-4583-7

定 价 42.00 元

版权所有, 侵权必究

前 言

C 语言自诞生以来就因其功能丰富、表达力强、程序效率高、可移植性好而成为被国内外广泛使用的一种计算机语言。C 语言既具有低级语言的特点又具备高级语言的优点,因此许多应用汇编语言编写的软件也可以用 C 语言编写(如 UNIX 操作系统),但是对于初学者来说,学习和使用 C 语言比学习和使用汇编语言要容易得多。

近年来,随着面向对象的计算机语言陆续问世和广泛使用,许多人认为面向过程的 C 语言过时了。这显然是一种误解,无论作为计算机程序的使用或开发人员,都不应把面向对象和面向过程对立起来,在面向对象程序设计中仍然需要面向过程的知识 and 技能,面向过程程序设计仍然是计算机工作者的基本功之一。

因此国内外许多高等院校不仅在计算机专业开设 C 语言相关课程,在非计算机专业也开设了 C 语言课程。全国计算机应用技术证书考试(NIT)、全国计算机等级考试、各地区大学生计算机统一考试均将 C 语言列入了考试范围。更重要的是,许多企事业单位应用 C 语言编写应用软件,学习 C 语言已经成为广大计算机应用人员和青年学子的迫切要求。

但由于 C 语言涉及的概念和规则比较繁杂,运用特别灵活,并且在实践中比较容易出错,不少学习者在开始学习时感到比较困难。因此编写适合于初学者的 C 语言程序设计教材,培养学生思考问题、分析问题和解决问题的水平和能力,提高其计算机的应用能力,既非必要,又非常重要。

本书集成了多位长期从事 C 语言教学工作的一线教师的经验总结,并参考大量国内外相关资料编写而成,系统阐述了 C 语言程序设计的基本概念、结构化编程思想和解决实际问题的方法技巧。突出了 C 语言在计算机相关学科中的基础地位,在整体架构上精心构思,在章节结构上合理安排,从概念到方法,在使用中将概念、思想和方法相结合,力求将复杂的概念用简洁浅显的语言来描述,做到深入浅出。

本书内容丰富、结构紧凑、概念阐述清楚、注重能力的培养,是一本内容全面的教材,可作为高等院校各专业计算机程序设计教学用书,以及计算机爱好者自学用书和各类工程技术人员的参考书。内容共分为 11 章,各章节的内容如下:

第 1 章 介绍 C 语言,包括 C 语言编程环境、C 程序的语法及结构。

第 2 章 介绍基本数据类型和运算符,包括常量和变量的知识、运算符、表达式及数据类型的转换。

第 3 章 介绍顺序和选择结构程序设计,包括数据的输入、输出和三种基本结构编程知识。

第 4 章 介绍循环结构程序设计,包括循环语句及嵌套和跳转语句编程。

第 5 章 介绍数组,包括不同类型的数组和使用数组的程序设计方法。

第6章 介绍函数和模块设计,包括函数的定义和调用、函数的嵌套调用、递归调用、内部函数和外部函数。

第7章 介绍指针,包括指针与一维数组、指针与多维数组、指针数组、多级指针及指针与函数。

第8章 介绍结构体与联合体,包括结构体、联合体和其他自定义数据类型。

第9章 介绍预处理和标准函数,包括预处理命令和输入/输出标准函数。

第10章 介绍文件,包括文件的基本概念及基本操作。

第11章 介绍数据结构和数据抽象,包括数据结构、线性表、堆栈和队列。

本书由王爱平、刘经天担任主编,其中第1章、第2章和第11章由刘华敏老师负责编写,第3章至第5章由周飞老师负责编写,第6章、第7章由刘经天老师负责编写,第8章至第10章由殷荣网老师负责编写,附录由谭同德、关秀芬老师共同编写。全书由安徽大学王爱平教授负责主审及统稿。

由于作者水平有限,加之时间仓促,错误与疏漏在所难免,敬请读者不吝赐正。

编 者

2017年7月



目 录

第 1 章 C 语言概述	1
1.1 C 语言的发展与特点	1
1.2 与程序设计语言相关的概念	3
1.3 C 程序的结构描述与分析	7
1.4 Visual C++ 6.0 上机简介	10
小 结	14
习 题	15
第 2 章 基本数据类型和运算符	17
2.1 常 量	17
2.2 变 量	22
2.3 运算符和表达式	27
2.4 数据类型的转换	39
小 结	42
习 题	42
第 3 章 顺序和选择结构程序设计	46
3.1 程序设计概述	46
3.2 数据的输入和输出	48
3.3 三种基本结构	54
3.4 if 选择结构语句	55
3.5 switch 选择结构语句	61
小 结	64
习 题	65

