



新世纪高等教育
计算机类课程规划教材

C语言实用教程

主 编 胡元义 吕林涛

主 审 崔俊凯



大连理工大学出版社



亲
让

教材

C语言实用教程

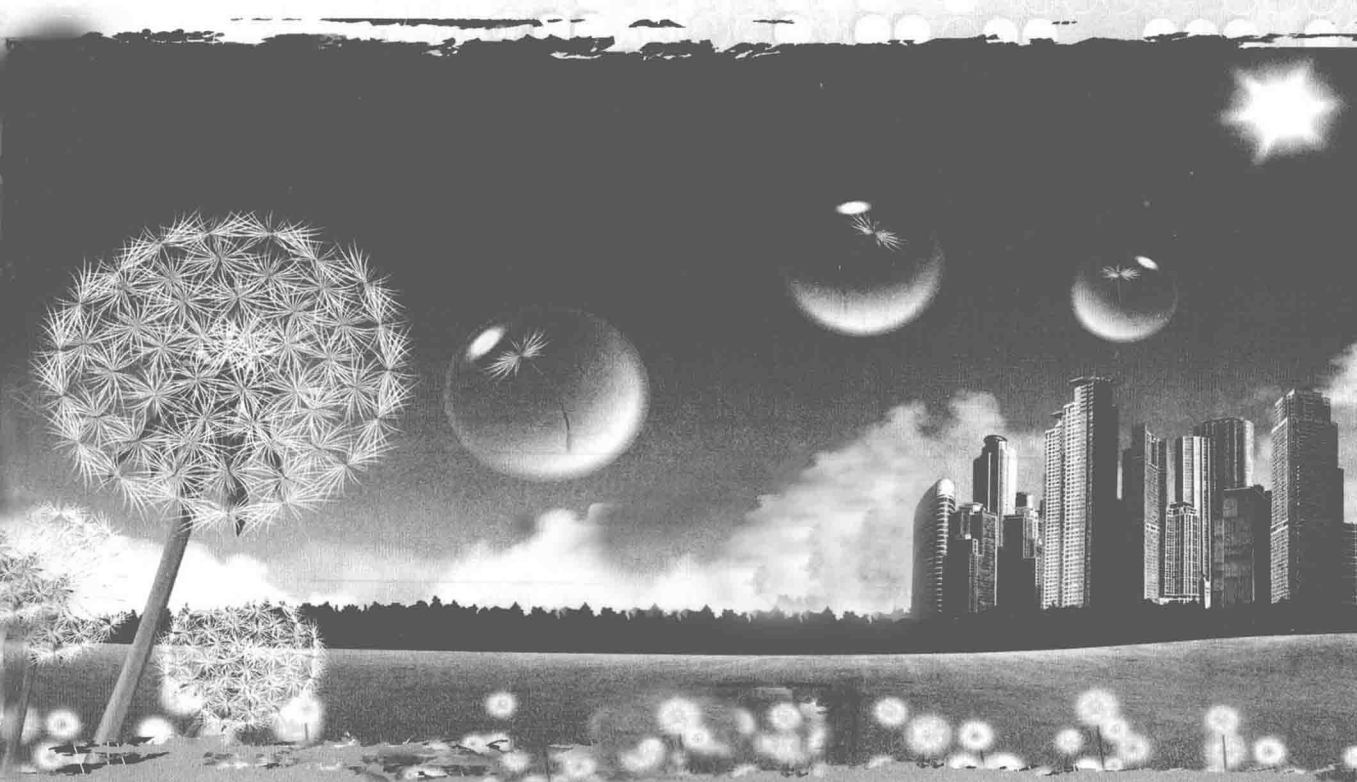
C YUYAN SHIYONG JIAOCHENG

主 编 胡元义 吕林涛

副主编 金海燕 孙钦东 谈姝辰 赵明华

杨 晓 张 鹏 刘岩松 高俊杰

主 审 崔俊凯



大连理工大学出版社

图书在版编目(CIP)数据

C 语言实用教程 / 胡元义, 吕林涛主编. — 大连 :
大连理工大学出版社, 2014. 2

新世纪高等教育计算机类课程规划教材

ISBN 978-7-5611-8595-7

I. ①C… II. ①胡… ②吕… III. ①C 语言—程序设计—高等学校—教材 IV. ①TP312

中国版本图书馆 CIP 数据核字(2014)第 021031 号

大连理工大学出版社出版

地址:大连市软件园路 80 号 邮政编码:116023

发行:0411-84708842 邮购:0411-84703636 传真:0411-84701466

E-mail: dutp@dutp. cn URL: <http://www.dutp. cn>

大连美跃彩色印刷有限公司印刷 大连理工大学出版社发行

幅面尺寸:185mm×260mm 印张:17.5 字数:404 千字
印数:1~2000

2014 年 2 月第 1 版

2014 年 2 月第 1 次印刷

责任编辑:王晓历

责任校对:栗蜚悦

封面设计:张莹

ISBN 978-7-5611-8595-7

定 价:37.00 元

前

言

本书作为应用型本科的程序设计课程的教材,在结构上注重知识的系统性和实用性,在讲授中循序渐进、深入浅出,将理论与实践有机结合,融知识传授与能力培养于一体。

编者在总结多年教学与实践的基础上,精选了大量内容生动、设计独到的例题作为典型概念示例和程序精讲,并且兼顾 C 语言等级考试,许多例题就是选自历年二级 C 语言等级考试试题。全书给出了 400 多道例题和习题,且所有程序例题与习题都在 VC++6.0 环境下上机通过。对重点章节如函数和指针内容,采用了独创的动态图分析方法来分析程序执行中函数或指针变化的情况,使函数和指针内容中难以掌握的难题迎刃而解。此外,对采用指针来指向数组元素的相关内容,采用了新颖的表述方法来解决同一个数组元素有多种表示法的问题。对于文件的讲解,编者也辅以图例来进行说明,以便读者能够深入了解文件内部的读写过程。

