



YUYAN CHENGXU SHEJI

C语言程序设计

100111101101010011010110010011010110101010010100100110

主编 李洁

10100110101010110010101110101010100100111

天津科学技术出版社

C 语言程序设计

主 编 李 洁
编写者 李 洁 万 红 王成霞
彭旭东 王海燕

天津科学技术出版社

图书在版编目(CIP)数据

C 语言程序设计 / 李洁主编. —天津:

天津科学技术出版社, 2011. 1

ISBN 978-7-5308-5439-6

I. ①C… II. ①李… III. ①C 语言—程序设计

IV. ①TP312

中国版本图书馆 CIP 数据核字(2011)第 004067 号

责任编辑:宋庆伟

装帧设计:薛 芹 晓 君

责任印刷:王 莹

天津科学技术出版社出版

出版人:蔡 颢

天津市西康路 35 号 邮编 300051

电话:(022)23332379(编辑室) 23332393(发行部)

网址:www.tjkjcs.com.cn

新华书店经销

三河市佳星印装有限公司印刷

开本 787×1092 1/16 印张 16.5 字数 370 000

2011 年 1 月第 1 版第 1 次印刷

定价:38.00 元

前 言

C 语言是大学的必修课。而且 C 语言有很长的历史,但它在现在的很多领域仍然有实用的价值。C 语言作为学习计算机语言的起步仍然是一个很好的范例,对学习更“高级”的语言是很有帮助的。

本书是针对大学计算机基础教育而编制的,是面向初学者的教材。重点介绍 C 语言的基本概念和算法,内容及章节是按照由浅入深、循序渐进的方式编排的。特别适合初学者使用。

全书共分为 10 章。第 1 章,C 语言程序设计概述;第 2 章,数据类型、运算符和表达式;第 3 章,C 程序的基本控制结构;第 4 章,数组;第 5 章,指针;第 6 章,函数;第 7 章,自定义数据类型;第 8 章,文件;第 9 章,编译预处理;第 10 章,管理内存。

为了便于教师教学和学生的学习,本教材还配有与之配套的上机指导和习题集《C 语言程序设计实训指导与模拟试题》。

本教材由李洁主编,其他参加编写的人员有万红、王成霞、彭旭东、王海燕。由于本书编写时间紧,且本书中的很多实例会因 C 语言环境和版本的不同难免出现偏差,因此产生的错误和不足之处,还望广大教师、读者多提宝贵意见,为今后修订提供帮助。

编 者

2010 年 12 月

目 录

第 1 章 C 语言程序设计概述	1
1.1 程序设计概述	1
1.1.1 程序与程序设计语言	1
1.1.2 程序设计的一般过程	2
1.1.3 程序设计方法	3
1.2 C 语言简介	3
1.2.1 C 语言出现的历史背景	3
1.2.2 C 语言的特点	4
1.2.3 C 语言字符集	5
1.2.4 C 语言标志符与关键字	5
1.2.5 C 语言程序基本结构	7
1.3 C 语言编程环境	10
1.3.1 C 语言程序的编译和执行	10
1.3.2 Visual C++ 6.0 编程环境	11
1.3.3 Turbo C 2.0 编程环境	14
第 2 章 数据类型、运算符和表达式	17
2.1 基本数据类型	17
2.2 常量与变量	18
2.2.1 常量	19
2.2.2 变量	22
2.3 运算符与表达式	26
2.3.1 运算符与表达式的概念	26
2.3.2 算术运算符和算术表达式	27
2.3.3 自增、自减运算符	29
2.3.4 赋值运算符和赋值表达式	29
2.3.5 逗号运算符和逗号表达式	33
2.3.6 求字节数运算符	33
2.3.7 数值型数据间的混合运算	34
第 3 章 C 程序的基本控制结构	36
3.1 顺序结构	38
3.1.1 C 程序的语句	38

