


011-01 01  
01 010  
0  
100  
100  
1011  
110  
100

国家示范性高职高专规划教材 · 计算机系列



# C语言程序设计 项目化教程

0101100  
0 10010 101101000  
0101101010 1 01 10  
00010  
0 10  
01 01  
01 010 00  
010 0 010

王宝库 主编

- 把握最新技术发展方向
- 突出学生实践能力培养
- 基于工作过程项目驱动
- 配电子教案、习题解答



清华大学出版社  
<http://www.tup.tsinghua.edu.cn>



北京交通大学出版社  
<http://press.bjtu.edu.cn>

国家示范性高职高专规划教材·计算机系列

# C 语言程序设计项目化教程

王宝库 主编  
王趾成 温丹丽 参编  
张艳丽 陈丹



清华大学出版社  
北京交通大学出版社

·北京·

## 内 容 简 介

C 语言程序设计项目化教程包含实用教程、习题与解答、上机实训三部分。实用教程部分内容包括 C 语言概述、C 数据类型、C 语言基本运算、顺序程序设计、分支程序设计、循环控制、数组、函数及存储类别、编译预处理、指针结构体、共用体及枚举类型、文件等；习题与解答部分给出了每个项目课后习题与参考答案；上机实训部分按内容给出了相关的实训操作。

本书可作为高等学校计算机相关专业“C 语言程序设计”课程教学的教材，也可作为非计算机专业 C 语言程序设计的参考用书。

本书封面贴有清华大学出版社防伪标签，无标签者不得销售。

版权所有，侵权必究。侵权举报电话：010-62782989 13501256678 13801310933

### 图书在版编目 (CIP) 数据

C 语言程序设计项目化教程 / 王宝库主编. — 北京：清华大学出版社；北京交通大学出版社，2012. 2

(国家示范性高职高专规划教材·计算机系列)

ISBN 978-7-5121-0872-1

I. ① C… II. ① 王… III. ① C 语言-程序设计-高等职业教育-教材  
IV. ① TP312

中国版本图书馆 CIP 数据核字 (2012) 第 002600 号

责任编辑：韩素华

出版发行：清华大学出版社 邮编：100084 电话：010-62776969

北京交通大学出版社 邮编：100044 电话：010-51686414

印刷者：北京瑞达方舟印务有限公司

经 销：全国新华书店

开 本：185×260 印张：18.25 字数：449 千字

版 次：2012 年 2 月第 1 版 2012 年 2 月第 1 次印刷

书 号：ISBN 978-7-5121-0872-1/TP·679

印 数：1～4 000 册 定价：29.00 元

---

本书如有质量问题，请向北京交通大学出版社质监局反映。对您的意见和批评，我们表示欢迎和感谢。  
投诉电话：010-51686043, 51686008；传真：010-62225406；E-mail: press@bjtu.edu.cn。

# 前 言

C 语言是一种通用的现代计算机程序设计语言，它是根据结构化程序设计原则设计并实现的，C 语言不仅适合于系统程序设计，也适用于应用程序设计，是一种较理想的程序设计语言。本教材既可以作为大专院校计算机专业及非计算机专业 C 语言教学教材，也可以作为自学教材。

本教材包含三部分。第一部分为实用教程，共 12 个项目：项目 1 为 C 语言概述；项目 2 为数据基本类型；项目 3 为 C 语言的基本运算；项目 4 为顺序程序设计；项目 5 为分支程序设计；项目 6 为循环控制；项目 7 为数组；项目 8 为函数及存储类别；项目 9 为编译预处理；项目 10 为指针；项目 11 为结构体、共用体及枚举类型；项目 12 为文件。第二部分为习题与解答，这部分根据课程内容安排了大量的较实用的习题，并给出了参考答案。第三部分为上机实训，安排了与讲课内容有关的实训操作。

本教材参考学时为 70 学时，各院校可根据各自的教学安排对本教材内容作适当的取舍。

本书由王宝库担任主编并负责统稿，具体编写人员分工为：辽宁工程技术大学职业技术学院王宝库编写项目 1~5、辽宁工程技术大学职业技术学院陈丹编写项目 6~8，沈阳师范大学职业技术学院温丹丽编写项目 9~10，石家庄职业技术学院王趾成编写项目 11~12；习题与解答由相应老师编写；辽宁工程技术大学职业技术学院张艳丽编写上机实训。