本书由胡元义、吕林涛任主编,金海燕、孙钦东、谈姝辰、赵明华、杨晓、张鹏、刘岩松、高俊杰任副主编,崔俊凯审阅了书稿并提出了修改意见。

本书不仅可以作为应用型本科的程序设计语言教材,还可以作为全国计算机等级考试的辅导教材或参考书。对于从事计算机专业的工作者,本书也是难得的一本资料书。

本书与配套的教材《C 语言程序设计题解及上机指导》配合使用将会收到更好的学习效果。书中带“*”的内容



为选讲内容,可根据讲授的课时数进行取舍。

由于编者的水平有限,加之时间仓促,书中难免会存在缺点与不足,希望广大读者与教师提出宝贵意见,并将意见和建议及时反馈给我们,以便下次修订时改进。

编 者

2014 年 2 月

所有意见和建议请发往:dutpbk@163.com

欢迎访问教材服务网站:<http://www.dutpbook.com>

联系电话:0411-84708462 84708445



第 1 章 C 语言与程序设计简介	1
1.1 计算机和程序设计基本概念	1
1.1.1 计算机系统组成	1
1.1.2 程序与程序设计语言	2
1.2 C 语言的发展历程	4
1.3 C 语言程序的基本组成	5
1.4 Visual C++ 上机操作	8
1.4.1 Visual C++ 的安装和启动	8
1.4.2 Visual C++ 环境的使用	9
1.5 C 语言的主要特点	15
 第 2 章 C 语言程序设计基础	18
2.1 C 语言基本符号与数据类型	18
2.1.1 C 语言基本符号	18
2.1.2 C 语言的数据类型	20
2.2 常量	21
2.2.1 整型常量与实型常量	22
2.2.2 字符常量与字符串常量	23
2.2.3 符号常量	25
2.3 变量	25
2.3.1 变量的概念、定义与初始化	25
2.3.2 整型变量、实型变量与字符型变量	27
2.4 运算符与表达式	32
2.4.1 C 语言运算符简介	32
2.4.2 算术运算符与算术表达式	34
2.4.3 关系运算符与关系表达式	36
2.4.4 逻辑运算符与逻辑表达式	38
2.4.5 赋值运算符与复合赋值运算符	40
2.4.6 表达式中数据类型自动和强制转换	41
2.4.7 逗号运算符与逗号表达式	43

