

21

世纪高等职业教育
计算机技术规划教材

C语言 程序设计

费志民 主编

 人民邮电出版社
POSTS & TELECOM PRESS

21 世纪高等职业教育计算机技术规划教材

C 语言程序设计

费志民 主编

人民邮电出版社

图书在版编目 (CIP) 数据

C 语言程序设计 / 费志民主编. —北京: 人民邮电出版社, 2005.9

21 世纪高等职业教育计算机技术规划教材

ISBN 7-115-13951-2

I. C... II. 费... III. C 语言—程序设计—高等学校: 技术学校—教材 IV. TP312

中国版本图书馆 CIP 数据核字 (2005) 第 095784 号

内 容 提 要

本书内容包括 12 章和 4 个附录。每章内容前面有学习目标, 后面有习题。习题包括填空题、选择题和编程题。

针对 C 语言比较难学的情况, 编者在对全书内容的处理上尽量做到分解难点, 用通俗易懂的语言和丰富的例题解释清楚复杂的概念。本书内容丰富、条理清晰、图文并茂、实用性强, 以边讲解边举例的教学方式, 讲解 C 程序语言的基础知识、实际应用和实际操作, 程序举例丰富, 方便读者轻松学习, 快速上手。

本书是“21 世纪高等职业教育计算机技术规划教材”之一, 也可供计算机等级考试备考人员和 C 语言爱好者学习使用。

21 世纪高等职业教育计算机技术规划教材

C 语言程序设计

-
- ◆ 主 编 费志民
责任编辑 杨 堃
 - ◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街 14 号
邮编 100061 电子函件 315@ptpress.com.cn
网址 <http://www.ptpress.com.cn>
北京通州大中印刷厂印刷
新华书店总店北京发行所经销
 - ◆ 开本: 787×1092 1/16
印张: 21.5
字数: 518 千字 2005 年 9 月第 1 版
印数: 1—3 000 册 2005 年 9 月北京第 1 次印刷

ISBN 7-115-13951-2/TP · 4938

定价: 30.00 元

读者服务热线: (010)67170985 印装质量热线: (010)67129223

21 世纪高等职业教育计算机技术规划教材

编审委员会

顾 问： 聂嘉恩 段 富 梁吉业 刘本群 张存伟
谢红薇

主 任： 屠全良

副 主 任： 汤春林 巩宁平 刘国锋 杨 志 郑 静
过莉莎 赵随民 向 伟

主 编： 汤春林

编 委： 刘国锋 刘贵强 过莉莎 张建忠 杨 志
武建京 罗建斌 侯中俊 费志民 柴巧叶
郭思延

执行委员： 潘春燕 杨 堃

丛书前言

进入 21 世纪以来,伴随着高等职业教育的跨越式发展,我国高等职业教育事业得到迅速发展。经过各级教育主管部门以及高职战线广大教职员工的艰苦努力,我国的高等职业教育取得了显著成绩。随着教育的不断深入,教学质量的不断提高,一个基本适应我国社会主义现代化建设需要的高等职业教育新体系已经初步形成。

在当前的形势下,如何按照高等职业教育自身的特点和规律组织教学体系,如何使教学同企业的实际需要衔接起来,成为目前高等职业教育的重要课题。人民邮电出版社与山西省高职院校的许多专家合作,共同组建了“21 世纪高等职业教育计算机技术规划教材编审委员会”。编审委员会针对现阶段高等职业教育计算机教学的特点和要求,汲取了近年来计算机职业教学改革的成功经验,充分贯彻教育部对高职高专“以就业为导向”和“必须、够用”的理论要求。以能力为本位,从职业分析入手,对职业岗位进行能力分解,明确了计算机职业教育从业方向的核心技能需求,并以此作为对本系列教材编写的整体要求。在此基础上,在全国范围内又经过反复地征求意见及研讨和修改,规划出版了《21 世纪高等职业教育计算机技术规划教材》系列教材。

本系列教材的作者,均为高等职业院校一线的优秀骨干教师。他们不仅从事了多年的职业教学工作、具有非常丰富的教学经验,而且对学生的从业方向和岗位的实际需要有深入的了解。

