

推荐教材

面向 21 世纪高等院校计算机规划教材

C 语言教程与等级考试

C YU YAN JIAO CHENG YU DENG JI KAO SHI

主编 武俊生 邵回祖

主审 陈立潮



中国铁道出版社
CHINA RAILWAY PUBLISHING HOUSE

面向 21 世纪高等院校计算机规划教材

C 语言教程与等级考试

武俊生 邵回祖 主 编

米晓萍 曹学飞 张 举 副主编

齐兴斌 郭新峰 赵 丽 王若慧 杨建英 参 编

陈立潮 主 审

内 容 简 介

本书是根据全国计算机等级考试的最新要求编写的,全书共 10 章: C 语言简介,变量、数据类型和运算符,基本语句与编译预处理,选择结构程序设计,循环结构程序设计,数组,函数,指针,结构体与共用体、位运算与文件等。

本书的主要特色是注重基础与程序设计思想的培养,内容循序渐进,并覆盖全国计算机等级考试相应的试题。每章都有学习目标、现场练习、实例、小结和习题,能够满足不同专业学生的要求。通过大量的例题、习题,力求通过实际问题的讲解,逐步提高学生编写程序的能力,通过大量的等级考试实例来提高学生的应试能力,同时,也加强了操作技能的实用性。

本书可以作为各类应用型本、专科计算机专业及非计算机专业的教材,也可作为计算机等级考试的复习用书,以及各类计算机培训班教材或初学者的自学用书。

图书在版编目(CIP)数据

C 语言教程与等级考试 / 武俊生,邵回祖主编.

—北京:中国铁道出版社,2012.4

面向 21 世纪高等院校计算机规划教材

ISBN 978-7-113-14358-9

I. ①C… II. ①武… ②邵… III. ①C 语言—程序设计—高等学校—教学参考资料 IV. ①TP312

中国版本图书馆 CIP 数据核字(2012)第 057469 号

书 名: C 语言教程与等级考试

作 者: 武俊生 邵回祖 主编

策 划: 侯 伟

读者热线: 400-668-0820

责任编辑: 冯彩茹

编辑助理: 包 宁

封面设计: 付 巍

封面制作: 刘 颖

责任印制: 李 佳

出版发行: 中国铁道出版社(100054,北京市西城区右安门西街 8 号)

网 址: <http://www.51eds.com>

印 刷: 北京海淀五色花印刷厂

版 次: 2012 年 4 月第 1 版 2012 年 4 月第 1 次印刷

开 本: 787mm×1092mm 1/16 印张: 15 字数: 362 千

书 号: ISBN 978-7-113-14358-9

定 价: 29.00 元

版权所有 侵权必究

凡购买铁道版图书,如有印制质量问题,请与本社教材图书营销部联系调换。电话:(010) 63550836

打击盗版举报电话:(010) 63549504

前 言

C 语言是针对高校各专业开设的第一门程序设计基础课程。本书为了适应各学科新形势发展的需要,为后续学习其他语言奠定基础,同时也是为使同学们在学完本课程后顺利通过二级 C 语言等级考试而编写的,主要是为学生提供一本既有理论基础,又更加注重操作技能实用性的程序设计教程与等级考试指南。本书针对高校教学的特点,不仅注重基础知识的系统性和基本概念的正确性,同时也更强调应用性和实用性。

本书的使用环境是 Visual C++ 6.0,书中所有源程序均已经在此环境下调试并成功运行。

通过学习后,掌握 C 语言程序设计的基本思想、方法和解决实际问题的应用技巧。

本书内容的主要特色是注重基础与程序设计思想的培养,循序渐进,并覆盖全国计算机等级考试相应的试题。每章都有学习目标、现场练习、实例、小结和习题,以满足不同专业学生的要求。通过大量的例题、习题,力求通过实际问题的讲解,逐步提高学生编写程序的能力,通过大量的等级考试实例来提高学生的应试能力。

本书由武俊生、邵回祖任主编,米晓萍、曹学飞、张举任副主编,齐兴斌、郭新峰、赵丽、王若慧、杨建英参与编写,陈立潮教授主审。其中,第 1 章由杨建英编写,第 2 章由张举编写,第 3 章由齐兴斌编写,第 4 章由王若慧编写,第 5 章由赵丽编写,第 6 章由郭新峰编写,第 7 章由曹学飞编写,第 8 章由邵回祖编写,第 9 章由武俊生编写,第 10 章由米晓萍编写。

本书每章后都设计了大量的习题、实验、综合实训,便于学生巩固复习,并在附录部分附有二级 C 语言考试大纲及全真二级 C 语言笔试试题与参考答案。本书以突出应用、强调技能为目标,同时覆盖全国计算机等级考试相关考试内容。