2.5 数据的输入/输出	44
2.5.1 字符输出/输入函数	44
2.5.2 格式输出函数	45
2.5.3 格式输入函数	48
第3章 三种基本结构的程序设计	57
3.1 程序的基本结构及 C 程序语句分类	57
3.1.1 程序的基本结构	57
3.1.2 C 程序语句分类	58
3.2 顺序结构程序设计	60
3.2.1 赋值语句	60
3.2.2 顺序结构程序	61
3.3 选择结构程序设计	62
3.3.1 if 语句	63
3.3.2 if 语句的嵌套	67
3.3.3 条件运算符与条件表达式	69
3.3.4 switch 语句	70
3.4 循环结构程序设计	75
3.4.1 while 语句	75
3.4.2 do...while 语句	78
3.4.3 for 语句	81
3.4.4 break 语句、continue 语句和 goto 语句	85
3.4.5 循环嵌套	89
第4章 数 组	101
4.1 一维数组	101
4.1.1 一维数组的定义	101
4.1.2 一维数组的引用和初始化	102
4.2 二维数组	108
4.2.1 二维数组的定义	108
4.2.2 二维数组的引用和初始化	109
4.3 字符数组和字符串	112
4.3.1 字符数组的定义、引用和初始化	112
4.3.2 字符串	115
4.3.3 常用字符串处理函数	117

第 5 章 函 数	127
5.1 函数的概念及分类	127
5.1.1 函数的概念	127
5.1.2 函数的分类	127
5.2 函数的定义和调用	128
5.2.1 函数的定义	128
5.2.2 函数的调用和返回值	130
5.2.3 函数的声明	132
5.2.4 函数执行的分析方法	133
5.3 变量的作用域	135
5.3.1 局部变量与全局变量	135
5.3.2 全局变量引起的函数副作用	137
5.4 函数的嵌套调用与递归调用	137
5.4.1 函数的嵌套调用	137
5.4.2 函数的递归调用	140
* 5.5 变量的存储类别与生命期	145
 第 6 章 指 针	158
6.1 指针和指针变量	158
6.1.1 地址和指针的概念	158
6.1.2 指针变量的定义和初始化	159
6.1.3 指针变量的引用和运算	160
6.2 指针与数组	164
6.2.1 指针与一维数组	164
6.2.2 指针变量与字符串	168
6.2.3 指针变量与二维数组	170
6.2.4 指针数组	174
6.3 多级指针变量	177
6.4 指针变量与函数	179
6.4.1 指针变量作为函数参数	179
6.4.2 用数组名作函数参数	182
6.4.3 返回指针值的函数	185
 第 7 章 结构体	197
7.1 结构体类型的定义与结构体变量	197
7.1.1 结构体类型的定义	197
7.1.2 结构体变量	200

7.1.3 用 typedef 定义类型标识符	206
7.2 结构体数组及指向结构体的指针变量	210
7.2.1 结构体数组	210
7.2.2 指向结构体的指针变量	213
7.3 链表	216
7.3.1 链表的概念	216
7.3.2 动态存储分配	218
7.3.3 动态链表的建立与查找	219
7.4 共用体	223
7.4.1 共用体的概念与定义	223
7.4.2 共用体变量的引用和赋值	224
7.5 枚举类型	229
 第 8 章 文 件	 244
8.1 文件的概念	244
8.1.1 文件的分类	244
8.1.2 文件指针变量及文件操作过程	245
8.2 文件的打开与关闭	246
8.2.1 文件的打开	247
8.2.2 文件的关闭	248
8.3 文件的读写	249
8.3.1 字符读/写函数	250
8.3.2 字符串读/写函数	252
8.3.3 格式化读/写函数	254
8.3.4 数据块读/写函数	256
8.4 文件的定位与随机读/写	259
 附 录	 269
附录 1 ASCII 表	269
附录 2 C 运算符和优先级	270
附录 3 常用 C 库函数	271