由于水平有限，加之时间仓促，错误和不足之处在所难免，敬请授课教师和广大读者批评指正。

编 者  
2011 年 11 月

# 目 录

## 第1部分 实用教程

项目1 C语言概述	3
1.1 C语言的发展和特点	3
1.1.1 C语言的发展	3
1.1.2 C语言的特点	4
1.2 C程序基本结构	5
1.3 C程序运行的上机步骤	7
项目2 数据基本类型	11
2.1 标识符、常量和变量	11
2.1.1 标识符	11
2.1.2 常量	11
2.1.3 变量	12
2.2 整型数据类型	12
2.2.1 整型常量	12
2.2.2 整型变量	13
2.3 实型数据类型(浮点型数据)	14
2.3.1 实型常量	14
2.3.2 实型变量	14
2.4 字符型数据类型	15
2.4.1 字符型常量	15
2.4.2 字符型变量	16
2.4.3 字符串常量	17
2.5 新类型定义	18
项目3 C语言的基本运算	19
3.1 算术运算	19
3.1.1 算术运算符	19
3.1.2 算术表达式	19
3.2 赋值运算	20
3.2.1 赋值运算符	20
3.2.2 赋值表达式	21
3.3 自增/自减运算和逗号运算	22
3.3.1 自增/自减运算符及其表达式	22

3.3.2	逗号运算符及其表达式	23
3.4	关系运算和逻辑运算	24
3.4.1	关系运算符及其表达式	24
3.4.2	逻辑运算符及其表达式	24
3.5	位运算	25
3.5.1	位运算符	25
3.5.2	位运算表达式	26
3.5.3	逻辑运算符与位运算符的区别	28
3.6	数据类型转换	29
3.6.1	自动转换	29
3.6.2	强制转换	30
<b>项目 4</b>	<b>顺序程序设计</b>	<b>32</b>
4.1	C 语言的基本语句	32
4.2	数据输出	34
4.2.1	格式输出函数 (printf()函数)	34
4.2.2	字符输出函数 (putchar()函数)	39
4.3	数据输入	40
4.3.1	格式输入函数 (scanf()函数)	40
4.3.2	字符输入函数 (getchar()函数)	43
<b>项目 5</b>	<b>分支程序设计</b>	<b>44</b>
5.1	if 语句与条件运算符	44
5.1.1	if 语句的三种形式	44
5.1.2	条件运算符	46
5.2	if 语句的嵌套	47
5.3	switch 语句	48
<b>项目 6</b>	<b>循环控制</b>	<b>51</b>
6.1	goto 语句及用 goto 语句构成循环	51
6.2	基本循环程序设计	52
6.2.1	while 语句	52
6.2.2	do-while 语句	53
6.2.3	for 语句	55
6.3	循环的嵌套	60
6.4	break 语句和 continue 语句	61
6.4.1	break 语句	61
6.4.2	continue 语句	62
<b>项目 7</b>	<b>数组</b>	<b>64</b>
7.1	一维数组	64
7.1.1	一维数组的定义和初始化	64
7.1.2	一维数组元素的引用	65

7.2	二维数组	67
7.2.1	二维数组的定义和初始化	67
7.2.2	二维数组元素的引用	69
7.3	字符数组	72
7.3.1	字符数组的定义和初始化	72
7.3.2	字符串及其结束标志	73
7.3.3	字符数组的输入输出	74
7.3.4	字符串处理函数	75
<b>项目 8</b>	<b>函数及存储类别</b>	<b>80</b>
8.1	函数的定义与调用	80
8.1.1	函数概述	80
8.1.2	函数定义的一般形式	81
8.1.3	函数调用的一般形式	83
8.1.4	函数说明	84
8.2	函数参数及函数返回值	85
8.2.1	函数参数	85
8.2.2	函数的返回值	87
8.3	函数调用的进一步讨论	88
8.3.1	函数的嵌套调用	88
8.3.2	函数的递归调用	89
8.4	存储类别	93
8.4.1	作用域和生存期	93
8.4.2	局部变量和全局变量	94
8.4.3	变量的存储类别	96
8.4.4	函数的存储类别	99
<b>项目 9</b>	<b>编译预处理</b>	<b>101</b>
9.1	宏定义	101
9.1.1	不带参数的宏定义	101
9.1.2	带参数的宏定义	102
9.1.3	终止宏定义	103
9.2	文件包含	105
<b>项目 10</b>	<b>指针</b>	<b>107</b>
10.1	指针的概念	107
10.1.1	地址和指针	107
10.1.2	指针变量的定义	108
10.2	指针的操作	109
10.2.1	指针变量的赋值及引用	109
10.2.2	指针的运算	111
10.3	指针与数组	112