3.1.2	数据输出函数 printf()和 putchar()	39
3.1.3	数据输入函数 scanf()和 getchar()	46
3.1.4	顺序结构程序举例	49
3.1.5	常见错误	51
3.2	选择结构	52
3.2.1	关系运算	52
3.2.2	逻辑运算	54
3.2.3	if 语句和条件运算符	55
3.2.4	switch 语句	61
3.2.5	选择结构程序举例	65
3.2.6	常见错误	66
3.3	循环结构	67
3.3.1	while 语句	67
3.3.2	do-while 语句	68
3.3.3	for 语句	71
3.3.4	三种循环的比较	74
3.3.5	循环的嵌套	75
3.3.6	break 和 continue 语句	77
3.3.7	循环结构举例	80
3.3.8	常见错误	82
第 4 章	数组	85
4.1	一维数组	85
4.1.1	一维数组的定义	85
4.1.2	一维数组的引用	86
4.1.3	一维数组的初始化	87
4.1.4	一维数组的应用	89
4.2	二维数组与多维数组	94
4.2.1	二维数组的定义	94
4.2.2	二维数组的引用	95
4.2.3	二维数组的初始化	96
4.2.4	二维数组应用举例	98
4.3	字符数组	106
4.3.1	字符数组的定义	106
4.3.2	字符数组与字符串	107
4.3.3	字符串处理函数	109
4.3.4	字符数组应用举例	112
第 5 章	指针	118
5.1	指针基础	118
5.1.1	地址与指针	118

5.1.2 指针变量及其定义	118
5.1.3 指针变量的引用	120
5.1.4 指针的移动与比较	124
5.1.5 三个特殊概念	125
5.2 指针与数组	127
5.2.1 数组中的地址	127
5.2.2 访问一维数组	127
5.2.3 指针变量与二维数组	131
5.2.4 行指针	132
5.2.5 指针数组	134
5.3 二级指针	138
5.4 常见错误	139
5.5 程序举例	141
第 6 章 函数	144
6.1 函数的概述	144
6.1.1 自定义函数	144
6.1.2 库函数	145
6.2 函数的定义	148
6.2.1 定义格式	148
6.2.2 返回与返回值	150
6.3 函数的调用	152
6.3.1 调用格式	152
6.3.2 调用过程	153
6.3.3 函数间的数据传递	154
6.3.4 递归调用	156
6.4 函数的说明	160
6.4.1 函数说明的格式	160
6.4.2 函数说明的省略	161
6.5 main 函数的参数	162
6.6 指向函数的指针	163
6.7 变量的作用域与存储类型	165
6.7.1 变量的作用域	165
6.7.2 变量的存储类型	168
6.8 函数的存储类型	174
6.9 程序举例	175
6.10 常见错误	176
第 7 章 自定义数据类型	177
7.1 结构体与结构体变量	177
7.1.1 结构体	177

7.1.2 定义与初始化结构体变量	177
7.1.3 访问结构体变量	179
7.1.4 嵌套定义结构体	181
7.2 结构体数组	182
7.2.1 定义与初始化结构体数组	182
7.2.2 访问结构体数组	183
7.3 结构体指针	184
7.3.1 操作结构体变量	184
7.3.2 操作结构体数组	186
7.4 结构体与函数	187
7.4.1 参数中的结构体变量	188
7.4.2 参数中的结构体指针	188
7.4.3 返回结构体类型的值	189
7.4.4 返回结构体类型的地址	191
7.5 结构体的应用	192
7.6 联合体	193
7.6.1 联合体	193
7.6.2 定义与初始化联合体变量	194
7.6.3 访问联合体变量	195
7.6.4 程序举例	196
7.7 位段	197
7.8 枚举	200
7.9 用 typedef 定义类型	203
第 8 章 文件	205
8.1 文件的基本概念	205
8.2 文件的类别	205
8.3 操作文件	206
8.3.1 打开与关闭文件	206
8.3.2 读写文件	208
8.3.3 定位读写	216
8.4 出错检测	217
8.5 程序举例	218
第 9 章 编译预处理	219
9.1 预处理概述	219
9.2 宏定义	219
9.2.1 无参数的宏	219
9.2.2 带参数的宏	222
9.4 文件包含	227
9.5 条件编译	227

第 10 章 管理内存	230
10.1 内存的动态分配与释放	230
10.2 操纵内存	232
10.3 位运算	238
10.4 常见问题	242
附 录	243
附录 1 ASCII 表	243
附录 2 运算符的优先级和结合性	245
附录 3 库函数表	247
附录 4 编译系统错误提示的解释	249
附录 5 进制和补码运算	251
参考文献	253