1.1 计算机和程序设计基本概念

1.1.1 计算机系统组成

计算机是一种能够自动、高速处理数据的工具。一个完整的计算机系统包括硬件和软件两大部分,其组成如图 1-1 所示。



图 1-1 计算机系统组成

1. 硬件

硬件是指计算机的机器部分,即我们所见到的物理设备和器件的总称。计算机硬件结构如图 1-2 所示。

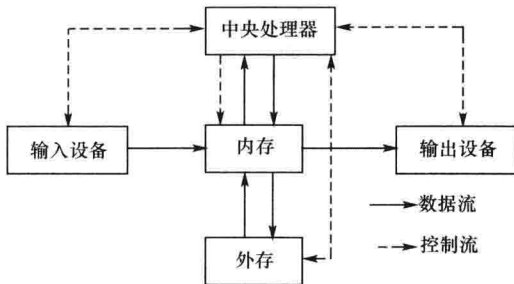


图 1-2 计算机硬件结构图

中央处理器(CPU)是计算机的核心,由控制器和运算器两部分组成。控制器是计算机的神经中枢,统一指挥和控制计算机各部分的工作;运算器对数据进行运算和处理。

计算机存储器分为内存储器和外存储器两种。内存储器简称内存,也叫主存,是计算机用于直接存取程序和数据的地方。内存可直接与 CPU 交换信息。内存存取信息的速度快,但容量有限。外存储器简称外存,也叫辅存,常用的外存有磁盘和 U 盘等。由于内存容量的限制,常用外存来存放大量暂时不用的信息,这些信息一般以文件的形式存放在外存上。CPU 不能直接处理外存中的信息,必须先将这些信息由外存调入内存后再

进行处理,因此程序只有装入内存后才可运行。外存的特点是存储容量大,信息可以长期保存,但存取信息的速度较慢。

输入、输出设备是计算机与外界传递信息的通道。输入设备用于把数据、图像、命令和程序等信息输入给计算机,向计算机中直接输入信息的最常用设备是键盘。输出设备是将计算机执行的结果输出反馈给使用者,主要的输出设备有显示器和打印机。外存中的磁盘和 U 盘既可作为输入设备,又可作为输出设备使用。

2. 软件

软件通常指计算机系统上的程序和数据,并按功能分为系统软件和应用软件两类。系统软件是指为进行计算机系统的管理和使用而必须配置的软件,如操作系统、汇编程序和编译程序等。应用软件是指针对某类专门应用需要而配置的软件,如计算机辅助教学 CAI、财务管理软件以及火车和飞机订票系统等。由于软件具有易于修改和复制的优点,因而便于推广应用。

仅有硬件的计算机系统(裸机)是难以进行工作的。为了对计算机所有软、硬件资源进行有效的控制和管理,在裸机基础上形成了第一层软件,即操作系统。

操作系统是最基本的系统软件,是对硬件机器的首次扩充。其他软件都是建立在操作系统之上,通过操作系统对硬件功能进一步扩充,并在操作系统的统一管理和支持下运行(如图 1-3 所示)。因此,操作系统在整个计算机系统中具有特殊的地位,不仅是硬件与其他软件的接口,而且是整个计算机系统的控制和管理中心,为人们提供了与计算机进行交互的良好界面。

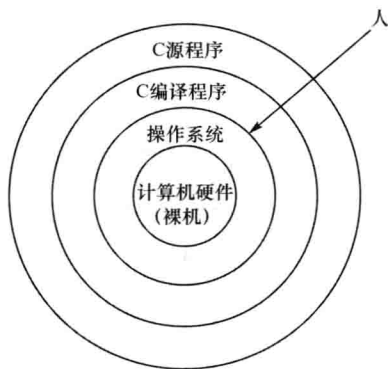


图 1-3 计算机硬件功能的扩展和人机交互的界面

综上所述,硬件是计算机的物质基础,软件是建立在硬件基础之上,对硬件功能的扩充与完善。两者缺一不可,没有软件,计算机的硬件难以工作;没有硬件,软件的功能无法实现。

1.1.2 程序与程序设计语言

要使计算机能够完成人们指定的工作,就必须把工作的具体实现步骤编写成计算机能够识别并执行的一条条指令。计算机执行这个指令序列后,就能完成指定的任务,这样的指令序列就是程序。因此,程序就是由人编写的指挥和控制计算机完成特定功能的