本系列教材在内容上,以提高学生的实际操作技能为主干,结合目前的新技术以及企业的实际需要,从实际操作入手,详略得当、深入浅出。文字上力求表达精练,通俗易懂。每章前有“学习目标”,列出知识点和重点内容,使学习方向明确;每章后有“习题”,以便对所学知识巩固与提高;书后配有较大比例的实训,使学习与操作相结合,更方便实训课程使用。今后我们会根据具体的使用情况和读者的意见,不断修订和完善,希望能够最大限度地适合高等职业教学的要求。

本系列教材适合高等职业教育计算机类及相关专业的学生使用,也可以作为计算机领域的自学读物。

高等职业教育改革和教材建设不是一朝一夕可以完成的,作为一项工程它需要反复地研讨和实践。我们衷心希望,全国关心高等职业教育的广大读者能够对本套教材的不当之处给予批评指正、提出修改意见。我们也热切盼望从事高等职业教育的教师、专家以及信息技术方面的专家和我们联系,共同探讨实用的课程组织方案和教材编写等相关问题。来信请发至 yangkun@ptpress.com.cn,我们也殷切地期待您的投稿。

21 世纪高等职业教育计算机技术规划教材编审委员会

2005 年 3 月

前 言

C 语言是优秀的结构化程序设计语言之一，是学习程序设计的必修基础课程，虽然面向对象的程序设计在今天已经很流行，但其设计理念掌握起来仍比较困难。为解决这个矛盾，许多高职院校采取了折中的办法，即先学习 C 语言，再学习 C++ 面向对象的程序设计。

本书推荐授课课时为 70~80 学时，并提倡实践教学的课时数不少于 30 学时。本书共由 12 章组成，主要内容有：C 语言和 Turbo C 语言的概况，Turbo C 的使用，C 语言的数据类型、程序控制结构、数组应用、函数调用、文件操作以及结构体、指针、编译预处理等。建议授课教师的教学重点放在表达式、控制结构与语句、数组、函数，并反复上机训练。结构体与共用体应注重讲授指针在函数中的使用，而链表等内容可以选讲。第 12 章文件中的缓冲型文件是应重点讲授的内容，必须让学生明白，只有文件存储的数据才能永久保存。

本教材针对计算机职业教育的特点，内容尽可能简练、通俗易懂，注重强化编程基础知识与程序设计能力的训练，每章中都配备了大量的例题，每章后都有习题。程序设计课程应注重培养学生的逻辑思维能力，建议教师在讲授程序时，适当配以流程图，并辅导学生自己设计流程图，并动手编写程序，上机调试程序，改正程序中的语法错误与逻辑错误。

本书由山西财贸职业技术学院费志民老师担任主编，并负责全书的统稿。邢文德老师编写第 1、10 章，刘丽峰老师编写第 2、6、11 章，曹伟老师编写第 3、8 章，费志民老师编写第 4、5、9 章，毛俊红老师编写第 7、12 章。刘丽峰老师对部分章节进行了改写，并对全书进行了更正。同时，每位老师还编写了各章相应的实训部分。

由于编者的水平及编写时间有限，书中难免存在缺点和错误，敬请广大读者批评指正。

编者

2005 年 7 月

目 录

第 1 章 C 语言概述	1
1.1 C 语言出现的历史背景及特点.....	1
1.1.1 C 语言的发展历史.....	1
1.1.2 C 语言的特点.....	2
1.2 简单的 C 程序介绍.....	3
1.3 C 程序的上机步骤.....	6
1.3.1 进入和退出 Turbo C 2.0.....	6
1.3.2 编辑源程序文件.....	8
1.3.3 编译和运行程序.....	9
习题.....	12
第 2 章 数据类型、运算符与表达式	13
2.1 C 语言的数据类型.....	13
2.2 常量与变量.....	15
2.2.1 常量.....	15
2.2.2 变量.....	16
2.3 整型数据.....	19
2.3.1 整型常量.....	19
2.3.2 整型变量.....	20
2.4 实型数据.....	22
2.4.1 实型常量.....	23
2.4.2 实型变量.....	23
2.5 字符型数据.....	25
2.5.1 字符型常量.....	25
2.5.2 字符型变量.....	27
2.5.3 字符串常量.....	30
2.6 变量赋初值.....	31
2.7 各类数值型数据间的混合运算.....	32
2.8 算术运算符和算术表达式.....	33
2.8.1 算术运算符.....	33
2.8.2 算术表达式.....	34
2.8.3 自增、自减运算符.....	35
2.9 赋值运算符和赋值表达式.....	38
2.9.1 赋值运算符.....	38

