



21 世纪高等学校规划教材

徐香坤 谢进军 主编
杨柯 王岩 李莹 副主编

C 语言程序设计

 中国质检出版社



21 世纪高等学校规划教材

徐香坤 谢进军 主编
杨柯 王岩 李莹 副主编

C 语言程序设计

中国质检出版社

北京

图书在版编目(CIP)数据

C 语言程序设计/徐香坤,谢进军主编. —北京:中国质检出版社,2011.7

21 世纪高等学校规划教材/刘国普主编

ISBN 978-7-5026-3464-3

I. ①C… II. ①徐…②谢… 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(2011)第 159828 号

内容提要

全书共 12 章,包括 C 语言概述、程序设计基础知识、顺序结构程序设计、选择结构程序设计、循环结构程序设计、数组、函数、指针、预处理命令、结构体与共用体、位运算和文件。阐述概念层次清楚,结合程序示例使得读者容易理解和掌握相关的知识要点。

本书适合作为工程类和经管类专业高校教材,也可供其他大专院校师生参考。

中国质检出版社出版发行

北京市朝阳区和平里西街甲 2 号(100013)

北京市西城区复外三里河北街 16 号(100045)

网址:www.spc.net.cn

电话:(010)64275360 68523946

中国标准出版社秦皇岛印刷厂印刷

各地新华书店经销

*

开本 787×1092 1/16 印张 18 字数 447 千字

2011 年 7 月第 1 版 2011 年 7 月第 1 次印刷

*

定价 35.00 元

如有印装差错 由本社发行中心调换

版权专有 侵权必究

举报电话:(010)68510107

前 言

随着计算机技术的迅猛发展,社会对人才的要求也在不断提高。不仅要使用计算机获取专业领域知识,还要会使用计算机编写程序,解决专业领域中的具体问题。C语言作为一种结构化程序设计语言,以其短小精悍,功能齐全,具备其他高级语言所不具备的低级语言的特性,深受广大用户的喜爱。此外,其编写的程序具有良好的可移植性,可以在多种操作系统环境下运行。因此,C语言是目前国内外广泛使用的主流程序设计语言之一,已经在广阔的领域里得到了应用。同时,C语言程序设计也成为高等学校各专业开设的一门必修的计算机基础课程。

C语言程序设计重点在于培养学生的程序设计思想和程序设计能力,通过C语言程序设计这门课程的学习,使学生掌握C语言的语法规则,以及C语言程序设计的基本算法和各种控制结构,掌握程序设计的基本规律,逐步掌握程序设计的思想和方法,不断提高解决实际问题的能力,为后续高级程序设计语言的学习打下良好的基础。为此,本书的作者结合多年来的教学经验和应用C语言的体会,根据“计算机技术基础”课程的基本要求,在广泛参考有关资料的基础上编写了这本教材。

本教材选用 Visual C++ 6.0 作为教学环境,深入浅出地讲解C语言的概念、基本语法和程序设计的基本思想。编写时力求“概念清晰,层次分明,通俗易懂”,使读者很容易理解和掌握本教材的内容。同时教材中精选了大量的案例程序,使读者更容易理解相关的语法知识。教材中所有的案例都在 Visual C++ 6.0 集成环境中调试通过,希望读者学好语法知识,多上机实践,多思考,多模仿,通过上机演示例题和习题,来逐步掌握C语言程序设计的基本思想和方法。全书共分12章,内容包括C语言的基本概念、基本数据类型、运算符和表达式、数据的输入输出、流程控制语句、数组、函数、指针、编译预处理、结构体和共用体,以及文件。通过本教材的学习使读者真正学会用C语言去编写程序,体会学习C语言的无穷乐趣。