限于编者水平,书中难免出现不足和疏漏,恳请读者批评指正。

编 者

2012 年 2 月

目 录

第 1 章 C 语言简介	1	2.4.8 数据间的混合运算与 类型转换	27
1.1 C 语言概述	1	小结	28
1.1.1 C 语言的发展过程	1	习题	29
1.1.2 C 语言的主要特点	2	第 3 章 基本语句与编译预处理	31
1.2 程序的算法及流程图	3	3.1 预处理命令	31
1.2.1 算法	3	3.1.1 宏定义	32
1.2.2 流程图	5	3.1.2 文件包含处理	34
1.3 C 语言程序介绍	6	3.2 C 语言中的语句	35
1.4 C 语言程序运行的步骤和方法	8	3.3 输入与输出函数的调用	35
1.4.1 Visual C++ 6.0 环境	8	3.3.1 putchar()函数	36
1.4.2 编写与运行 C 语言程序	9	3.3.2 getchar()函数	37
1.4.3 程序的简单调试	11	3.3.3 printf()函数	38
小结	12	3.3.4 scanf()函数	42
习题	12	小结	46
第 2 章 变量、数据类型和运算符	13	习题	46
2.1 概述	13	第 4 章 选择结构程序设计	48
2.2 常量与变量	14	4.1 if 语句	48
2.3 基本数据类型	16	4.1.1 简单 if 语句的两种基本 形式	48
2.3.1 整型数据类型	16	4.1.2 if 语句的嵌套	51
2.3.2 实型数据类型	18	4.1.3 if 语句实例	54
2.3.3 字符型数据类型	19	4.2 switch 语句	56
2.3.4 字符串数据类型	19	4.2.1 switch 语句	56
2.4 运算符和表达式	20	4.2.2 switch 语句实例	58
2.4.1 运算符与表达式简介	20	4.3 if 语句和 switch 语句的比较	59
2.4.2 算术运算符与表达式	21	4.4 条件运算符	59
2.4.3 赋值运算符与表达式	23	小结	60
2.4.4 关系运算符与表达式	24	习题	61
2.4.5 逻辑运算符与表达式	25	第 5 章 循环结构程序设计	64
2.4.6 sizeof 运算符与表 达式	26	5.1 概述	64
2.4.7 运算符的优先级及结合 方式	26	5.2 while 循环	65

5.3 do...while 循环.....67	6.4.7 常用字符串处理函数.....100
5.4 while 循环和 do...while 循环的 比较68	6.4.8 字符数组的应用104
5.5 for 循环69	小结105
5.6 转移语句及其构成的循环.....71	习题105
5.7 3 种循环语句的比较72	第 7 章 函数 112
5.8 break 和 continue 跳转语句72	7.1 概述112
5.8.1 break 跳转语句.....72	7.2 函数定义的一般形式114
5.8.2 continue 跳转语句.....73	7.2.1 无参函数.....114
5.9 循环的嵌套74	7.2.2 有参函数.....114
5.10 程序实例76	7.2.3 空函数115
小结78	7.3 常用的库函数.....115
习题79	7.4 函数的调用.....116
第 6 章 数组83	7.4.1 区分形参与实参117
6.1 数组的基本概念83	7.4.2 函数的参数数据传递.....118
6.2 一维数组的定义与应用84	7.4.3 数组作为函数参数118
6.2.1 一维数组的定义84	7.5 函数声明120
6.2.2 一维数组的存储结构85	7.6 函数的嵌套与递归调用121
6.2.3 一维数组元素的引用86	7.6.1 函数的嵌套调用121
6.2.4 一维数组的初始化87	7.6.2 函数的递归调用124
6.2.5 一维数组程序的应用87	7.7 变量的作用域.....126
6.3 二维数组的定义与应用90	7.7.1 局部变量.....127
6.3.1 二维数组的定义90	7.7.2 全局变量.....127
6.3.2 二维数组元素的存储 结构92	7.8 变量存储类型130
6.3.3 二维数组元素的引用93	7.8.1 自动、寄存器和外变量 存储类型130
6.3.4 二维数组的初始化94	7.8.2 static 存储类型131
6.3.5 二维数组程序的应用95	7.9 内部函数与外部函数的概念.....133
6.4 字符数组96	7.9.1 内部函数.....133
6.4.1 字符数组的定义96	7.9.2 外部函数.....134
6.4.2 字符数组的引用96	小结135
6.4.3 字符数组的初始化96	习题135
6.4.4 字符串及字符串结束 标记98	第 8 章 指针 137
6.4.5 字符数组与字符串的 区别98	8.1 指针的基本概念137
6.4.6 字符数组的输入与输出.....99	8.2 指针变量的定义和引用138
	8.2.1 指针变量的定义138
	8.2.2 指针变量的引用139
	8.2.3 指针变量作函数参数.....141