指令序列。书写程序所使用的语言称为程序设计语言,它是人与计算机进行信息通信的工具,而设计、编写和调试程序的过程则称为程序设计。

计算机发展到今天,程序设计语言经历了机器语言、汇编语言和高级语言三个阶段。

我们知道,计算机所能执行的每一种操作称为一条指令,计算机能够执行的全部指令集合就是该计算机的指令系统。由于计算机硬件的器件特性,决定了计算机本身只能直接接受由0和1编码的二进制指令和数据,这种二进制形式的指令集合称为计算机的机器语言,它是计算机唯一能够直接识别并接受的语言。

用机器语言编写程序很不方便且容易出错,编写出来的程序也难以调试、阅读和交流。为此,出现了用助记符来代替机器语言二进制编码的另外一种语言,即汇编语言。汇编语言是建立在机器语言之上的,因为它是机器语言的符号化形式,所以比机器语言直观。但是计算机并不能直接识别这种符号化的语言,因此用汇编语言编写的程序必须翻译成机器语言之后计算机才能够执行,这种“翻译”是通过专门的软件——汇编程序来实现的。

尽管汇编语言与机器语言相比在阅读和理解上有了长足的进步,但其依赖具体机器的特性是无法改变的。能够编写好汇编程序除了掌握汇编语言之外,还必须了解计算机的内部结构和硬件特性,再加上不同计算机上的汇编语言又各不相同,这无疑给程序设计增加了难度。

随着计算机应用需求的不断增长,出现了更加接近人类自然语言的、功能更强、抽象级别更高的面向各种应用的高级语言。高级语言由于接近人们的自然语言,使用方便,编写的程序也符合人们的习惯,能够较自然地描述各种问题,从而极大地提高了编程的效率,编写的程序也便于查错、阅读和修改。更为重要的是,高级语言已经从具体机器中抽象出来,摆脱了依赖具体机器的问题,使用高级语言编写的程序几乎在不改动的情況下就能在任何计算机上运行,并且编程人员在编写程序时也无须了解计算机内部的硬件结构。这些都是机器语言和汇编语言难以做到的。

与汇编语言一样,计算机也不能够直接识别用高级语言编写的程序,即必须经过编译程序的分析和加工,将其翻译成机器语言程序后再执行。编译程序是一个把高级语言程序翻译成等价的机器语言程序的程序。在编译方式下,高级语言程序的执行分为两个阶段:编译阶段和运行阶段。编译阶段把高级语言程序翻译成机器语言程序,运行阶段才真正执行这个机器语言程序(如图1-4所示)。

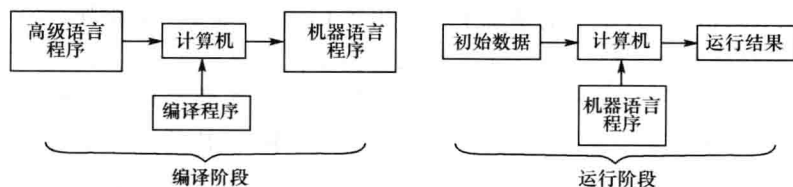


图 1-4 高级语言程序执行过程示意图

例如,给内存中16进制地址为1000的单元中的数据加上十进制数10,则用机器语言、汇编语言和C高级语言表示分别如下:

(1)用 8086/8088 机器语言表示

```
10100001 11010000 00000111    //将 16 进制 1000 地址中的数据存入 AX 寄存器
10000011 00001010              //将 AX 寄存器中的数据加 10
10100011 11010000 00000111    //将 AX 寄存器中的数据存入 16 进制 1000 地址单元中
```

(2)用 8086/8088 汇编语言表示

```
MOV AX, [1000H]                //将 1000 地址中的数据存入 AX 寄存器
ADD AX, 10                      //将 AX 寄存器中的数据加 10
MOV [1000H], AX                //将 AX 寄存器中的数据存入 16 进制 1000 地址单元中
```

(3)用 C 高级语言表示

```
X=X+10;                        //X 为 1000 地址的变量名
```

自从 20 世纪 50 年代中期第一个高级语言——FORTRAN 语言问世以来,全球相继已经出现了几千种高级语言,但广泛使用的高级语言也不过数十种。例如,适用于科学计算的 FORTRAN 语言,适用于商业事务处理的 COBOL 语言,第一个体现结构化程序设计思想的 PASCAL 语言,用于人工智能程序设计的 PROLOG 语言,功能丰富的 C 语言,以及面向对象程序设计的 C++、Java 和 Delphi 语言等。