第 1 章 C 语言程序设计概述

学习目标

- (1)了解程序设计语言的发展及其特点。
- (2)掌握机器语言、汇编语言和高级语言的差异。
- (3)了解C语言的发展历史及特点和应用。
- (4)掌握C语言程序的基本结构。
- (5)掌握编写C语言程序的基本步骤和调试过程。

1.1 程序设计概述

程序设计(Programming)是给出解决特定问题程序的过程,是软件构造活动中的重要组成部分。程序设计往往以某种程序设计语言为工具,给出这种语言下的程序。程序设计过程应当包括分析、设计、编码、测试、排错等不同阶段。

1.1.1 程序与程序设计语言

1. 程序

程序就是为实现预期目的而进行操作的一系列语句和指令。一般分为系统程序和应用程序两大类。程序就是为使电子计算机执行一个或多个操作,或执行某一任务,按序设计的计算机指令的集合。程序(program)是为实现特定目标或解决特定问题而用计算机语言编写的命令序列的集合,是用汇编语言、高级语言等开发编制出来的可以运行的文件,在计算机中称可执行文件(后缀名一般为.exe)。我们玩的游戏一般都是应用程序(但现在后缀名为.swf的Flash影片类游戏也比较流行)。

通常,一个计算机程序主要描述两部分的内容,其一是描述问题的每个对象及它们之间的关系;其二是描述对这些对象进行处理的规则。其中关于对象及它们之间的关系涉及到数据结构的内容,而处理规则却是求解某个问题的算法。因此,对程序的描述,经常有如下等式:

程序 = 数据结构 + 算法

算法应当具有以下特性。

- (1)有穷性 一个算法包含的操作步骤应该是有限的。
- (2)确定性 算法中每一条指令必须有确切的含义,不能有二义性,对于相同的输入必须得到相同的执行结果。
- (3)可行性 算法中指定的操作,都可以通过基本运算执行有限次后实现。

(4)有0个或多个输入 算法是用来处理数据对象的,在大多数情况下需要输入这些数据对象。

(5)有一个或多个输出 算法的目的是为了求解,求解结果只有通过输出才能得到。

一个设计合理的数据结构往往可以简化算法,而且一个好的程序有可靠性、易读性、可维护性等良好特性。

2. 程序设计语言

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。它是一种被标准化的交流技巧,用来向计算机发出指令。一种计算机语言让程序员能够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。

程序设计语言可以分为机器语言、汇编语言和高级语言三大类。

机器语言、汇编语言这两类程序设计语言依赖于计算机硬件,因机器而异,又称为低级语言,而高级语言与计算机硬件基本无关,是目前使用非常广泛的程序设计语言。

机器语言是一种用二进制数“0”和“1”表示计算机各种操作的程序设计语言。计算机能够直接理解和执行使用机器语言编写的程序,执行速度快,占用内存少;但难于记忆,编程困难,可读性差,容易出错。机器语言是面向机器的语言,可移植性极差。

汇编语言是一种用助记符来表示各种基本操作的程序设计语言。与机器语言相比,汇编语言大大减少了程序编写、阅读、修改、维护等方面的工作量,但汇编语言也是一种面向机器的语言,计算机不能直接执行,通用性和可移植性差,必须经过汇编程序翻译成机器语言程序后才能在计算机上执行,需要程序员对计算机内部结构非常了解。

高级语言是一种用接近自然语言和数学语言的语法、符号描述基本操作的程序设计语言。高级语言编写的程序不依赖于具体的机器,程序员编程时无须关心计算机内部的细节,从而大大提高了编程效率。计算机不能直接执行使用高级语言编写的程序,需用专门的翻译程序将其转换成机器语言程序后才能执行。

C语言是一种高级语言,因而必须经过编译、连接,成为机器语言程序后才能执行。

1.1.2 程序设计的一般过程

所谓程序设计,就是根据计算机要完成的任务,提出相应的需求,在此基础上设计数据结构和算法,然后再编写相应的程序代码并测试该代码运行的正确性,直到能够得到正确的运行结果为止。

概括地说,程序设计就是分析问题、编写程序、调试程序的过程。

程序设计步骤如下。

- ①分析要解决的问题,明确任务。
- ②建立数学模型,并选择合适的解决方案。
- ③确定数据结构和算法:数据结构 + 算法 = 程序。
- ④编写程序。
- ⑤调试程序。
- ⑥整理资料,交付使用。