8.2.4 指针作函数返回值	142	9.4.4 共用体类型变量的 赋值和使用	174
8.3 数组与指针	143	9.5 枚举类型	175
8.3.1 一维数组和指针	144	9.6 用 typedef 定义类型	176
8.3.2 二维数组和指针	146	小结	178
8.3.3 用数组名作函数参数	150	习题	178
8.4 字符串与指针	152	第 10 章 位运算与文件	181
8.4.1 通过赋初值的方式使 指针指向一个字符串	152	10.1 位运算符及位运算	181
8.4.2 通过赋值运算使指针 指向一个字符串	153	10.1.1 位逻辑运算	181
8.4.3 字符指针作函数参数	154	10.1.2 移位运算	183
8.5 指针数组与多级指针	154	10.1.3 不同长度的数据 进行位运算	184
8.5.1 指针数组	154	10.2 文件的概述	185
8.5.2 多级指针	156	10.2.1 文件的概念	185
8.6 函数指针	157	10.2.2 文件的分类	186
小结	158	10.2.3 文件指针	186
习题	159	10.3 打开文件和关闭文件	187
第 9 章 结构体与共用体	161	10.3.1 文件操作的步骤	187
9.1 结构体与共用体概述	161	10.3.2 文件的打开	188
9.2 结构体类型定义和使用	161	10.3.3 文件的关闭	188
9.2.1 定义结构体类型的 语法	161	10.4 文件的顺序读写	189
9.2.2 定义结构体类型变量	162	10.4.1 字符读写函数	189
9.2.3 结构类型变量初始化	163	10.4.2 字符串读写函数	190
9.2.4 结构体变量的引用	164	10.4.3 格式化读写函数	192
9.2.5 结构体类型数组	166	10.4.4 数据块读写函数	192
9.3 用指针处理链表	168	10.5 文件的定位与随机读/写	194
9.3.1 链表	168	10.5.1 位置指针复位函数 rewind()	194
9.3.2 动态存储分配	169	10.5.2 随机位置指针函数 fseek()	195
9.3.3 链表的基本操作	169	10.5.3 返回文件当前位置的 函数 ftell()	196
9.4 共用体类型的定义和使用	173	10.6 文件检测函数	196
9.4.1 定义共用体类型的 语法	173	10.6.1 读写文件出错检测函数 ferror()	196
9.4.2 定义共用体类型 变量	173	10.6.2 文件结束检测函数 feof()	196
9.4.3 共用体类型变量的初 始化	174		

10.6.3 文件出错标志和文件结束标志置零函数 clearerr().....	197
小结	197
习题	198
附录 A 全国计算机等级考试二级 C 语言考试大纲	202
附录 B 2011 年 3 月全国计算机等级考试 二级 C 语言笔试试题及 参考答案	205
附录 C 部分字符的 ASCII 码对照表	215
附录 D Visual C++ 6.0 编译错误信息	216
附录 E 常用 C 库函数	219
附录 F 课后习题答案	223

第 1 章 | C 语言简介

本章要点:

- 了解 C 语言发展史及其特点。
- 理解程序、算法和流程图的概念。
- 熟练掌握 C 程序的结构。
- 熟练使用 Visual C++ 6.0 编辑和运行 C 程序。
- 熟练在 Visual C++ 6.0 下进行程序调试。

1.1 C 语言概述

程序设计语言 (Programming Language) 是用于书写计算机程序的语言。语言的基础是一组记号和一组规则。根据规则由记号构成的记号串的总体就是语言。在程序设计语言中, 这些记号串就是程序。程序设计语言有 3 方面的因素, 即语法、语义和语用。

语法表示程序的结构或形式, 亦即表示构成语言的各个记号之间的组合规律, 但不涉及这些记号的特定含义, 也不涉及使用者。

语义表示程序的含义, 亦即表示按照各种方法所表示的各个记号的特定含义, 但不涉及使用者。

语用表示程序与使用者的关系。

还有许多用于特殊用途的语言, 但只在特殊情况下使用。例如, PHP 专门用来显示网页; Perl 更适合文本处理; C 语言被广泛用于操作系统和编译器 (所谓的系统编程) 的开发。