本教材第1、第2、第12章由王岩编写,第3、第4、第5章由徐

香坤编写,第6、第7章由杨柯编写,第8、第9、第11章由谢进军编写,第10章由李莹编写。全书由徐香坤统稿。

本教材在编写过程中,得到了孙承福教授的支持和帮助,在此表示深深的谢意。

感谢读者选择使用本书,由于作者的水平和经验有限,本教材中难免存在错误或疏漏之处,真诚地欢迎各位专家和读者批评指正,以帮助我们进一步把教材完善。

作 者

2011年5月

目 录

第 1 章 C 语言概述	1
1.1 程序设计概述	1
1.2 算法	2
1.3 C 语言简介	5
1.4 C 程序的上机步骤	9
第 2 章 C 程序设计基础知识	15
2.1 数据类型与标识符	15
2.2 常量与变量	17
2.3 运算符和表达式	26
第 3 章 顺序结构程序设计	35
3.1 语句概述	35
3.2 赋值语句	37
3.3 字符输出/字符输入函数	37
3.4 格式化输出/输入函数	38
3.5 顺序结构程序设计举例	44
第 4 章 选择结构程序设计	52
4.1 关系运算符与关系表达式	52
4.2 逻辑运算符与逻辑表达式	54
4.3 if 语句	55
4.4 条件运算符与条件表达式	60
4.5 switch 语句	60
4.6 选择结构程序设计举例	62
第 5 章 循环结构程序设计	73
5.1 while 语句	73
5.2 do...while 语句	74
5.3 for 语句	76
5.4 循环的嵌套	79
5.5 break 语句和 continue 语句	81
5.6 几种循环语句的比较	82
5.7 循环结构程序设计举例	83
第 6 章 数组	93
6.1 一维数组的定义与引用	93
6.2 二维数组的定义与引用	98
6.3 字符数组的定义与引用	101
第 7 章 函数	121
7.1 函数概述	121
7.2 函数的定义与调用	122
7.3 函数的声明	130
7.4 函数的嵌套调用	131
7.5 函数的递归调用	133
7.6 数组作为函数参数	137

7.7	局部变量和全局变量	143
7.8	变量的存储类别	147
7.9	内部函数和外部函数	155
第8章	指针	167
8.1	指针概述	167
8.2	指针和指针变量	168
8.3	指针和数组	174
8.4	指针和字符串	186
8.5	指针和函数	190
8.6	指针数组和指向指针的指针	193
第9章	预处理命令	209
9.1	概述	209
9.2	宏定义	209
9.3	文件包含	217
9.4	条件编译	219
第10章	结构体与共用体	224
10.1	概述	224
10.2	结构体的定义	225
10.3	结构体变量的定义	225
10.4	结构体变量的引用	227
10.5	结构体变量的初始化	227
10.6	结构体数组	228
10.7	结构体指针	230
10.8	共用体	232
10.9	枚举类型	234
10.10	类型重定义 typedef	235
10.11	结构体指针的应用——链表	236
第11章	位运算	244
11.1	概述	244
11.2	位运算符	244
11.3	位运算	244
11.4	位段	248
第12章	文件	252
12.1	C 文件概述	252
12.2	文件指针	253
12.3	文件的打开与关闭	253
12.4	文件的读写	255
12.5	文件定位	263
12.6	文件检测函数	265
附录一	C 语言的关键字	268
附录二	ASCII 与字符对照表	269
附录三	常用 C 语言库函数	272
附录四	C 语言的运算符及其结合性	279
	参考文献	281

第1章 C语言概述

学习指导

C语言是一种使用广泛的程序设计语言,它拥有丰富的数据结构、结构化的流程控制和较高效率的目标代码。C语言作为一种高级语言与其他高级语言相比功能更强,它既具有高级语言的功能也具有低级语言的许多功能,因此既可以用于编写系统软件,又可以用于编写应用软件。本章主要介绍程序设计的基础,算法的概念及特性,C语言的产生及特点,简单介绍了C程序的结构,以及如何在VC环境下开发C程序。

1.1 程序设计概述

1.1.1 程序与程序设计

随着计算机的普及,人们可以利用它来完成各种各样的事情,无论是复杂的计算、游戏娱乐等都可以,计算机能完成如此繁杂的工作,正是由于有不同程序的支持,而程序正是人们利用程序语言经过程序设计过程得到的智力结晶。更确切地说,所谓程序,就是用计算机语言对所要解决的问题中的数据以及处理问题的方法和步骤所做的完整而准确的描述,这个描述的过程就称为程序设计。著名的计算机科学家沃思(Nikiklaus Wirth)提出了关于程序的著名公式如下:

程序 = 数据结构 + 算法

其中:

- ① 数据结构(data structure):对数据的描述。在程序中要指定数据的类型和数据的组织形式。
- ② 算法(algorithm):对操作的描述。即操作步骤。

知道了程序的概念之后,到底应该如何进行程序设计呢?一般包含如下4个步骤:

- ① 分析问题,建立数学模型。对问题进行分析,找出已知的数据和条件,确定输入、处理及输出对象。将解题过程归纳为一系列的数学表达式,建立各种量之间的关系,即建立起解决问题的数学模型。
- ② 确定数据结构和算法。针对建立的数学模型确定数据结构,选择合适的算法加以实现。
- ③ 编制程序。根据确定的数据和算法,用程序语言描述出来,即编写程序代码。
- ④ 调试程序。在计算机上运行已编好的程序,分析得到的结果,进行测试和调整,直至得到预期结果。

综上一个完整的程序要涉及4方面的问题:数据结构、算法、编程语言和程序设计方法。

1.1.2 结构化程序设计方法

程序设计方法按照程序的结构性质,有结构化程序设计与非结构化程序设计之分。前者

是指具有结构性的程序设计方法与过程。它具有由基本结构构成复杂结构的层次性,后者反之。其中结构化程序设计方法是公认的面向过程编程应遵循的基本方法和原则。结构化程序设计方法的基本思想是,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,就像玩积木游戏那样,用简单的模块搭建起复杂的程序。

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,怎样才能得到一个结构化的程序呢?具体应采取的方法为:自顶向下,逐步细化,模块化设计,结构化编码。采用模块化结构,自上而下,逐步求精。即首先把一个复杂的大问题分解为若干相对独立的小问题。若小问题仍复杂,则继续细化,直至将小问题分解为语句。然后,对应每一个小问题编写出一个功能上相对独立的程序块来,这种具有一定功能的程序块被称为模块,确切地说,模块是程序对象的集合,模块化就是把程序划分成若干个模块,每个模块完成一个确定的功能,把这些模块集中起来组成一个整体,就可以完成对问题的求解。结构化编码,指程序基于三种基本的结构:顺序、选择及循环结构。由这三种基本结构组成的程序,就是结构化程序。

1. 顺序结构

顺序结构是最简单的一种基本结构,指程序流沿着一个方向,按先后顺序逐条执行,没有分支,没有转移。

2. 选择结构

选择结构也称分支结构,该结构中的语句在执行时,程序将根据不同的条件执行不同分支。根据分支的情况,选择结构有单分支、双分支和多分支三种形式。

3. 循环结构

循环结构是指根据条件,反复执行某一部分的语句。循环结构有两种形式:当型循环和直到型循环。

① 当型循环,是先判断条件,若成立,则重复执行某个操作;不成立,则结束循环。

② 直到型循环,是先执行,后判断。重复执行某一操作,直到条件成立为止,不成立则继续循环。

1.2 算 法

1.2.1 算法的概念

广义地说,算法就是为解决问题而采取的方法和步骤。计算机要完成一定的任务,也是按照指定的步骤执行一系列的操作,计算机中的算法是指用计算机语言解决问题的方法和步骤。因此,算法是程序的组成部分之一。

计算机中算法可分为两类:数值运算算法和非数值运算算法。数值运算算法的目的是求数值解,例如求平方根、求三角形的面积等数学运算,是基于数学模型的算法。非数值运算算法涉及面很广,最常见的如人事管理、图书检索、财务管理等。