2.9.2 赋值表达式	41
2.10 逗号运算符和逗号表达式	43
习题	44
第 3 章 顺序程序设计	47
3.1 C 语句概述	47
3.2 程序的三种基本结构	48
3.2.1 顺序结构	49
3.2.2 选择结构	49
3.2.3 循环结构	50
3.3 字符数据的输入输出	50
3.3.1 putchar 函数 (字符输出函数)	51
3.3.2 getchar 函数 (键盘输入函数)	51
3.4 格式输入与输出	53
3.4.1 printf 函数 (格式输出函数)	53
3.4.2 scanf 函数 (格式输入函数)	57
3.5 顺序结构程序设计举例	63
习题	64
第 4 章 选择结构程序设计	69
4.1 关系运算符和关系表达式	69
4.1.1 关系运算符	69
4.1.2 关系表达式	69
4.2 逻辑运算符及逻辑表达式	71
4.2.1 逻辑运算符	71
4.2.2 逻辑表达式	72
4.3 if 语句	74
4.3.1 if 语句的形式	74
4.3.2 if 语句的嵌套形式	76
4.3.3 条件运算符	81
4.4 switch 语句	82
4.5 程序举例	86
习题	89
第 5 章 循环结构程序设计	93
5.1 概述	93
5.2 goto 语句	93
5.3 while 语句	94
5.4 do-while 语句	96
5.5 for 语句	98
5.5.1 for 语句的一般形式	98
5.5.2 for 语句的几种变化形式	100

5.6	循环的嵌套	103
5.7	break 语句和 continue 语句	105
5.7.1	break 语句	105
5.7.2	continue 语句	107
5.8	程序举例	108
	习题	109
第 6 章	函数	114
6.1	概述	114
6.2	函数定义的一般形式	116
6.3	函数参数和函数的值	118
6.3.1	函数参数	118
6.3.2	函数返回值	120
6.3.3	函数的类型声明	122
6.4	函数的调用	123
6.4.1	函数调用的一般形式	123
6.4.2	函数调用的方式	125
6.5	函数的嵌套调用	125
6.6	函数的递归调用	127
6.7	局部变量和全局变量	131
6.7.1	局部变量	131
6.7.2	全局变量	132
6.8	变量的存储类别	135
6.8.1	动态存储方式和静态存储方式	135
6.8.2	自动变量	136
6.8.3	寄存器变量	136
6.8.4	外部变量	137
6.8.5	内部静态变量和外部静态变量	139
6.8.6	变量的定义和声明	142
6.8.7	存储类别小结	143
6.9	内部函数和外部函数	145
6.9.1	内部函数	145
6.9.2	外部函数	145
6.10	多文件程序的运行	146
	习题	147
第 7 章	预处理命令	151
7.1	宏定义	151
7.1.1	符号常量宏定义	151
7.1.2	带参数的宏定义	152
7.1.3	宏定义的解除	157

7.2	文件包含	157
7.3	条件编译	159
7.3.1	#if...#else...#endif 命令	159
7.3.2	#ifdef...#else...#endif 命令	160
7.3.3	#ifndef...#else...#endif 命令	160
7.3.4	条件编译的作用	161
7.4	常用库函数的应用	162
	习题	165
第 8 章	指针	166
8.1	地址和指针的概念	166
8.1.1	数据在内存中的存储	166
8.1.2	访问变量的两种方式	167
8.1.3	指针及指针变量	168
8.2	指针变量的定义和基本类型	168
8.2.1	指针变量的定义和赋值	168
8.2.2	指针变量的基本类型	170
8.3	指向变量的操作	170
8.3.1	指针变量的引用	170
8.3.2	指针运算	172
8.3.3	指针变量作为函数参数	173
8.4	函数的指针和指向函数的指针变量	175
8.4.1	用函数指针变量调用函数	175
8.4.2	用指向函数的指针作函数参数	176
8.4.3	返回指针值的函数	177
8.4.4	void 指针类型	177
	习题	178
第 9 章	数组	183
9.1	一维数组的定义和引用	183
9.1.1	一维数组的定义	183
9.1.2	一维数组的引用	184
9.1.3	一维数组的初始化	191
9.1.4	一维数组的应用	194
9.2	二维数组的定义和引用	198
9.2.1	二维数组的定义	198
9.2.2	二维数组的引用	199
9.2.3	二维数组的初始化	200
9.3	指针和数组	206
9.3.1	数组指针的定义与赋值	206
9.3.2	利用指针引用数组元素	207