按语言级别来划分, 可分为低级语言和高级语言。低级语言包括字位码、机器语言和汇编语言。低级语言的特点与特定的机器有关, 且功效高, 但使用复杂、烦琐、费时、易出差错。其中, 字位码是计算机唯一可直接理解的语言, 但由于它是一连串的字位, 所以比较复杂、烦琐、冗长, 几乎无人直接使用。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较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的机器无关, 易学、易用、易维护。当高级语言程序翻译成相应的低级语言程序时, 一般说来, 一个高级语言程序单位要对应多条机器指令, 相应的编译程序所产生的目标程序往往功效较低。而 C 语言是一门高级语言。

1.1.1 C 语言的发展过程

C 语言是目前世界上流行、使用最广泛的高级程序设计语言之一。

对操作系统和系统使用程序以及需要对硬件进行操作的场合,使用 C 语言明显优于其他高级语言,许多大型应用软件也都是用 C 语言编写的。

C 语言具有绘图能力强,可移植性,并具备很强的数据处理能力,因此适于编写系统软件、三维图形、二维图形和动画,它是数值计算的高级语言。

常用的编译软件有 TC 和 Microsoft Visual C++等。

C 语言的发展颇为有趣,它的原型是 ALGOL 60 语言。

1963 年,剑桥大学将 ALGOL 60 语言发展成为 CPL (Combined Programming Language) 语言。

1967 年,剑桥大学的 Martin Richards 对 CPL 语言进行了简化,于是产生了 BCPL 语言。

1970 年,贝尔实验室的 Ken Thompson 将 BCPL 语言进行了修改,并为它起了一个有趣的名字“B 语言”,意思是将 CPL 语言煮干,提炼出它的精华。而且他用 B 语言编写了第一个 UNIX 操作系统。

在 1973 年,贝尔实验室的 D.M.Ritchie 在 B 语言的基础上设计出了一种新的语言,取 BCPL 的第二字母作为这种语言的名字,这就是 C 语言。为了使 UNIX 操作系统得到推广,1977 年 D.M.Ritchie 发表了不依赖于具体机器系统的 C 语言编译文本《可移植 C 语言编译程序》。

1978 年 B.W.Kernighan 和 D.M.Ritchie 出版了 *The C Programming Language* 一书,从而使 C 语言成为世界上流行最广泛的高级程序设计语言。

1988 年,随着微型计算机的日益普及,出现了许多 C 语言版本。由于没有统一的标准,使得这些 C 语言之间出现了一些不一致的地方。为了改变这种情况,美国国家标准研究所 (ANSI) 为 C 语言制定了一套 ANSI 标准,成为现行的 C 语言标准。

1.1.2 C 语言的主要特点

(1) 简洁紧凑、灵活方便

C 语言共有 32 个关键字,9 种控制语句,程序书写自由,主要用小写字母表示。C 语言把高级语言的基本结构和语句与低级语言的实用性结合起来,可以像汇编语言一样直接对位、字节和地址进行操作。

(2) 运算符丰富

C 的运算符包含的范围很广泛,共有 34 种运算符。C 语言把括号、赋值、强制类型转换等都作为运算符处理,从而使 C 的运算类型极其丰富,且表达式类型多样化,灵活使用各种运算符可以实现在其他高级语言中难以实现的运算。

(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、共用体类型等,可用来实现各种复杂的数据类型的运算。并引入了指针概念,使程序效率更高。另外,C 语言还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、逻辑判断功能也很强大。

(4) C 语言是结构式语言

结构式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,可方便地调用这些函数,且具有多种循环、条件语句控制程序流向,从而使程序完全结构化。

(5) 语法限制不太严格, 程序设计自由度大

一般的高级语言其语法检查比较严格, 能够检查出几乎所有的语法错误。而 C 语言允许程序编写者有较大的自由度。

(6) 允许直接访问物理地址, 可以直接对硬件进行操作

C 语言既具有高级语言的功能, 又具有低级语言的许多功能, 能像汇编语言一样对位、字节和地址进行操作, 而这三者都是计算机最基本的工作单元, 可以用来编写系统软件。

(7) 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) 适用范围大, 可移植性好

C 语言适合于多种操作系统, 如 DOS、UNIX 等。

1.2 程序的算法及流程图

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。一个程序应包括对数据的描述和对操作的描述。对数据的描述是指在程序中要指定数据的类型和数据的组织形式, 即数据结构 (Data Structure)。对操作的描述即操作步骤, 也就是算法 (Algorithm)。

程序=数据结构+算法

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。专业的程序设计人员常被称为程序员。

程序设计的步骤如下:

① 分析问题: 对于接受的任务要进行认真的分析, 研究所给定的条件, 分析最后应达到的目标, 找出解决问题的规律, 选择解题的方法, 完成实际问题。

② 设计算法: 设计出解题的方法和具体步骤, 画出流程图。

③ 编写程序: 根据得到的算法的流程图, 用一种高级语言编写出源程序并通过测试。

④ 对源程序进行编辑、编译和连接。

⑤ 运行程序, 分析结果。运行可执行程序, 得到运行结果。能得到运行结果并不意味着程序正确, 要对结果进行分析, 看程序是否合理。若不合理就要对程序进行调试, 即通过上机发现和排除程序中的故障的过程。

⑥ 编写程序文档: 许多程序是提供给别人使用的, 就像正式的产品应当提供产品说明书一样, 提供给用户使用的程序, 必须向用户提供程序说明书。说明书的内容应包括程序名称、程序功能、运行环境、程序的安装和启动、需要输入的数据, 以及使用注意事项等。

1.2.1 算法

算法就是解决问题的步骤。算法也是程序设计的灵魂。

例如解决生活中计算长方形面积的简单问题, 需要执行以下步骤:

① 了解此长方形的长和宽两个值。

② 判断长和宽的值是否大于零。

③ 如果大于零，将长和宽两个值相乘得到面积。

④ 显示面积。

由于同一个问题可以有不同的解决方法，所以不同的两个人也可能编写出不同的算法而得到相同的结果。

在实际应用中可以用自然语言、流程图和 N-S 图等形式来描述算法。通常情况下使用流程图来描述，因为流程图是算法的一种图形化表示方式，直观、清晰，更有利于人们设计算法。算法使用一组预定义的符号来说明如何执行特定任务，这些预定义的符号已标准化，从而让全世界的开发人员都可以采用这些符号而不会引起混淆。

1. 算法的特征

一个算法应具有以下 7 个重要的特征：

① 有穷性 (Finiteness)。算法的有穷性是指算法必须能在执行有限个步骤之后终止。

② 确切性 (Definiteness)。算法的每一步骤必须有确切的定义。

③ 输入项 (Input)。一个算法有 0 个或多个输入，以刻画运算对象的初始情况，所谓 0 个输入是指算法本身给出了初始条件。

④ 输出项 (Output)。一个算法有一个或多个输出，以反映对输入数据加工后的结果。没有输出的算法是毫无意义的。

⑤ 可行性 (Effectiveness)。算法中执行的任何计算都是可以分解为基本的可执行的操作，即每个计算都可以在有限时间内完成。

⑥ 高效性 (High efficiency)。算法不仅执行速度快，且占用资源少。

⑦ 健壮性 (Robustness)。算法的健壮性是正确响应数据。

2. 算法的设计与分析

算法的设计与分析的有如下 3 种方法。

(1) 递推法

递推法是一种用若干步可重复的简运算 (规律) 来描述复杂问题的方法。

递推是序列计算中的一种常用算法。它是按照一定的规律来计算序列中的每个项，通常是通过计算机前面的一些项来得出序列中的指定项的值。其思想是把一个复杂的庞大的计算过程转化为简单过程的多次重复。该算法利用了计算机速度快和不知疲倦的特点。

(2) 递归法

程序调用自身的编程技巧称为递归 (Recursion)。一个过程或函数在其定义或说明中有直接或间接调用自身的一种方法，通常把一个大型复杂的问题层层转化为一个与原问题相似的规模较小的问题来求解，递归策略只需少量的程序就可描述出解题过程所需要的多次重复计算，大大减少了程序的代码量。递归的能力在于用有限的语句来定义对象的无限集合。一般来说，递归需要有边界条件、递归前进段和递归返回段。当边界条件不满足时，递归前进；当边界条件满足时，递归返回。

— 注意 —

① 递归就是在过程或函数中调用自身。

② 在使用递归策略时，必须有一个明确的递归结束条件，又称递归出口。

(3) 穷举法

穷举法又称暴力破解法，是一种针对于密码的破译方法，即将密码进行逐个推算直到找出真正的密码为止。例如，一个已知是4位且全部由数字组成的密码，可能共有10 000种组合，因此最多尝试10 000次就能找到正确的密码。理论上利用这种方法可以破解任何一种密码，问题只在于如何缩短试误时间。因此有些人运用计算机来增加效率，有些人辅以字典来缩小密码组合的范围。

1.2.2 流程图

20世纪60年代末，著名学者E.W.Dijkstra首先提出了“结构化程序设计”的思想。这种思想要求程序设计者按照一定的结构形式来设计和编写程序，使程序易阅读、易理解、易修改和易维护。这个结构形式主要包括两方面的内容：