1.2.2 算法的特性

(1) 有穷性:一个算法应包含有限的操作步骤而不能是无限的。

- (2) 确定性:算法中每一个步骤应当是确定的,而不应当是含糊的、模棱两可的。
- (3) 有零个或多个输入。所谓输入是指在执行算法时需要从外界取得必要的信息。
- (4) 有一个或多个输出。算法得到的结果就是输出。
- (5) 有效性:算法中每一个步骤应当能有效地执行,并得到确定的结果。

1.2.3 算法的表示方法

算法有两个要素:基本功能操作和控制结构。其基本功能操作是用来描述数据运算和传输的,控制结构,即顺序、选择和循环三种基本结构。对算法的描述有多种方法,常用的有自然语言、传统流程图、结构化流程图以及伪代码等。

1. 自然语言表示法

自然语言表示方法虽然通俗易懂,但由于文字冗长,使用不方便,容易出现“歧义性”,所以,只有在描述简单问题时,才会被使用。

【例 1.1】 输入三个数 A、B、C,然后输出最大数 MAX。用自然语言写出算法。

算法表示如下:

- (1) 输入 A,B,C;
- (2) 若 $A > B$,则 $A \rightarrow \text{MAX}$;若 $A < B$,则 $B \rightarrow \text{MAX}$ 。
- (3) 若 $C > \text{MAX}$,则 $C \rightarrow \text{MAX}$ 。
- (4) 输出 MAX,MAX 即是最大数。

2. 传统流程图表示法

流程图也称框图,是用一些几何框图、流程线和文字说明表示各种类型的操作。形象直观,易于理解。ANSI 规定了一些常用的流程图符号。如图 1.1 所示。

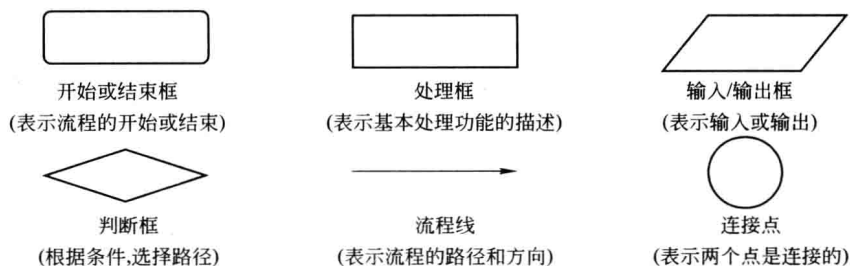


图 1.1 流程图中的符号

传统的流程图用流程线指出各框图的执行顺序,形象直观,简单方便。下面用传统的流程图来表示三种基本结构。

(1) 顺序结构

如图 1.2 所示,A、B 两个框是顺序执行的。即在执行完 A 之后,接着执行 B。

(2) 选择结构

如图 1.3 所示,判断框里放的是条件 P,判断条件 P 成立与否。若成立,则执行框 A;不成立,则执行框 B。

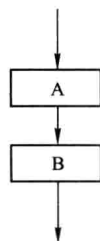


图 1.2 顺序结构

(3) 循环结构

当型循环如图 1.4 所示,先执行判断框,当条件 $P1$ 成立时,执行框 A 操作,执行完 A 后,再判断条件 $P1$,若成立,再执行 A ,如此反复执行 A ,直至条件 $P1$ 不满足为止,结束循环。

直到型循环如图 1.5 所示,先执行 A 框,然后判断条件 $P2$,若条件不成立,则再执行 A ,然后再判断 $P2$,不成立,再执行 A ,如此反复,直至条件 $P2$ 成立为止,结束循环。

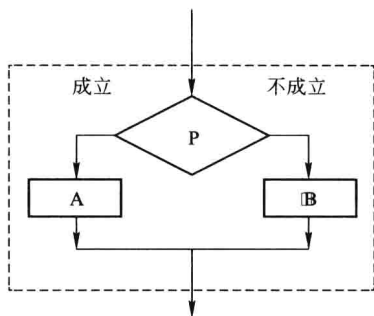


图 1.3 选择结构

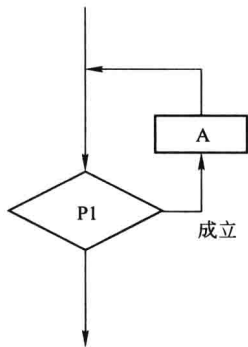


图 1.4 当型循环

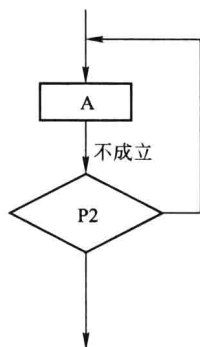


图 1.5 直到型循环

以上三种基本结构,有以下两个共同特点:

- ① 只有一个入口和出口。
- ② 结构内的每一部分都有执行的机会。

【例 1.2】 将例 1.1 用传统流程图描述其算法。见图 1.6。

3. N-S 流程图表示法

传统的流程图,由于对流程线的走向没有任何限制,可以任意转向,在描述复杂的算法时所占篇幅较多,费时费力且不易阅读。一种新的流程图形式于 1973 年被美国学者提出,这种流程图完全去掉了流程线,算法的每一步都用一个矩形框来描述,把一个个矩形框按执行的次序连接起来就是一个完整的算法描述。这种流程图称为 N-S 图。下面用 N-S 图描述三种基本结构,见图 1.7~图 1.10。

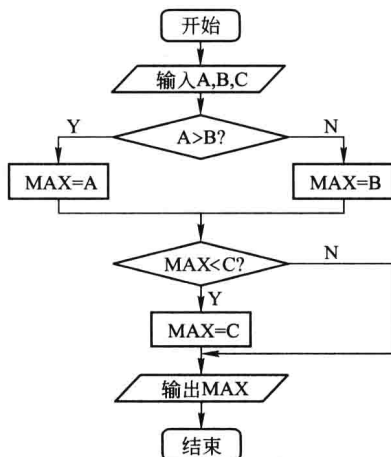


图 1.6

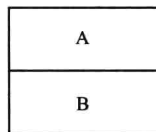


图 1.7 顺序结构

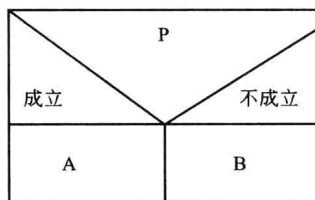


图 1.8 选择结构

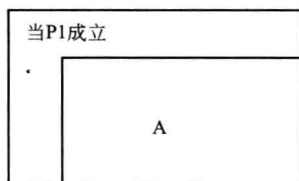


图 1.9 当型循环结构

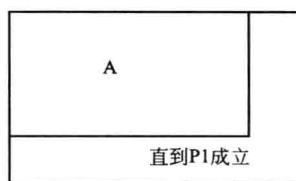


图 1.10 直到型循环结构

【例 1.3】 将例 1.1 用 N-S 流程图描述其算法。见图 1.11。

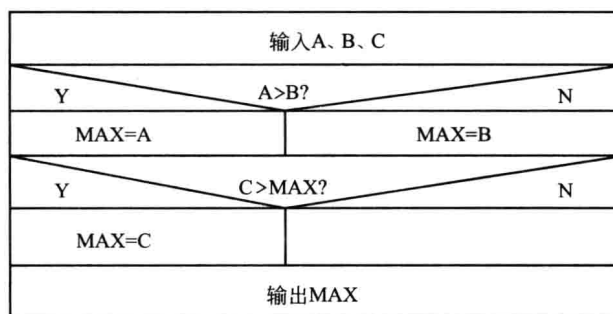


图 1.11

4. 伪代码表示法

伪代码是用介于自然语言和计算机语言之间的文字和符号来描述算法。自上而下按照顺序写，每一行表示一个操作，不用图形符号，因此，书写方便，结构紧凑，比较好懂，同时也方便向计算机程序过渡。