9.3.3 指向多维数组的指针	208
9.4 字符串数组、字符串的指针及指向字符串的指针变量	212
9.4.1 字符数组的定义和初始化	212
9.4.2 字符串指针及指向字符串的指针变量	219
9.5 指针数组和指针的指针	220
9.5.1 指针数组	220
9.5.2 指向指针的指针	221
9.6 数组作为函数参数	223
9.6.1 用数组作为函数参数	223
9.6.2 指针数组作为 main 函数的形参	225
9.7 数组程序举例	226
习题	231
第 10 章 结构体与共用体	235
10.1 概述	235
10.2 结构体类型及结构体变量的定义	235
10.2.1 结构体类型的定义	235
10.2.2 结构体变量的定义	236
10.3 结构体变量的引用及初始化	239
10.3.1 结构体变量的引用	239
10.3.2 结构体的初始化	240
10.3.3 结构体的应用	241
10.4 结构体数组	242
10.4.1 结构体数组的定义	243
10.4.2 结构体数组的初始化	243
10.4.3 结构体数组的应用	244
10.5 指向结构体类型数据的指针	246
10.5.1 指向结构体变量的指针	246
10.5.2 指向结构体数组的指针	248
10.5.3 用结构体变量和指向结构体的指针作为函数参数	249
10.6 链表	251
10.6.1 链表概述	251
10.6.2 链表的建立	253
10.6.3 结点的删除与插入	257
10.7 共用体	260
10.7.1 共用体变量的定义	260
10.7.2 共用体变量的引用	261
10.8 枚举类型	263
10.9 用 typedef 定义类型	266
习题	268

第 11 章 位运算	271
11.1 位运算简介	271
11.1.1 按位运算符	272
11.1.2 移位运算符	275
11.1.3 位运算举例	277
11.2 位段简介	279
11.2.1 位段的概念与定义方法	279
11.2.2 位段的引用方法	281
习题	282
第 12 章 文件	284
12.1 C 文件概述	284
12.2 文件类型指针	285
12.3 文件的打开与关闭	286
12.3.1 文件的打开 (fopen 函数)	286
12.3.2 文件的关闭 (fclose 函数)	288
12.4 文件的读写	288
12.4.1 fputc (putc) 函数和 fgetc (getc) 函数	288
12.4.2 fread 函数和 fwrite 函数	291
12.4.3 fprintf 函数和 fscanf 函数	295
12.4.4 其他读写函数	296
12.5 文件定位与出错检查	296
12.5.1 文件定位	296
12.5.2 出错检测	298
12.6 文件输入输出小结	298
习题	299
附录 A C 语言中的关键字	301
附录 B 运算符的优先级和结合性	302
附录 C 库函数	303
附录 D 实训	308
参考文献	332

第 1 章

C 语言概述

学习目标

本章主要讲述 C 语言的产生过程、C 语言的结构特征、C 语言的书写格式及注释语句的使用。通过学习,要了解 C 语言的历史及 C 语言的特点,掌握 C 语言的编程格式,熟练掌握 C 语言的上机操作。