① 在程序设计中，采用自顶向下、逐步细化的原则。

按照自顶向下、逐步细化的原则，应用程序设计过程应自顶向下分成若干层次，逐步加以解决：每一个层次在前一层的基础上，是对前一层设计的细化，这个过程形成一个树结构。这样，一个较复杂的大问题，就被层层分解成为多个相对独立、易于解决的小模块，不仅有利于程序设计工作的分工和组织，也使调试工作更容易进行。

② 在程序设计中，编写程序的控制结构由顺序结构、选择结构和循环结构组成。

顺序结构是一种最简单的程序结构，按从上到下的顺序执行程序。也就是说，先执行语句块A，再执行语句块B，然后执行语句块C。各语句块既可以是一条语句，也可以由多条语句构成。

选择结构又称分支结构，是一种跳转性语句的程序结构。选择结构根据给定的条件判断应该执行哪一条分支，程序判断条件为真，执行语句块A，否则执行语句块B。

循环结构是一种重复执行的结构，根据给定的条件，判断是否需重复执行循环体。循环结构有当型结构和直到型结构两种形式。当型结构在条件成立时反复执行循环体，否则退出循环结构，先判断后执行。直到型结构反复执行循环体，直到给出的条件成立时退出循环结构，先执行后判断。直到型结构的循环体至少执行一次，而当型结构的循环体可能一次都不执行。

流程图是算法的一种图形化的表示方式，目前常见的流程图有传统流程图和N-S图两种。

1. 传统流程图

用传统流程图表示程序的3种控制结构如图1.1～图1.3所示。

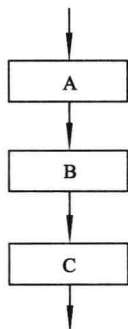


图 1.1 顺序结构

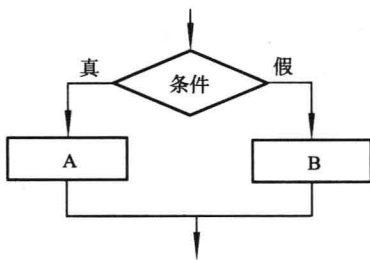


图 1.2 选择结构

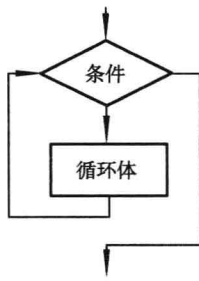


图 1.3 循环结构

2. N-S 图

N-S 图又称方框图，是在传统流程图的基础上改进的一种新的流程图。传统流程图用流程线指出各框的执行次序，对流程线的使用没有严格的控制，使用者可以不受限制地使流程随意转来转去，但是流程看起来很混乱。N-S 图则较为清晰，用 N-S 图表示程序的 3 种控制结构如图 1.4～图 1.6 所示。

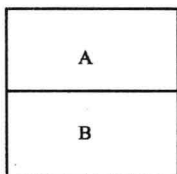


图 1.4 顺序结构

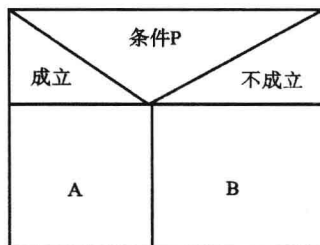


图 1.5 选择结构



图 1.6 循环结构

1.3 C 语言程序介绍

【例 1.1】输出一个字符串。

程序代码如下：

```
#include<stdio.h>
void main()
{
    printf("欢迎学习 C 语言\n");
}
```

程序运行结果如图 1.7 所示。

程序说明如下：

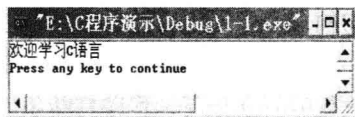


图 1.7 程序运行结果

① `#include`：以 `#` 开始的语句称为预处理语句，在编译器开始工作之前，先对这些命令进行预处理，然后将预处理的结果和源程序一起再进行通常的编译处理，以得到目标代码。并不是每个 C 程序都必须有该语句，但是，如果程序有该语句，就必须将它放在程序的开始处，位于任何其他语句之前。

② `<stdio.h>`：以 `.h` 为扩展名的文件称为头文件，它可以是 C 语言程序中现成的标准库文件，也可以是自定义的库文件。标准库文件定义了任何程序内可以使用的函数，使得开发人员可以轻松地完成日常任务。`stdio.h` 文件中包含了有关输入/输出语句的函数。

③ `void main()`：`main()` 函数是 C 语言程序处理的起点。`main()` 函数可以返回一个值，也可以不返回值。如果某个函数没有返回值，那在该函数的前面必有一个前缀 `void`。除了 `main()` 函数外，C 语言程序还可以包括一个或多个自定义函数。