第 4 章 循环结构程序设计	69
4.1 循环语句	69
4.2 跳转语句	77
4.3 循环语句的嵌套	82
4.4 程序设计举例	85
小 结	89
习 题	89
第 5 章 数 组	92
5.1 一维数组	92
5.2 二维数组	97
5.3 字符数组	103
5.4 使用数组的程序设计方法	115
5.5 程序设计举例	121
小 结	126
习 题	126
第 6 章 函数和模块设计	131
6.1 结构化程序设计	131
6.2 函数的定义和调用	133
6.3 函数的嵌套调用和递归调用	150
6.4 作用域和存储类型	156
6.5 内部函数和外部函数	165
6.6 模块化程序设计	167
6.7 程序设计举例	169
小 结	175
习 题	176
第 7 章 指 针	179
7.1 指针的概念	179
7.2 指针变量作为函数参数	187
7.3 指针与一维数组	190
7.4 指针与多维数组	198
7.5 指针数组和多级指针	202
7.6 指针与函数	208
7.7 命令行参数	214

小 结	215
习 题	217
第 8 章 结构体与联合体	219
8.1 结构体	219
8.2 联合体	240
8.3 其他自定义数据类型	242
小 结	244
习 题	244
第 9 章 预处理和标准函数	246
9.1 预处理命令	246
9.2 输入/输出标准函数	252
小 结	258
习 题	258
第 10 章 文 件	259
10.1 文件的基本概念	259
10.2 文件的基本操作	260
10.3 程序设计举例	269
小 结	272
习 题	273
第 11 章 数据结构和数据抽象	274
11.1 数据抽象	274
11.2 线性表	275
11.3 堆 栈	280
11.4 队 列	282
小 结	286
习 题	287
附录 1 ASCII 代码对照表	288
附录 2 C 库函数	289
附录 3 Debugger 调试器使用简介	297
参考文献	305

第 1 章 C 语言概述

程序设计语言,通常简称为编程语言,是一组用来定义计算机程序的语法规则,用来向计算机发出指令,让程序员准确地定义计算机所需要使用的数据,并精确地定义在不同情况下所应当采取的行动。

C 语言是 Combined Language(组合语言)的简称,是一种被广泛应用的计算机程序设计语言,它既可以用来编写系统软件,也可以用来编写应用软件。

为了使读者能逐步地从简单的模仿中体会到程序设计的基本思想和方法,本章将简要介绍 C 语言的特点、语法,C 语言源程序的结构及运行环境等知识。

1.1 C 语言的发展与特点



1.1.1 C 语言的发展

早期的系统软件(包括操作系统)主要用汇编语言编写,因而程序与计算机硬件的关系十分密切,致使程序的编写难度大、可读性差且难于移植。为顺应时代的发展,需有一种与计算机硬件关系不紧密的高级语言(High-level Programming Language)用于编程。

早在 1971 年,美国电话与电报公司(AT&T)贝尔实验室的 Dennis Ritchie 就在早期的编程语言 BCPL(Basic Combined Programming Language, B 语言)基础上编写了 C 语言,并于 1972 年首次在 UNIX 操作系统的 DEC PDP-11 计算机上使用。1973 年,他与 Ken Thompson 用 C 语言重写了 UNIX 操作系统,使之成为 UNIX 发展的基础。之后,C 语言又发展成为通用程序设计语言。1983 年,美国国家标准化协会(ANSI)成立了一个委员会,制定了 C 语言标准(ANSI C),为 C 语言的进一步发展奠定了良好的基础。

1995 年,C 语言中增添了一些新的函数,使之具有了 C++的一些特征,使 C89 成为 C++的子集。1999 年推出的 C99 在基本保留 C 语言特征的基础上,增加了一系列面向对象的新特征。C 语言也就从面向过程的语言发展成为面向对象的语言。

C 语言是 C++的基础,C++语言和 C 语言在很多方面是兼容的,因此,掌握了 C 语言,可为将来学习 C++打下坚实的基础。本教材使用 Visual C++6.0 作为 C 语言程序的运行环境。