1.1 C 语言出现的历史背景及特点

1.1.1 C 语言的发展历史

任何事物的产生都有一定的历史背景,C 语言也是这样。20 世纪 60 年代,随着计算机技术的迅速发展,高级程序设计语言得到了广泛的应用。但是,仍没有一种可以用于开发操作系统等系统软件的高级语言,人们不得不用汇编语言(甚至机器语言)来开发。然而汇编语言缺点太多,给编程带来了极大的不便。为此,人们希望能够找到一种既具有一般高级语言的特点——方便易用,又具有低级语言的特性——可以直接对硬件进行操作的语言。C 语言就是在这种背景下诞生的。

C 语言是在 B 语言的基础上发展起来的,1960 年的 ALGOL 60 是它的祖先。1963 年与 1967 年,英国剑桥大学在 ALGOL 60 语言的基础上先后推出了 CPL (Combined Programming Language) 语言及 BCPL (Basic CPL) 语言。这两种语言比 ALGOL 60 更接近硬件,但规模较大,难懂,也难实现。1970 年美国贝尔实验室对 BCPL 语言做了进一步简化,设计出了 B 语言(取 BCPL 的第一个字母)。但 B 语言过于简单,只适合解决一些特殊类型的问题。于是在 1972~1973 年间,贝尔实验室的 D. M. Ritchie 在 B 语言的基础上设计出了一种既保持了 B 语言的精练特点,又提供了丰富的描述手段(数据类型和运算)的语言,取名 C 语言(取 BCPL 的第二个字母)。最初的 C 语言只是为描述和实现 UNIX 操作系统提供的一种语言工具,后来,对 C 语言进行了多次改进,它的优点引起了人们广泛的注意,渐渐风靡全球,成为国际上使用的最广泛的几种计算机语言之一。

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 语言的发展和扩充,制定了统一的标准,称为 ANSI C。1987 年,ANSI 又公布了新标准——87 ANSI C。1990 年国际标准化组织(International Standard Organization, ISO)接受 87 ANSI C 为 ISO C 的全球统一标准(ISO 9899—1990)。这是目前功能最完善、性能最优良的 C 语言新版本,目前流行的 C 语言编译系统都是以它为基础的。当前在我国微机上常用的 C 语言版本主要有:

Borland International 公司的 Turbo C (V2.0, V3.0); 微软公司的 Microsoft C (V6.0, V7.0) 等, 本教材在微机上用的编译程序是 DOS 环境下的 Turbo C2.0。

1.1.2 C 语言的特点

C 语言能够被计算机界广泛接受, 并具有很强的生命力, 正是由于它自身具备的突出特点, 归纳起来 C 语言有以下特点。

1. 语言简洁、紧凑, 使用方便、灵活

C 语言一共只有 32 个关键字 (见附录 A) 和 9 种控制语句, 程序书写自由, 压缩了一切不必要的成分, 比其他语言的源程序短, 因此输入程序时工作量少。

2. 运算符丰富

C 语言具有种类丰富的运算符, 共 34 种运算符和 15 种等级的运算优先顺序 (见附录 B)。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 语言的运算类型极其丰富, 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. 数据类型丰富, 具有现代化语言的各种数据结构

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、联合体类型及枚举类型等。可以实现各种复杂的数据结构的运算, 特别是指针类型, 使用起来更是灵活、多样。因此, C 语言具有较强的数据处理能力。

4. 模块化结构

C 语言以“函数”作为程序结构的基本单位, 这样不仅十分有利于把整体程序分割成若干个具有相对独立功能的模块, 而且便于模块间相互调用及传递数据。

5. 具有结构化的控制语句

C 语言提供了 if-else、for、while、do-while、switch-case 等结构化的控制语句, 便于采用自顶向下、逐步细化的结构化程序设计方法, 符合现代编程风格的要求。

6. 比较接近硬件与系统

C 语言能实现汇编语言的大部分功能, 如允许直接访问物理地址、能进行位运算、可直接对硬件进行操作等。

7. 生成目标代码质量高, 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. 可移植性好

与汇编语言相比, 用 C 语言编写的程序基本上不做修改就能用于各种型号的计算机和各种操作系统。

虽然 C 语言有以上优点, 但和其他程序设计语言一样, 它也有一些弱点。比如运算符的优先级较多, 有些还与常规约定不同, 不便记忆; 语法限制不太严格, 如对数组下标越界不做检查等; 这就使初学者编一个正确的 C 程序可能感到会比编一个其他高级语言程序难一些。所以, 对使用 C 语言编程者来说, 必须熟练掌握它才能灵活、正确地使用。