④ `{`：位于函数定义的后面，表示函数的开始。大括号的后面是函数的主体或构成函数的语句。大括号也可以用于将语句块括起来。

⑤ `printf("欢迎学习 C 语言\n");`：在屏幕上产生一行输出“欢迎学习 C 语言”，并换行。函数主体中的每个语句都以分号结束。C 语言程序中的一个语句可以跨越多行，并用分号通知编译器该语句已结束。

⑥ }：位于函数定义的结尾处。

⑦ C语言源程序扩展名为.c。

C程序可以包含注释，以便对程序的某部分进行说明。注释语句不影响程序的运行，只起解释说明的作用。从书写清晰，便于阅读、理解和维护的角度出发，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字符后书写，以便看起来更加清晰，增加程序的可读性。

④ 函数与函数之间加空行，以清晰地显示程序中有几个函数。

⑤ 对于数据的输入，运行时最好出现输入提示；对于数据的输出，也要有一定的提示格式。

⑥ 为了增加程序的可读性，应对语句和函数加上适当的注释。

【例 1.2】求两数之和。

程序代码如下：

```
#include<stdio.h>
void main()      /*求两数之和*/
{
    int a,b,sum; /*声明，定义变量为整型*/
    a=123; b=456;
    sum=a+b;
    printf("sum is %d\n",sum);
}
```

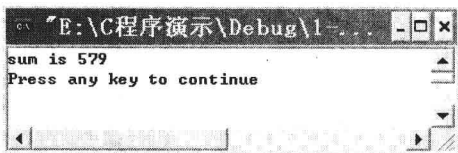


图 1.8 程序运行结果

程序运行结果如图 1.8 所示。

【例 1.3】比较两个数的大小，并把比较大的数输出。

程序代码如下：

```
#include<stdio.h>
void main()
{
    int max(int x,int y); /*对被调用函数max()的声明*/
    int a, b, c;          /*定义变量a、b、c*/
    scanf("%d,%d",&a,&b); /*输入变量a和b的值*/
    c=max(a,b);           /*调用max函数，将得到的值赋给c*/
    printf("max=%d\n",c); /*输出c的值*/
}

int max(int x,int y)
{
    int z;
    if(x>y) z=x;
    else z=y;
    return(z);
}
```

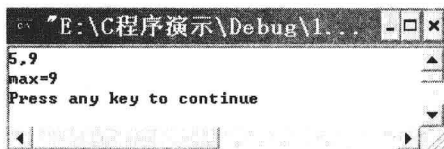


图 1.9 程序运行结果

程序运行结果如图 1.9 所示。

说 明

本例包括 main()函数和被调用函数 max()两个函数。max()函数的作用是将 x 和 y 中较大者的值赋给变量 z，return 语句将 z 的值返回给主调函数 main()。

通过几个实例，可以看出：

- ① 一个 C 语言程序可以由多个源程序组成。
- ② 一个源程序由一个主函数（main()函数）和若干个子函数构成，可以没有子函数，但主函数必须存在。例如，例 1.1 和例 1.2，并且一个源程序最多只能包含一个主函数。main()函数（主函数）是每个程序执行的起始点。
- ③ 一个函数由函数头和函数体两部分组成，函数体一般包括声明部分和执行部分。
- ④ C 程序书写格式自由，一行可以写几个语句，一个语句也可以写在多行上。在程序中允许加入一定量的空白（使用空格键或 Enter 键）。每条语句的最后必须有一个分号，分号是 C 语句的组成部分。
- ⑤ 可以在 C 程序的任何部分进行注释，注释语句不影响程序的执行结构，只是增强程序的可读性。
- ⑥ C 语言本身不提供输入/输出语句，输入/输出的操作是通过调用库函数（scanf、printf 等）来完成。程序中使用库函数时必须用#include<文件名.h>命令，并写在程序的开始处。

1.4 C 语言程序运行的步骤和方法

C 语言程序的编译系统有许多种，早期非常流行的编译系统有 Turbo C，它是 Borland 公司生产的一套 DOS 平台上的 C 语言编译系统。随着面向对象技术的飞速发展，面向对象技术的 C++、C#在 Windows 程序和大型软件开发中得到了广泛使用。

为了方便程序开发，人们开发一类称为 IDE 的软件，Visual C++ 6.0 是目前国内比较流行的一种 C++语言源程序的编译系统，使用该系统也可以编辑和运行 C 语言源程序。还可以使用 Visual C++ 6.0 创建控制台应用程序以及 Windows 应用程序。

本书采用 Visual C++ 6.0 作为 C 语言程序开发工具，所有实例程序均在 Visual C++ 6.0 中调试运行。

1.4.1 Visual C++ 6.0 环境

安装 Visual C++ 6.0 后，选择“开始”|“所有程序”| Microsoft Visual C++ 6.0 | Microsoft Visual C++ 6.0 命令，打开 Visual C++ 6.0 的操作窗口，如图 1.10 所示。