1.3 C 语言简介

程序设计语言是用于编写计算机程序的语言。按照语言级别可以分为低级语言和高级语言。低级语言有机器语言和汇编语言。低级语言与特定的机器有关、功效高，但使用复杂、繁琐、费时、易出差错。机器语言是表示成数码形式的机器基本指令集，或者是操作码经过符号化的基本指令集。汇编语言是机器语言中地址部分符号化的结果，或进一步包括宏构造。高级语言的表示方法要比低级语言更接近于待解问题的表示方法，其特点是在一定程度上与具体机器无关，易学、易用、易维护。C 语言作为一种高级语言与其他高级语言相比功能更强，它既具有高级语言的功能也具有低级语言的许多功能，因此也被称为中级语言，既可以用于编写系统软件，又可以用于编写应用软件。

1.3.1 C 语言的发展过程

C 语言是由美国贝尔实验室的 Dennis M. Ritchie 于 1972 年设计实现的。C 语言的起源可以追溯到 ALGOL 60 (ALGO rithm Language)。ALGOL 60 是 1960 年由国际计算机委员

会设计的一种面向过程的结构化程序设计语言,用它编写的程序具有可读性和可移植性好的特点,但是,它不能直接对硬件进行操作,不宜用来编写系统软件。系统程序主要用汇编语言编写,而汇编语言是面向机器的程序语言,用它编写的程序可读性和可移植性都比较差。为此,人们开始考虑设计一种集高级语言和低级语言功能于一身的语言,用以编写可读性和可移植性都较好的系统软件。

1963 年,英国的剑桥大学在 ALGOL 60 的基础上推出了 CPL(Combined Programming Language)语言,该语言较接近硬件,但太复杂难以实现。1967 年英国剑桥大学的 Martin Richards 对 CPL 进行了简化,推出了 BCPL(Basic Combined Programming Language)语言。1970 年,美国贝尔实验室的 Ken Thompson 将 BCPL 进一步改进为 B 语言,并用 B 语言开发了第一个高级语言的 UNIX 操作系统。B 语言是一种解释性语言,功能上也不够强,为了更好的适应系统程序设计的要求,1972 年,Dennis M. Ritchie 把 B 语言发展为 C 语言,C 语言既保持了 BCPL 和 B 语言的精练和接近于硬件的特点,也克服了它们过于简单、数据无类型等缺点。1973 年,Ken Thompson 和 Dennis M. Ritchie 又合作将 1969 年用汇编语言编写的 UNIX 操作系统改用 C 语言编写,其中 C 语言代码占 90% 以上,只保留了少量汇编语言代码。这样就使得 UNIX 操作系统向其他类型的机器上移植变得相当简单。到 20 世纪 70 年代中期,UNIX 操作系统和 C 语言作为软件设计的良好工具开始风靡世界。



图 1.12 C 语言的产生

图 1.12 演示了 C 语言的产生过程。

1978 年,以 UNIX 第 7 版中的 C 编译程序为基础,Brian W. Kernighan 和 Dennis M. Ritchie 合著了影响深远的名著《The C Programming Language》。这本书中介绍的 C 语言成为后来广泛使用的 C 语言版本的基础,称为 K&R C 语言,从而使 C 语言成为目前世界上流行最广泛的高级程序设计语言。1983 年,美国国家标准化协会(ANSI)根据 C 语言问世以来各种版本对 C 的发展和扩充,制定了新的标准,称为 ANSI C。1987 年,美国国家标准化协会(ANSI)又公布了新的 C 语言标准,称为 87 ANSI C。1990 年,国际标准化组织(ISO)接受 87 ANSI C 为 ISO C 的标准,称为标准 C。目前广泛流行的各种版本 C 语言都是以它为基础的。

1.3.2 C 语言的特点

C 语言既具有高级语言的特点又具有低级语言的特点,使之得以存在和发展,并具有生命力。具体来说,C 语言的主要特点如下。

(1) C 语言简洁、紧凑,使用方便、灵活。

ANSI C 一共只有 32 个关键字

(详见表 2.1)。9 种控制语句,程序书写形式自由。

(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) 数据结构丰富,可以实现各种复杂的数据结构的运算。

(4) 具有结构化的控制语句,用函数作为程序的模块单位,所以,C 语言是一种结构化的程序设计语言。

(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。可用于编写系统软件,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 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。