总之, C 语言对程序员的要求较高, 程序员使用 C 语言编写程序会感到限制少、灵活性强、功能强, 可以编写出任何类型的程序。现在, C 语言已不仅用来编写系统软件, 也用来编写应用软件。学习和使用 C 语言的人越来越多。

1.2 简单的C程序介绍

为了使大家对C程序有一个感性认识，下面先介绍几个简单的C程序。

【例 1.1】

```
1   main()
2   {
3   printf("This is a c program.\n");
4   }
```

这个程序运行结果是下面一行信息：

```
This is a c program.
```

本例中，第1行 `main()` 表示主函数。每一个C程序都必须有一个 `main()` 函数。第2行和第四行是一对花括号 `{ }`，花括号内是函数体。第3行表明本例中主函数只有一个输出语句，由C语言中的输出函数 `printf()` 构成（详见第3章）。双引号内的字符串按原样输出，“`\n`”是换行符，即在输出“`This is a c program.`”后回车换行，语句最后有一分号。表示该语句结束，这个分号必不可少。

【例 1.2】

```
1   /* 求两数之和的程序 */
2   main()                               /*主函数   */
3   {
4   int a,b,sum;                          /* 定义变量 */
5   a=135;b=246;                          /* 给变量赋值 */
6   sum=a+b;                              /* 求和     */
7   printf("sum is %d\n",sum);           /* 输出结果 */
8   }
```

本程序的作用是求两个整数 `a`、`b` 之和 `sum`，并在屏幕上输出 `sum`。

程序中，第1行和每行的右侧，以“`/*`”开头到“`*/`”结尾之间的内容表示注释，为便于理解，我们用汉字表示注释，也可以用英语或汉字拼音表示注释。注释是程序员对程序某部分的功能和作用所做的说明，是给阅读程序的人看的，对编译和运行不起作用。注释可以加在程序中的任何位置。第4行是变量说明，说明 `a`、`b` 和 `sum` 为整型 (`int`) 变量。第5行是两个赋值语句，使 `a`、`b` 的值分别为 135 和 456（C语言中，一行可以有多个语句，但要用分号分割）。第6行也是一个赋值语句，先计算 `a`、`b` 之和，再把和赋给 `sum`。第7行中的“`%d`”是输入输出的“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”。`printf()` 函数中括号内最右端 `sum` 是要输出的变量，现在它的值为 381，所以本程序执行后输出一行信息为：

```
sum is 381
```

【例 1.3】

```
1   /* 比较两个数大小的程序 */
```

```

2   main()                               /*主函数*/
3   {
4   int a,b,c;                            /*声明部分, 定义三个整型变量*/
5   scanf("%d,%d",&a,&b&);                /*输入变量 a、b 的值*/
6   c=max(a,b);                           /*调用 max 函数, a、b 为实际参数, 将得到的值赋给 c*/
7   printf("max=%d",c);                   /*输出 c 的值*/
8   }
9
9                                       /*空一行, 使程序清晰可读*/
10  int max(int x,int y) /*定义 max 函数、函数值为整型, x、y 为整型形式参数*/
11  {
12  int z;                                 /*max 函数中的声明部分, 定义本函数中的变量 z 为整型*/
13  if(x>y)z=x;                            /*将 x、y 中较大的值赋给 z*/
14  else z=y;
15  return(z);                             /*将 z 值返回, 通过 max 函数带回调用处*/
16  }

```

本程序的作用是从键盘输入两个整数, 然后在屏幕上输出它们的较大值。

该程序包括两个函数: 一个主函数 main() 和一个自定义 max()。max() 函数的作用是将 x 和 y 中较大的数赋给变量 z, 再通过 return 语句将 z 值返回给主调函数 main()。程序中第 5 行的 scanf() 是 C 语言提供的标准输入函数 (printf() 是 C 语言提供的标准输出函数), 其作用是从键盘输入 a 和 b 的值。其中, “%d,%d” 用于指定输入的两个数据按十进制整数形式输入, &a 和 &b 中的 “&” 表示 “取地址”, 即将两个值分别输入到变量 a 和 b 的地址所标志的单元中, 这种形式是与其他语言不同的, 关于 scanf() 详见第 3 章。