1.1.3 程序设计方法

1. 结构化程序设计方法

设计程序时只能使用顺序、选择、循环三种基本结构。

结构化程序设计强调程序设计风格和程序结构的规范化,提倡清晰的结构。

结构化程序设计方法的基本思路,是把一个复杂问题的求解过程分阶段进行,每个阶段处理的问题都控制在人们容易理解和处理的范围内。

(1)自顶向下 是将复杂、大的问题划分为小问题,找出问题的关键、重点所在,然后用精确的思维定性、定量地去描述问题。

(2)逐步细化 是将现实世界的问题经抽象转化为逻辑空间或求解空间的问题。复杂问题经抽象化处理变为相对比较简单的问题。经若干步抽象(精化)处理,最后到求解域中只是比较简单的编程问题。

(3)模块化设计 就是程序的编写不是开始就逐条录入计算机语句和指令,而是首先用主程序、子程序等框架把软件的主要结构和流程描述出来,并定义和调试好各个框架之间的输入、输出链接关系。逐步求精的结果是得到一系列以功能块为单位的算法描述。以功能块为单位进行程序设计,实现其求解算法的方法称为模块化。模块化的目的是为了降低程序复杂度,使程序设计、调试和维护等操作简单化。

(4)结构化编码 使用语言中顺序、选择、重复等有限的基本控制结构表示程序的逻辑结构。

2. 程序设计的基本原则

正确性:程序本身必须具备且只能具备程序设计规格说明书中所列举的全部功能。它是判断程序质量的首要标准。

可靠性:程序在多次反复使用过程中不失败的概率。

简明性:简明性的目标是要求程序简明易读。

有效性:程序在计算机上运行需要使用一定数量的计算机资源,如CPU的时间、存储器的存储空间。有效性就是要在一定的软、硬件条件下,反映出程序的综合效率。

可维护性:程序的可维护可分为校正性维护、适应性维护和完善性维护。一个软件的可维护性如何直接关系到程序的可用性,因此应特别予以关注。

可移植性:程序主要与其所完成的任务有关,但也与它的运行环境有着一定的联系。软件的开发应尽可能远离机器的特征,以提高它的可移植程度。

1.2 C语言简介

C语言是目前国际上广泛流行的一种结构化的程序设计语言,它不仅是开发系统软件很好的工具,而且也是开发应用软件很好的程序设计语言。因此,它深受广大程序设计者的欢迎。

1.2.1 C语言出现的历史背景

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认识,到了20世纪80年代,C语言开始进入其他操作系统,特别是Windows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。

在C的基础上,1983年又由贝尔实验室的Bjarne Stroustrup推出了C++。C++进一步扩充和完善了C语言,成为一种面向对象的程序设计语言。C++目前流行的最新版本是Borland C++ 4.5, Symantec C++ 6.1,和Microsoft Visual C++ 2.0。C++提出了一些更为深入的概念,它所支持的这些面向对象的概念容易将问题空间直接地映像到程序空间,为程序员提供了一种与传统结构程序设计不同的思维方式和编程方法。因而也增加了整个语言的复杂性,掌握起来有一定难度。

但是,C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,再进一步学习C++就能以一种熟悉的语法来学习面向对象的语言,从而达到事半功倍的目的。

目前可在微机上运行的C语言版本主要有Microsoft C/C++, Turbo C, Quick C, Visual C/C++等版本。

1.2.2 C语言的特点

C语言是一种结构化语言。它层次清晰,便于按模块化方式组织程序,易于调试和维护。C语言的表现能力和处理能力极强。它不仅具有丰富的运算符和数据类型,便于实现各类复杂的数据结构。它还可以直接访问内存的物理地址,进行位(bit)一级的操作。由于C语言实现了对硬件的编程操作,因此C语言集高级语言和低级语言的功能于一体。既可用于系统软件的开发,也适合于应用软件的开发。此外,C语言还具有效率高,可移植性强等特点。因此,C语言广泛地移植到了各类各型计算机上,从而形成了多种版本的C语言。