1.2 C 语言的发展历程

由于操作系统等系统程序依赖于计算机硬件,以前这类系统程序主要是用汇编语言编写的。但是汇编语言程序的可读性和可移植性都很差,严重影响了系统程序的编写效率。在这种情况下,人们希望有一种语言既有高级语言可读性高、便于移植的优点,同时又具有汇编语言能够直接访问计算机硬件的特点。因此,C 语言应运而生。

C 语言的起源可以追溯到 ALGOL 60 语言。1963 年,英国剑桥大学在 ALGOL 60 的基础上推出了 CPL(Combined Programming Language)语言,但该语言由于规模较大而难以实现。1967 年,英国剑桥大学的 Martin Richards 对 CPL 语言做了简化和改进,推出了 BCPL(Basic Combined Programming Language)语言。1970 年,美国贝尔实验室的 Ken Thompson 以 BCPL 语言为基础,又做了进一步的简化,设计出简单且接近硬件的 B 语言(取 BCPL 的第一个字母),并且用 B 语言写出了第一个 UNIX 操作系统,并在 DEC PDP-7 型计算机上实现。1971 年,在 DEC PDP-11 计算机上实现了 B 语言。1972 年,由美国的 D. M. Ritchie 在 B 语言的基础上设计出了 C 语言(取 BCPL 的第二个字母),并首次在 UNIX 操作系统的 DEC PDP-11 计算机上使用。

后来 C 语言又进行了多次改进,但主要还是在贝尔实验室内部使用。1977 年,D. M. Ritchie 发表了不依赖于具体机器的 C 语言编译文本《可移植 C 语言编译程序》,使 C 语言移植到其他机器时所做的工作大为简化,这也推动了 UNIX 操作系统迅速地在各种机器上实现。随着 UNIX 操作系统的广泛使用,C 语言也迅速得到推广,成为世界上应用最为广泛的程序设计语言之一。

1978 年,B. W. Kernighan 和 D. M. Ritchie 两人合作出版了 C 语言白皮书《The C Programming Language》,给出了 C 语言的详细定义。1983 年,美国国家标准化协会

(ANSI)对C语言的各种版本做了扩充和完善,制定了C语言的标准(称为ANSI C),这就给C语言程序的移植创造了更有利的环境。1990年,ANSI C为国际标准化组织ISO所接受。

微机上使用的C语言编译系统多为Microsoft C、Turbo C、Borland C和Quick C等,它们都是按标准C语言编写的,相互之间略有差异,每一种编译系统又有不同的版本,各版本之间也存在着差异,主要是版本越高的编译系统所提供的函数越多、编译能力越强、使用越方便。

1.3 C语言程序的基本组成

下面通过几个简单的C语言程序大致了解一下C语言程序的基本组成,以便有一个初步的概念。

【例 1-1】 在显示器输出“Hello,China!”。

```
#include <stdio.h>           //使用C语言提供的标准输入输出函数
void main()                  //主函数 main
{
    printf("Hello,China! \n"); //用输出函数 printf 实现输出显示字符串
}
```

运行结果:

Hello,China!

程序说明如下:

(1)一个C程序有一个名为main的主函数,main是主函数名。一个C语言有且仅有一个主函数main,程序执行时就是从主函数main开始,具体来讲就是从main下面的“{”开始到“}”结束。花括号“{ }”中间的内容称为函数体,该函数体就是实现某种功能的一段程序。例1-1的函数体只有一个printf函数调用语句,该语句实现将双引号“”括起的字符串照原样输出到显示器上,而字符串中的“\n”字符表示换行的意思,即在输出以下字符串:“Hello,China!”后光标换到下一行的开始处。

(2)在main前加上“void”,表示主函数main没有返回值。main后面的圆括号“()”表明主函数main没有参数。

(3)printf函数调用语句后面有一个分号“;”,表示该语句的结束。C语言规定:语句以分号“;”表示结束。

(4)程序中的“//”表示注释,即“//”之后的当前行文字是注释内容,表示程序中的注释是给阅读程序的人看的,是为了提高程序的可读性而加入的说明性信息。注释内容的有无都不影响程序的功能和运行结果。