程序第 6 行为调用 max() 函数, 在调用过程中将实际参数 a 和 b 的值分别传送给第 10 行的 max() 函数中的形式参数 x 和 y。经过执行 max() 函数得到一个返回值 (即 max() 函数中的变量 z 的值), 并把这个值赋给 c, 然后输出 c 的值。程序运行情况如下:

```

3, 5 ✓ (✓ 表示回车)    (输入 3 和 5 给 a 和 b)
max=5                  (输出 c 的值)

```

从以上 3 个具体 C 语言程序的例题, 我们可以发现一个完整的 C 源程序应符合以下几点:

(1) C 程序是由函数构成的, 这种特点有利于实现程序的结构化 (结构清晰)。一个 C 语言源程序有一个 main() 函数, 也可以包含一个 main() 函数和若干个其他函数。因此函数是 C 源程序的基本单位。C 语言程序中有 3 种类型的函数:

① main() 函数, 这是系统提供的特殊函数, 称为主函数, 每一个 C 程序都必须且只能包含一个 main() 函数。

② 库函数, 如 printf()、scanf() 等, 这是系统提供的函数, 用户使用时可直接调用。Turbo C 系统提供了 300 多个这类函数。善于使用库函数可以减少编程量, 提高编程效率。

③ 编程人员自己设计的函数, 如例 1.3 中的 max() 函数, 这是用户为解决某一类具体问题而构造的函数。

(2) 一个函数由两部分构成。

① 函数首部，即函数的第一行。包括函数名、函数类型、函数属性、函数参数（形参）名、参数类型。函数首部后面没有分号。

例 1.3 中的 max()函数的首部为：

```

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

```

一个函数名后必须跟一对圆括号，函数参数可以没有，如 main()。

② 函数体，即函数首部下面的大括弧{——}内的部分。如果一个函数内有多个大括弧，则最外的一对{——}内为函数体的范围。

函数体一般包括：

- 声明部分：在这部分中定义所用到的变量，对要调用的函数进行声明。
- 执行部分：由若干个语句组成。

当然，在某些情况下可以没有声明部分（如例 1.1）甚至可以既无声明部分，也无执行部分。例如：

```

dump()
{

```

它是一个空函数，什么也不干，但这是合法的。

(3) 一个 C 程序总是从 main()函数开始执行的，而不论 main()函数在整个程序中的位置如何。执行完 main()中的所有语句后，程序就结束。

(4) 每个语句和变量定义的最后必须有一个分号，分号是 C 程序的必要组成部分，如“c=a+b;”。

分号不可少，即使是程序的最后一个语句也应包含分号。

注意：函数首部后、预处理命令、大括号{}之后不能加分号。

(5) C 程序书写格式自由，一行内可以写几个语句，一个语句可以分写在多行。C 程序没有行号。

(6) C 语言本身没有提供输入输出语句，输入输出操作是通过库函数 scanf()和 printf()来实现的。

(7) C 语言用“/*……*/”对程序进行注释。一个好的、有价值的 C 源程序都应当加上必要的注释，以增加程序的可读性。

为了写出一个层次清楚、可读性强的 C 程序，我们建议在写程序时遵循如下 C 程序风格：

- ① C 程序一般使用小写英文字母书写，大写字母一般用作符号常量或其他特殊用途。
- ② 锯齿形。虽然 C 程序书写自由，但为了使程序层次清楚，可把程序写成锯齿形，即在语句之前加上适当数量的空格，使处于同一层次的语句从同一列开始。
- ③ 尽管 C 程序一行内可以写几个语句，但建议采用一行一句的风格。
- ④ 标识符的命名尽量“见其名而知其意”。
- ⑤ 适当地使用注释。注释分为序言性注释和功能性注释。序言性注释一般写在函数定义之外，用以对整个函数加以说明，帮助读者理解整个函数用途。功能性注释用以说明某一块程序或某几个语句，以帮助读者理解这些程序块或语句。注释在编译时被忽略掉，故注释不