归纳起来C语言具有下列特点。

(1)C语言是中级语言 它把高级语言的基本结构和语句与汇编语言的控制和灵活性有机地结合了起来。C语言可以像汇编语言一样对位、字节和地址进行操作,而这三者是计算机最基本的工作单元。

(2)C语言是结构化语言 结构化语言的显著特点是代码及数据的分隔化,即程序的各个部分除了必要的信息交流外彼此独立。这种结构化方式可使程序层次清晰,便于使用、维护以及调试。C语言是以函数形式提供给用户的,这些函数可方便地调用,并具有多种循环、条件语句控制程序流向,从而使程序完全结构化。

(3)C语言功能齐全 C语言具有各种各样的数据类型,并引入了指针概念,可使程序效率更高。另外C语言也具有强大的图形功能,支持多种显示器和驱动器。而且计算功能、逻辑判断功能也比较强大,可以实现决策目的。

(4)可移植性 C语言程序具有较高的移植性,可以把为某种计算机编写的软件运行在另一种机器或操作系统上,如在DOS下写的程序,能够方便地在Windows 2000下运行,这个程序就是一个可移植的程序。C语言不包含依赖硬件的输入输出机制,其输入输出功能是由独立于C

语言的库函数来实现。这样就使C语言程序本身不依赖于硬件系统,也便于在不同的机器系统间移植。

C语言的优点很多,但是也存在一些缺点,如运算优先级太多,运算能力方面不像其他高级语言那样强,语法定义不严格等。但是这些都不能阻止C语言成为一门广受欢迎的计算机编程语言!

1.2.3 C语言字符集

同汉语和英语等自然语言一样,C语言也具有相应的语法结构和构成规则,具有字符、单词、语句、文章(程序)的基本成分和结构,由字符可以构成单词,由单词可以构成语句,由语句可以构成段落(函数模块),由函数模块可以构成程序。

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。

在C语言中使用的所有字符(符号),可归结为以下几种类别。

- (1)大、小写英文字母(52个) A~Z,a~z,大小写不等效,如A和a是不同的字符。
- (2)十进制数字符号(10个) 0~9。
- (3)标点符号 逗号(,)、分号(;)、单引号(')、双引号(")、冒号(:)、空格()、左花括号({)、右花括号(})、回车(CR)等,还有半角西文符号。
- (4)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不计。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。

表 1-1 C 语言编程中可以使用的特殊符号

+	-	*	/	%	=
{	}	(	)	[	]
_(下画线)	'(单引号)	.	:	?	~
<	>	&	;	"	
!	#	空格			

1.2.4 C语言标志符与关键字

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)、下画线组成的字符串,并且其第一个字符必须是字母或下画线。

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)

标志符的命名规则如下。

- (1)所有标志符必须由一个字母(a~z或A~Z)或下画线(_)开头。
- (2)标志符的其他部分可以用字母、下画线或数字(0~9)组成。
- (3)大小写字母表示不同意义,即代表不同的标志符,如前面的cout和Cout。
- (4)不能与C语言编译系统已经预定义的、具有特殊用途的保留标志符(即关键字)同名。

在定义标志符时,虽然语法上允许用下画线开头,但是,我们最好避免定义用下画线开头的标志符,因为编译器常常定义一些下画线开头的标志符。

C语言没有限制一个标志符中字符的个数,但是,大多数的编译器都会有限制。不过,我们在定义标志符时,通常并不担心标志符中字符数会不会超过编译器的限制,因为编译器限制的数字很大(例如255)。

不同的C编译程序对标志符所用字符个数有不同的规定,ANSI C可识别标志符的前31个字符,但有的C编译程序只识别前8个字符。在这种编译程序中,finaldatal与finaldata2被认为是同一个标志符,应特别注意。此外,编译程序对标志符的大小写字母是要加以区分的,如name和NAme、NAME是三个不同的标志符。

一个写得好的程序,标志符应该尽量有意义。比如表示年可以用year,表示长度可以用length,表示累加和可以用sum等,这样的标志符本身就增加了程序的可读性,使程序更加清晰易懂。

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。如例句中用到的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。

需要注意的是:在C语言中,关键字都是小写的。

●ANSI C关键字 C语言以简洁、紧凑,使用方便、灵活而著称。ANSI C一共只有32个关键字。

表 1-2 ANSI C 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	sizof	struct	switch	typedef	union
unsigned	void	volatile	while			