1.3.3 C 程序的结构组成与书写风格

任何一种计算机程序语言,都具有特定的语法规则和一定的表现形式。按照规定的结构编写程序,不仅使程序设计人员和使用程序的人容易理解,更重要的是,程序输入进计算机时,计算机能够充分认识,从而能正确执行出结果。

先通过一个简单的例子来认识一下 C 语言程序的基本结构和书写格式。

【例 1.4】 编写程序,在屏幕上打印出 hello,world!。

```
#include<stdio. h>
void main()
{
    printf("hello,world!\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 1.5】 求一个数的正弦值。

```
#include<math. h>
#include<stdio. h>
void main()
{
    double x,s;
    printf("input number:\n");
    scanf("%lf",&x);
    s=sin(x);
    printf("sine of %lf is %lf\n",x,s);
}
```

- `include` 称为文件包含命令。
- 扩展名为 `.h` 的文件称为头文件。
- 定义两个实数变量, 以被后面程序使用。
- 显示提示信息。
- 从键盘获得一个实数 `x`。
- 求 `x` 的正弦, 并把它赋给变量 `s`。
- 显示程序运算结果。
- `main` 函数结束。

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`。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<stdio.h>`。

同样, 在例 1.4 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, 另一部为分执行部分。说明是指变量的类型说明。例题 1.4 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。

运行本程序时, 首先在显示器屏幕上给出提示串 `input number`, 这是由执行部分的第一行完成的。用户在提示下从键盘上键入某一数, 如 5, 按下回车键, 接着在屏幕上给出计算结果。

在前两个例子中用到了输入和输出函数 `scanf` 和 `printf`, 在以后要详细介绍。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 3 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印。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。

通过以上的例子,可以看到如下 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) 标识符,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。

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。

1.4 C 程序的上机步骤

1.4.1 C 程序的执行过程

C 语言作为一门高级语言,更接近于人们自然语言,独立于机器,编码相对简单,可读性强。用高级语言编写出来的程序称为源程序,而计算机只能识别 0 和 1,因此一个 C 程序必须通过编辑、编译、连接和运行后,才能得到运行结果。

1. 编辑

编辑是指输入 C 语言源程序并进行修改,最后以文本文件的形式存放在磁盘上,文件名由用户自行定义,最好具有一定的意义,扩展名一般为“.c”,例如 sample.c, test.c 等。

2. 编译

把 C 源程序翻译成可重定位的二进制目标程序。编译过程对源程序进行句法和语法检查,当发现句法和语法错误时,就会将错误的类型和在程序中的位置显示出来,以帮助用户修改源程序中的错误。如果没发现错误,就自动形成目标代码进行优化后生成目标文件。扩展名一般为“.obj”。

3. 连接

连接也称链接或装配,是用连接程序将编译过的目标程序和程序中用到的库函数装配在一起,形成可执行的目标程序。扩展名为“.exe”。

4. 运行

将可执行的目标文件投入运行由计算机执行后,可得到程序的运行结果。

1.4.2 在 Visual C++ 环境下建立和运行 C 程序的步骤

Visual C++是一个可视化的集成环境,在此环境下可以开发 C 程序,下面介绍如何新建或打开 C 程序,以及如何编辑、编译、连接和运行 C 程序。

1. 启动 Visual C++

选择【开始】|【程序】|【Microsoft Visual Studio 6.0】|【Microsoft Visual C++ 6.0】菜单,启动 Visual C++编译系统。

2. 建立一个 C 程序项目

选择【文件】|【新建】菜单,在“新建”窗口的“工程”选项卡中选择“Win32 Console Application”项,在右侧的“位置”栏中选择 D 盘 myc 文件夹,“工程”栏中输入“c1”,这时界面变为如图 1.13 所示。单击“确定”后,在出现的应用框架选择向导窗口和新建工程信息窗口中分别单击“完成”和“确定”按钮,完成新工程的建立,如图 1.14 所示。



图 1.13 “新建”窗口的“工程”选项卡