1.1.2 C 语言的特点

C 语言之所以能存在和发展并具有强大的生命力,主要是因为它具有强大的功能。

1. C 语言简洁、紧凑、使用方便、灵活

C 语言一共只有 32 个关键字、9 种控制语句,程序书写形式自由,语法控制不严格,表达式简练、灵活、实用。

2. 运算符丰富

C 语言中共有 34 个运算符,它们与丰富的数据类型相结合,构成了各种各样的表达式,实现了在其他高级语言中难以实现的各种复杂运算。

3. 数据结构丰富

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、联合体类型等,能用来实现各种复杂的数据类型的运算。尤其是指针类型数据的引入,使程序运行效率更高。另外,C 语言具有强大的图形功能,支持多种显示器和驱动器,且计算功能、逻辑判断功能强大。

4. C 语言是结构式语言

结构式语言的显著特点是程序代码模块化。C 语言的主要成分是函数,函数是 C 语言程序的基本结构单位,函数之间彼此独立,程序的许多操作可由不同功能的函数有机组成实现,从而达到结构化程序设计中模块的要求。另外,C 语言还提供了一套完整的控制语句(如顺序、分支、循环)和构造数据类型(如结构、数组、指针),使程序流程与数据描述也具有了良好的结构性。C 语言的这种结构化方式使程序层次更清晰,使用、维护及调试更方便。

5. C 语法限制不太严格、程序设计自由度大

一般的高级语言语法检查比较严,能够检查出几乎所有的语法错误。而 C 语言则放宽了语法检查,允许程序编写者有较大的自由度。例如:对数组下标越界不做检查。因此,在程序设计中,程序员不要过分依赖编译器的语法检查。

6. C 语言允许直接访问物理地址

C 语言既具有高级语言的特性,又具有低级语言的许多功能,能够像汇编语言一样对位、字节和地址进行操作,还可以用来编写系统软件。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。有人把 C 语言称为“高级语言中的低级语言”。

7. C 语言程序生成代码质量高

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. C 语言适用范围大、可移植性好

C 语言编写的程序中没有依赖于硬件的输入输出语句,程序的输入输出功能是通过调用输入输出函数实现的,而这些函数是由系统提供的独立于 C 语言的程序模块,所以编写好的 C 源程序基本上不做修改就可以用于各种型号的计算机和各种操作系统,从而便于在硬



件结构不同的计算机之间实现程序的移植。

1.2 与程序设计语言相关的概念

一般把用程序设计语言编写的未经编译的程序称为源程序(Source Code, 源代码), 而源程序的编写必须符合相应语言的语法(Grammar)。那么, 从语法的角度来说, 源程序实际上是一个字符序列。这些字符序列按顺序分别组成了一系列的“单词”。这些“单词”是为了按照一定的语法规则构成语言的各种成分而规定的。下面分别介绍 C 语言中的一些常用词汇和语法单位。



1.2.1 C 语言中的词汇

1. C 语言字符集

组成 C 语言源程序代码的基本字符称为 C 语言字符集, 它是构成 C 语言的基本元素。C 语言允许使用的基本字符如下。

- (1) 大小写英文字符: A~Z, a~z。
- (2) 数字字符: 0~9。
- (3) 特殊字符: + = - _ (下划线) () * & ^ % # ! , . ; : ? ' " ~ | / < > { } [] 。
- (4) 不可打印的字符: 空格、换行符、制表符、响铃符。

一般的 C 语言源程序仅仅包含以上字符集中的字符, 在具体的 C 语言编译系统中可对上述字符集合加以扩充。

2. 关键字

关键字是具有特定含义的、专门用来说明 C 语言的特定成分的一类单词。例如, 关键字 int 用来定义整型变量, 而关键字 float 则用来定义实型变量。C 语言的关键字都用小写字母书写, 不能用大写字母书写。例如, 关键字 int 不能写成 Int。由于每个关键字都有特定的含义, 所以不能作为用户程序中的变量名和函数名等, 否则会产生编译错误。在 C89 标准中共有 32 个关键字:

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

在新的 C99 标准中, 又增加了 5 个关键字:

```
_Bool _Complex _Imaginary inline restrict
```

3. 标识符

计算机程序处理的对象是数据, 程序是用来描述数据处理的过程。在程序中, 通过名字建立对象定义与使用的关系。为了满足这种需要, 每种程序语言都规定了在程序中名字描

述的规则。在 C 语言中用于标识名字的有效字符序列称为标识符,并对标识符做了如下规定。

(1) 标识符的第一个字符必须是英文字母或下划线(`_`)。

(2) 如果第一个字符后面还有字符序列,则它应是英文字母、下划线符或数字组成的序列。标识符中的英文字母大小写是有区别的,如标识符 `abc` 与标识符 `ABC` 不相同。为了便于读者对标识符有进一步的认识,下面列举若干正确的标识符和不正确的标识符。

正确的标识符:

`Abc` `abc` `_Abc` `_4a5`

不正确的标识符:

`A?` (含有不合法字符“?”)