• Turbo C关键字 Turbo C在ANSI C基础上扩充了11个关键字:

asm _cs _ds _es _ss cdecl far
hugeinterrupt near pascal

• C++关键字 在C++中,关键字很多,标准 C++中预定义了63个关键字,参见表1-3。另外,还定义了11个运算符关键字,它们是:and、and_eq、bitand、bitor、compl、not、not_eq、or、or_eq、xor ?、xor_eq。在我们后面要学习的内容中,并没有涉及到C++的所有关键字,但会逐步介绍最重要和最常用的一些关键字。

另外,有些标志符虽然不是关键字,但C++语言总是以固定的形式用于专门的地方,也不能把它们当做一般标志符使用,以免造成混乱。这样的标志符有include、define等,我们在后面的学习中会逐渐遇到。

表 1-3 C++关键字

asm	default	float	operator	static_cast	union
auto	delete	for	private	struct	unsigned
bool	do	friend	protected	switch	using
break	double	goto	public	template	virtual
case	dynamic_cast	if	register	this	void
catch	else	inline	reinterpret_cast	throw	volatile
char	enum	int	return	true	wchar_t
class	explicit	long	short	try	while
const	export	mutable	signed	typedef	
const_cast	extern	namespace	sizeof	typeid	
continue	false	new	static	typename	

1.2.5 C语言程序基本结构

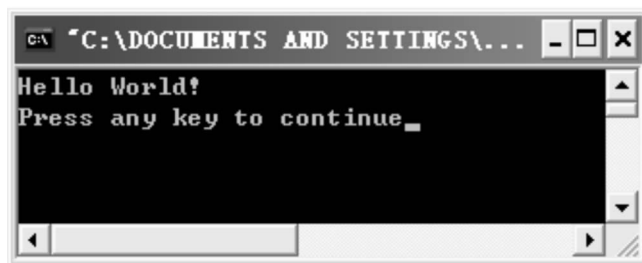
1. C源程序的基本结构

首先是C语言的经典例题,“Hello,world!”。

具体程序代码如下:

```
#include <stdio.h>
main()
{
    printf (“Hello World! \n”);
}
```

在Visual C++ 6.0中,经过编译、连接,运行时在程序运行窗口中显示如下内容:



这是一个最简单的C程序,它把字符串“Hello World!”显示在屏幕上。

该程序由一个函数main()(叫主函数)构成。任何一个程序都必须有此函数,花括号{}所括的内容是main的函数体,每个C语言程序的函数都至少有一对{}。

“printf(…)”是由系统提供的标准库函数,它完成输出功能,C语言的输出不像有的高级语言由语句完成,C语言的输出是由函数来完成的,这是它的特点之一,因此需要调用函数库中的函数printf()。为了顺利调用函数库中的函数printf(),还需要嵌入stdio.h这个头文件。头文件中定义了许多函数原型,C语言的系统中预定义了很多不同的头文件,供用户调用,以满足基本要求,在头文件stdio.h中就定义了函数printf()的原型,因此,这里要调用系统预定义函数printf(),就必须在程序的最前面增加#include<stdio.h>这个预处理命令,否则,printf()函数将无法工作。

#include是预编译程序命令,它把头文件“stdio.h”的内容展开在#include<stdio.h>所在的行位置处。“stdio.h”文件中定义了I/O库所用到的某些宏和变量。因此,在每一个引用标准库函数的程序中都必须带有该#include <stdio.h>命令行。

“Hello World!”是要输出的内容。“\n”表示换行字符,它是由“\”和“n”二字符构成,属转义字符,有关转义字符,在后面将会具体介绍。printf()后的分号是语句结束符,C语言的每一个语句都以分号“;”终止。

该程序运行时只是输出一行字——“Hello,world”。

我们再看一个实例:

此程序的功能是输入两个整数,输出其中的大数。

```
int max(int a,int b);
main() /* 主函数 */
{
    int x,y,z;
    printf("input two numbers:\n");
    scanf("%d%d",&x,&y);
    z=max(x,y); /* 调用max函数 */
    printf("maximum=%d\n",z); /* 输出 */
}
int max(int a,int b) /* 定义max函数 */
{
    if(a>b)
        return a;
    else
```