(5)程序第一行的“#include <stdio.h>”是编译预处理命令行,该命令通知编译系统将包含输入和输出标准库函数的stdio.h文件作为当前源程序的一部分。如输出函数printf以及输入函数scanf都要使用stdio.h才能实现输出和输入功能。

【例 1-2】 求两个数 a 与 b 之和。

```
#include <stdio.h>

void main()
{
    int x,y,sum;                //定义 x、y、sum 3 个整型变量
    printf("Input x and y:\n"); //在显示器上显示提示输入的信息
    scanf("%d%d",&x,&y);        //由键盘输入 x 和 y 的值
    sum=x+y;                    //完成 x+y 的求和并将结果赋值给 sum
    printf("x+y=%d\n",sum);     //输出求和结果
}
```

运行结果:

Input x and y:

12 15 ✓

x+y=27

程序说明如下:

为了表示变化的数据,程序中引入了变量,即通过变量来保存数据的值。在例 1-2 中定义了 3 个变量,分别命名为 x、y 和 sum。第一个函数调用语句 printf 输出“Input x and y:”,用来输出提示输入 x 和 y 值的信息,这时就可以由键盘输入变量 x 和变量 y 两个数值了。scanf 是 C 语言的输入函数,语句中的“%d”是整型数据输入格式,用来指定由键盘输入数据的格式和类型。“%d%d”表示读入两个数,输入时要用空格符分隔。scanf 语句中的“&x,&y”则表示将输入的两个数分别送给变量 x 和变量 y,这里的“&”表示地址,即数据是送到 x 和 y 对应的内存地址中。语句“sum=x+y;”实现 x+y 的求和并将结果赋值给变量 sum。第二个函数调用语句 printf 先输出“x+y=”,然后按格式“%d”将 sum 的值输出。

【例 1-3】 从键盘上输入两个整数,在屏幕上输出它们的最大值。

```
#include <stdio.h>

int max(int x,int y);          //函数声明
void main()                   //主函数
{
    int a,b,c;                //定义变量 a、b、c 为整型
    printf("Input a,b=");     //输出提示字符串“Input a,b=”
    scanf("%d,%d",&a,&b);      //由键盘输入 a、b 值
    c=max(a,b);               //调用函数 max(),并将 max() 的返回值送给变量 c
    printf("Max is:%d\n",c);   //输出结果
}

int max(int x,int y) //定义函数 max(),形参 x、y 为整型。开头的 int 表示返回值为整型
{
    int z;              //定义变量 z 为整型
    if(x>y)             //条件判断语句,判断 x 是否大于 y
        z=x;           //x>y 为真时将 x 值赋给 z
}
```

```
else
    z=y;                //x>y 为假时将 y 值赋给 z
return (z);            //将 z 值返回给调用函数 main()
}
```

运行结果:

Input a,b=8,12 ✓

Max is:12

程序说明如下:

(1)程序中定义了两个函数:主函数 main 和被调函数 max。注意函数 scanf 和 printf 是 C 语言提供的标准函数,无须用户定义,只要通过预处理命令“#include”将“stdio.h”包含在程序中即可直接使用。而函数 max 是用户自行定义的函数,即在程序中必须写出函数 max 的定义。

(2)一个自定义函数由两部分组成。

①函数首部:包括函数名、函数类型、参数类型和参数名。例如:

int	max	(	int	x	,	int	y	)
↓	↓		↓	↓		↓	↓	
函数类型	函数名		参数类型	参数名		参数类型	参数名	

②函数体:即函数首部下面的花括号“{ }”内的部分,如果一个函数内有多个花括号“{ }”,则最外层的一对“{ }”为函数的范围。函数体一般包括说明部分和执行部分,它们都是 C 语言的语句。说明部分用来定义函数内部所用到的变量及其数据类型。执行部分则实现函数要完成的功能,例如,函数 max 就是将 x、y 两个数中较大者赋给 z,然后返回 z 值给主函数 main。

③由于本例中自定义函数放置于主函数 main 之后,因此要先在主函数 main 之前予以声明,即将自定义函数 max 的首部作为声明语句在主函数 main 之前重写一次(并添加“;”)进行声明。