`2abc` (第一个字符不允许为数字)

`a b` (标识符中不允许有空格)

`yes/no` (含有不合法字符“/”)

`πr` (“ π ”为不合法字符)

标识符中有效字符个数(也称长度)视系统不同而不同。例如,Turbo C 规定前 32 个字符有效,超过的部分忽略。比如,对于 8 个字符有效的标识符而言,`identifi` 与 `identifier` 被视为同一标识符,因后者中的 `er` 已被忽略。

后面还将讲到,标识符还可用来为变量、符号常量、数组、函数等取名。使用时,标识符的选择由程序员自定,但是不能与关键字相同。另外,为了增加程序的可读性,选择标识符时应遵循“见名知义”的原则,即选择描述性的标识符,标识符应尽量与所要命名的对象间有一定的联系,以助于识别和记忆。例如:

`length` (表示长度)

`time` (表示时间)

`pi` (表示圆周率 π)

4. 保留标识符

保留标识符是系统保留的一部分标识符,通常用于系统定义和标准库函数的名字。例如,以下划线开始的标识符通常用于定义系统变量。虽然它们也是合法的标识符,但用作一般标识符时可能会出现运行错误,因此不能使用这些标识符来定义自己的变量。

5. 注释

在 C 语言程序中,注释部分的格式是:“`/* 注释内容 */`”“`//注释内容`”。

`/* 注释内容 */` 对多行语句进行注释

`//注释内容` 对单行语句进行注释

例如:“`/* 学生成绩管理程序 */`”“`//My c program`”。

注释不是程序代码,只是对程序解释说明的一种标注,帮助用户理解程序,提高程序的阅读效率。程序编译时会把注释部分作为空白符跳过而不予处理。

使用注释需要注意三点。

(1) 注释不允许嵌套使用。例如:`/* test 1/* test 2 */test 3 */`,编译器解释时,只把



/* test 1/* test 2 */看成注释,后面的 test 3 */无法解释,错误。

(2)用“/* */”形式表示注释时,可以跨行,也可以穿插在一行代码中。

(3)用“//.....”形式注释内容时,表示从//开始直到该行末尾的所有字符都属于注释,这种注释不能跨行,也不能穿插在一行代码中间,这是从 C++借鉴的语法。

1.2.2 C语言的主要语法单位

1. 变量声明

在 C 语言中,变量使用之前必须先声明。在声明变量的时候一定要对其类型加以说明,变量类型的不同,说明其在计算机内存中所占的存储空间大小也不同。

2. 表达式

由运算符及其运算对象可以组成形形色色的表达式,如: $3.14 * 3 * \sin(x)$ 。表达式中的运算符有运算符优先级,如:表达式 $2+3 * 4-4/4$ 中,应先执行运算符 $*$ 和 $/$,再执行运算符 $+$ 和 $-$ 。

3. 语句

语句是程序最基本的执行单位,程序的功能就是通过执行一系列的语句来实现的。C 语言提供了多种语句,大致可分为五类:表达式语句、复合语句、空语句、控制语句、函数调用语句。

(1) 表达式语句

表达式语句由表达式末尾加上分号“;”组成。其一般格式为:

表达式;

执行表达式语句就是计算表达式的值。例如:

`m=2;    //赋值语句`

`m+n;    //算术表达式语句`

(2) 复合语句

把多个相关语句用一对花括号“{}”括起来,组成的一个语句就称为复合语句。在程序中,应把复合语句当作是单条语句,而不是多条语句。例如:

```
{
    temp=x;      /* 将 x 的值赋予 temp */
    x=y;         /* 将 y 的值赋予 x */
    y=temp;      /* 将 temp 的值赋予 y */
}
```