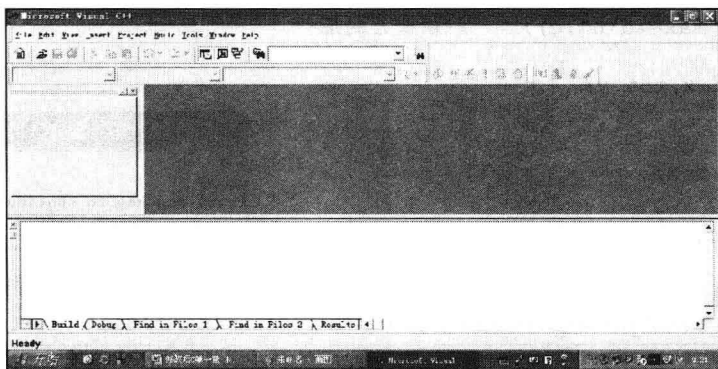


图 1.10 Visual C++ 6.0 的操作窗口

该窗口由标题栏、菜单栏、常用工具栏、向导栏、工作区和输出观察窗口组成。标题栏由应用程序的图标、应用程序的名称 3 个控制按钮构成；菜单栏由 9 个菜单构成；常用工具栏包含了常用菜单项的按钮显示；向导栏说明当前工作状态，向导栏的右侧有一个迷你调试工具栏；窗口中间部分左边是工作区，右面是文本编辑区；窗口的最下面是输出观察窗口，显示调试程序时的文本信息和出错信息。

1.4.2 编写与运行 C 语言程序

1. 输入和编辑源程序

打开 Visual C++ 编译系统后，选择 File | New 命令，在弹出的 New 对话框中选择 Files 选项卡，在列表框中选择 C++ Source File 选项，然后在 File 文本框中输入源程序的名称，例如 xx.c，如图 1.11 所示。

单击 OK 按钮，开始编辑源程序，在文本编辑器中输入源程序，如图 1.12 所示。

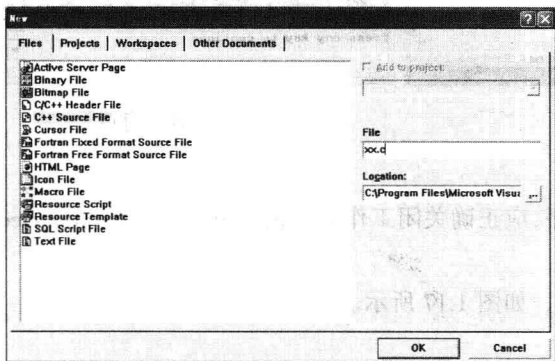


图 1.11 New 对话框



图 1.12 编辑源程序

2. 程序的编译与连接

选择 Build | Compile 命令，如图 1.13 所示，对源程序进行编译；编译完成后再选择 Build | Build 命令，执行连接命令，如图 1.14 所示。

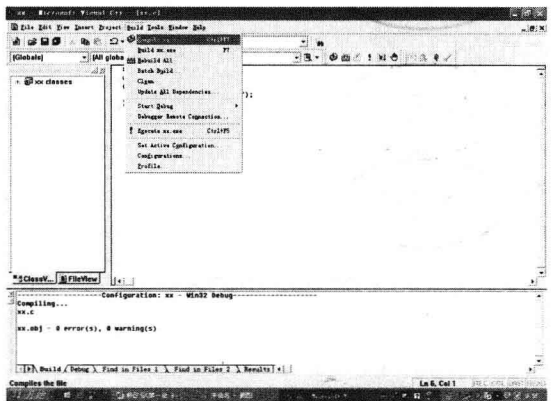


图 1.13 编译程序

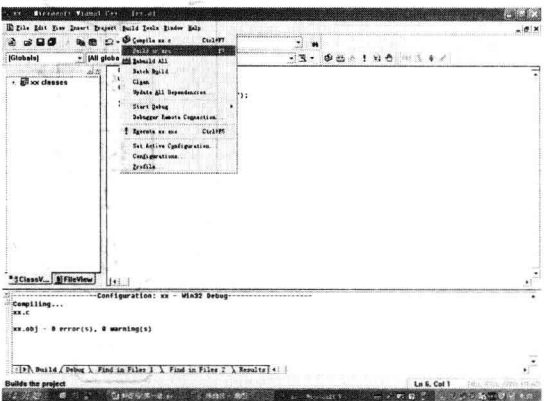


图 1.14 连接程序