(3)主函数 main 中的函数 printf 和函数 scanf 的功能同例 1-2,语句“c=max(a,b);”的作用是:a 和 b 作为实参调用函数 max,a 和 b 的值分别传给函数 max 的形参 x 和 y,且函数 max 最终通过 return 语句将 z 值返回给主函数 main 中的 c 变量。

通过以上几个例子,可以概括出 C 语言程序的结构特点:

(1)C 语言程序主要由函数构成。C 语言程序中有主函数 main、系统提供的库函数(如 printf 和 scanf)以及由编程者自定义的函数(如 max 等)三种类型函数。

(2)一个函数由说明部分和执行部分组成。说明部分在前,执行部分在后,这两部分顺序不能颠倒。

(3)一个程序总是从主函数 main 开始执行,无论主函数 main 写在程序中的什么位置。

(4)C 语言书写格式自由。一个语句可以占多行,一行也可写多个语句。

(5)C 语言的语句都是以分号“;”结尾。

(6)为了增加程序的可读性,C 程序中可用“//”对程序进行注释,注释部分可以放置到程序的任何位置。

(7)程序中可以有由“#”开头的预处理命令行(include 仅为其中的一种),预处理命令行通常应放在程序的最前面。

1.4 Visual C++ 上机操作

Visual C++ 是一个功能强大的可视化软件开发工具。自 1993 年 Microsoft 公司推出 Visual C++ 1.0 后,其新版本不断问世,现在常用的是 Visual C++ 6.0 版本。Visual C++ 6.0 不仅是一个 C++ 编译器,而且是一个基于 Windows 操作系统的可视化集成开发环境,它由许多组件组成,包括编辑器、编译器、链接器、生成实用程序、调试器以及为开发 Windows 下的 C/C++ 程序而设计的各种工具。本书以 Visual C++ 6.0 英文版为背景来介绍 Visual C++ 的上机操作。

1.4.1 Visual C++ 的安装和启动

如果你的计算机未安装 Visual C++ 6.0,则应先安装。Visual C++ 是 Visual Studio 的一部分,因此需要找到 Visual Studio 的光盘,执行其中的 setup.exe,然后按屏幕上的提示进行安装即可。安装结束后,在 Windows 的“开始”菜单的“程序”子菜单中就会出现“Microsoft Visual Studio 6.0”子菜单。

在使用 Visual C++ 时,只需从桌面上顺序选择【开始】|【程序】|【Microsoft Visual Studio 6.0】|【Visual C++ 6.0】即可,此时屏幕上出现 Visual C++ 6.0 的主窗口,如图 1-5 所示。

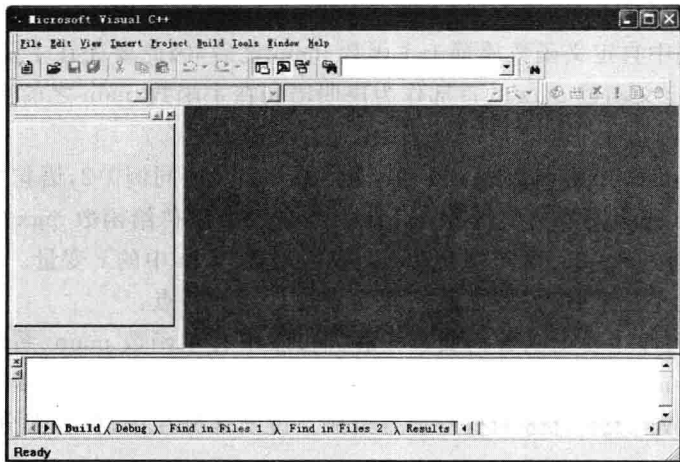


图 1-5 Visual C++ 6.0 主窗口

在 Visual C++ 主窗口的顶部是 Visual C++ 的主菜单栏,其中包含 9 个菜单项:File(文件)、Edit(编辑)、View(查看)、Insert(插入)、Project(项目)、Build(构建)、Tools(工具)、Window(窗口)和 Help(帮助)。

主窗口的左侧是项目工作区窗口,用来显示所设定的工作区信息;右侧是程序编辑窗口,用来输入和编辑源程序。