这就是一条复合语句。复合语句内的各条语句都必须以分号“;”结尾,但是在花括号“{}”外却不能加分号。

(3) 空语句

只有分号“;”组成的语句称为空语句,空语句是什么也不执行的语句。

例如:`while( getchar() != '\n' );` 本语句的功能是,只要从键盘输入的字符不是回车则重新输入。这里的循环体为空语句。

1.2.3 控制语句与函数

1. 控制语句

控制语句用于控制程序的流程,以实现程序的各种结构方式。C 语言有 9 个控制语句,可分成以下三类。

(1) 条件判断语句,包括 if 语句和 switch 语句。

(2) 循环语句,包括 for 语句、while 语句和 do-while 语句。

(3) 跳转语句,包括 break 语句、continue 语句、return 语句和 goto 语句。其中, goto 语句应尽量少用,因为这会给结构化程序设计带来诸多不便,滥用它会使程序流程无规律、可读性差。

2. 函数调用语句

(1) 函数定义

函数是完成特定任务的独立模块,是 C 语言唯一的子程序形式。函数的目的通常是接收 0 个或多个数据(称为函数的参数),并返回 0 个或 1 个结果(称为函数的返回值)。函数的使用主要涉及函数的定义与调用。

函数定义的主要内容是通过编写一系列语句来规定其所完成的功能。完整的函数定义涉及函数头和函数体。其中,函数头包括函数的返回值类型、函数名、参数类型;而函数体是一个程序模块,规定了该函数所具有的功能。以下是函数定义的一个简单例子。

```
int max(int m,int n) /* 函数头:函数类型说明符 函数名(函数参数列表) */
{
    /* 函数体的开始 */
    int x;          /* 声明一个整型变量 x */
    if(m>n)
    {
        x=m;
    }
    else
    {
        x=n;
    }
    /* 判断 m,n 的大小,将其中较大者的值赋予变量 x */
    return x;       /* 结束函数调用,并返回变量 x 的值 */
}                  /* 函数体的结束 */
```

(2) 函数的调用

函数调用是通过传递函数的参数并执行函数定义所规定的程序的过程。函数调用语句由函数名、实际参数加上分号“;”组成。其一般格式为:

函数名(实际参数表);

执行函数调用语句就是调用函数,并把实际参数赋予被调用函数定义中的形式参数,然后执行被调函数体中的语句,求取函数值。

(3) 输入/输出函数

C 语言没有输入输出语句,它通过调用系统库函数中的有关函数(如 printf()、scanf() 函数)实现数据的输入和输出,这种处理方式对 C 语言在不同硬件平台上的可移植性提供了良好的基础。



1.3 C 程序的结构描述与分析

用C语言编写的程序称为C语言源程序,简称为C程序。为了说明C语言源程序的结构特点,先看以下几个程序。这几个程序由易到难,虽然有关内容还未介绍,但可以从中学了解到C语言源程序在基本组成结构上的特点及其书写风格。



1.3.1 简单C程序举例

【例 1-1】编写一个C语言程序,输出“Hello!”。

程序如下:

```
/* c1_1.c */
#include <stdio.h>      /* 为文件包含,其扩展名为.h,称为头文件 */
void main()
{
    printf("Hello! \n"); /* 通过显示器输出 Hello! */
}
```

说明:

(1)C语言程序中可以随时使用注释,但注释内容不参与编译。

(2)#include 称为文件包含命令,#include <stdio.h>的意义是把尖括号<>或引号""内指定的文件包含到本程序来,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,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。需注意:文件包含命令的末尾不加分号。详细内容将在后面章节介绍。

(3)main 是主函数的函数名,表示这是一个主函数。每个完整的C语言源程序都必须有主函数,且只能有一个主函数(main()函数),程序总是从main()函数开始执行,并终止于main()函数。函数体由一对大括弧“{ }”括起来,其间一般包括程序的说明部分和执行部分。

(4)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。其功能是将输出的内容送到显示器显示。

该程序正确执行后,会在显示器上显示输出:

Hello!

【例 1-2】从键盘输入两个整数(m,n),输出其平均值。

```
/* c1_2.c */
#include<stdio.h>
```

```
void main( )
{
    int m,n,s;                /* 定义 3 个整型变量 */
    printf( "Input two numbers:" );    /* 显示提示信息 */
    scanf( "%d%d", &m, &n );    /* 输入 m,n 值 */
    s=m+n;                    /* 求出 m 与 n 之和,并把它赋予变量 s */
    printf( "average=%d\n", s/2 );    /* 输出 m 与 n 之和的平均成绩 */
}
```

程序分析:

(1) 该程序中使用了 m 、 n 和 s 三个变量,所以在变量使用之前先定义。

(2) `scanf()` 函数是一个由系统定义的标准函数,可在程序中直接调用。它的功能是输入变量 m 和 n 的值。 $\&m$ 和 $\&n$ 中“ $\&$ ”的含义是“取变量地址”,表示将把从键盘输入的两个值分别存放到地址标识为 m 、 n 的存储单元中。

(3) “ $\%d$ ”是输入/输出数据的“格式说明”,用来指定输入/输出时的数据类型和格式, $\%d$ 表示“十进制整数类型”,在执行输出时,屏幕上显示一个十进制整数。

(4) “ $s=m+n$;”为赋值表达式,表示将 $m+n$ 之和赋值给 s 变量所标识的存储单元。

该程序正确执行后,会在显示器上显示输出:

```
Input two numbers:9 5
average = 7
```

【例 1-3】 输入两个整数,进行比较后将较小数输出。

```
/* c1_3.c */
#include<stdio.h>
void main( )
{
    int x,y,z;                /* 定义三个整型变量 */
    int min( int a,int b );    /* 函数类型说明 */
    printf( "Input two numbers:" );    /* 显示提示信息 */
    scanf( "%d%d", &x, &y );    /* 输入 x,y 值 */
    z=min( x,y );              /* 调用 min() 函数 */
    printf( "min=%d\n", z );    /* 将较小数输出 */
}

int min( int a,int b )        /* 定义 min() 函数 */
{
    int c;                    /* 定义一个整型变量 */
    c=a<b? a:b;                /* 求出变量 c 的值 */
    return c;                  /* 将 c 的值返回到主调函数 */
}
```



程序分析:

(1) 本程序包括两个函数:主函数 `main()` 和自定义函数 `min()`。`min()` 函数的作用是将 `a` 和 `b` 中较小者的值赋予变量 `c`, `return` 语句将 `c` 的值返回主调函数。

(2) 在调用 `min()` 函数时,将实际参数 `x` 和 `y` 的值分别对应传给 `min()` 函数中的形式参数 `a` 和 `b`。

(3) “`a<b? a:b;`”是一个条件表达式。当 `a<b` 成立时,“`a<b? a:b;`”表达式的值为 `a` 的值;反之则为 `b` 的值。详细内容将在第二章介绍。

该程序正确执行后,会在显示器上显示输出:

```
Input two numbers:6 3
```

```
min=3
```

本例中涉及函数调用、实际参数和形式参数等概念,如果读者对此不大理解,可先不予以深究,第六章中将会有详尽介绍。



1.3.2 C语言程序的结构特点

通过上面3个C语言源程序,可以看出其基本结构具有以下几个特点。

(1) C语言源程序的基本组成单位是函数。所有的C语言程序都由一个或多个函数构成,其中 `main()` 函数必须有且只能有一个。

(2) `main()` 函数可以出现在C语言源程序的任何位置,程序执行时总是从 `main()` 函数开始,又在 `main()` 函数结束。主函数可以调用标准库函数(如 `printf()`、`scanf()` 等)和用户自定义函数,但标准库函数和用户自定义函数却不能调用主函数。

(3) 源程序中的预处理命令通常放在源文件或源程序的最前面。

(4) 分号“`;`”是C语句的必要组成部分。每个语句或每个变量说明都必须以分号结尾。但预处理命令、函数头和花括号“`{}`”“`{}`”后面不能加分号。

(5) 标识符、关键字之间必须至少加一个空格以示分隔。

(6) 可以在程序的任何位置用“`/* 注释内容 */`”或“`//注释内容`”的形式对程序或语句进行注释,以增加程序的可读性。



1.3.3 书写程序时应遵循的规则

C语言程序的书写格式非常自由,但从书写清晰,便于阅读、理解、维护的角度出发,建议在书写C语言程序时遵循以下几个规则。

(1) 一个说明或一条语句占一行。

(2) 用`{}`括起来的部分,通常表示了程序的某一层结构(如函数体、循环体、复合语句等)。`{}`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比高一层次的语句或说明向后缩进若干格后书写,同一层次的语句或说明左对齐,以增强程序编写的层次感,增加程序的可读性。

(4) 函数块与函数块之间加一空行分隔,以便清楚地分出程序中有几个函数。

在编程时应力求遵循上述规则,以养成良好的编程习惯。