10.3.1	指针与数组概述 .....	113
10.3.2	一维数组和指针 .....	113
10.3.3	二维数组和指针 .....	115
10.4	指针与函数 .....	118
10.4.1	指针作为函数的参数 .....	118
10.4.2	返回指针的函数和指向函数的指针变量 .....	122
项目 11	结构体、共用体及枚举类型 .....	126
11.1	结构体 .....	126
11.1.1	结构体类型及变量的定义 .....	126
11.1.2	结构体变量的初始化及引用 .....	128
11.1.3	链表处理 .....	136
11.2	共用体 .....	142
11.2.1	共用体类型及变量的定义 .....	143
11.2.2	共用体变量的初始化及引用 .....	144
11.3	枚举类型 .....	145
项目 12	文件 .....	149
12.1	文件概述 .....	149
12.2	文件的打开与关闭 .....	150
12.2.1	文件类型指针 .....	150
12.2.2	文件的打开 (fopen 函数) .....	151
12.2.3	文件关闭 (fclose 函数) .....	153
12.3	文件的读写操作 .....	153
12.3.1	fputc 函数和 fgetc 函数 (putc 函数和 getc 函数) .....	153
12.3.2	fread 函数和 fwrite 函数 .....	157
12.3.3	fprintf 函数和 fscanf 函数 .....	160
12.3.4	其他读写函数 .....	160
12.4	文件的定位 .....	162
12.4.1	rewind 函数 .....	162
12.4.2	fseek 函数和随机读写 .....	162
12.4.3	ftell 函数 .....	164
12.5	文件的出错检测 .....	164
12.5.1	ferror 函数 .....	164
12.5.2	clearerr 函数 .....	164

## 第 2 部分 习题与解答

习题 1 及参考答案 .....	167
习题 2 及参考答案 .....	169
习题 3 及参考答案 .....	172



习题 4 及参考答案	175
习题 5 及参考答案	180
习题 6 及参考答案	187
习题 7 及参考答案	197
习题 8 及参考答案	205
习题 9 及参考答案	211
习题 10 及参考答案	217
习题 11 及参考答案	227
习题 12 及参考答案	232

### 第 3 部分 上机实训

实训 1	247
实训 2	249
实训 3	251
实训 4	251
实训 5	253
实训 6	255
实训 7	257
实训 8	258
实训 9	260
实训 10	262
实训 11	263
实训 12	265

### 附 录

附录 A 标准 ASCII 字符集	271
附录 B C 语言关键字	271
附录 C C 运算符的优先级与结合性	272
附录 D Turbo C 2.0 常用库函数	273
附录 E Turbo C 2.0 上机过程	276
附录 F Turbo C 2.0 常见编译错误信息	279

# 第1部分 实用教程



# 项目 1 C 语言概述

C 语言是目前世界上最流行的通用的计算机高级程序设计语言之一,它不仅能在 UNIX 系统下运行,还能在许多操作系统(如 PC-DOS, CP/M, VMS 等)支持下的各种 16 位和 32 位计算机上运行。

C 语言是介于低级语言(如汇编语言)与高级语言(如 COBOL 语言)之间的一种语言,有人称之为中级语言,它设计精巧,功能齐全,既适合于编写应用软件,又特别适合于编写系统软件,故又称之为系统程序设计语言。另外,它也完全适合编写不同领域中的应用程序,可以充当一些程序设计的宿主语言(如某些数据库语言的宿主语言等)。

目前,人们借助于 C 语言已经开发出了各种各样的系统程序和应用程序,如图形处理、通信、工程、数据库、CAD/CAM 及字处理等各领域多方面的多种软件。

## 1.1 C 语言的发展和特点

### 1.1.1 C 语言的发展

C 语言的发展可以追溯到 ALGOL 60。ALGOL 60 是 1960 年由国际计算机委员会设计的一种面向过程的高级语言。它是一种结构化语言,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,不宜于来编写系统程序。系统程序主要采用汇编语言编写,而汇编语言是面向机器的,用它编写的程序可读性和可移植性都比较差。因此,人们开始考虑设计一种集高级语言和低级语言特点于一身的语言,C 语言就是在这种情况下应运而生的,它符合人们的这种期望,集中了高级语言和低级语言的优点。

1963 年,英国剑桥大学和伦敦大学首先将 ALGOL 60 发展成 CPL(Combined Programming Language)语言,该语言已比较接近硬件,但规模较大,难以实用。

1967 年,剑桥大学的 Martin Richards 将 CPL 改写成 BCPL(Basic Combined Programming Language)语言。BCPL 将 CPL 大大简化,使其既具有结构化程序设计语言的特点,也能直接处理与硬件相关的一些数据。

1970 年,美国贝尔实验室的 Ken Thompson 将 BCPL 改写成 B 语言,并用 B 语言描述和开发了 UNIX 操作系统(用高级语言编写的第一个操作系统),并在 DEC 公司的 PDP-7 小型机上实现。

1972 年, Ken Thompson 在 UNIX 系统上的合作者 Dennis M. Ritchie 又将 B 语言修改设计成 C 语言,C 语言保持了 BCPL 和 B 语言的精练和接近硬件的特点,同时也克服了它们的语法过于简单、数据无类型等缺点。

1973 年, Ken Thompson 和 Dennis M. Ritchie 合作将 1969 年编写的 UNIX 操作系统用 C 语言重新进行了编写,即 UNIX 第 5 版,其中 C 语言代码占 90%以上,只保留了少量汇编语言代码,这样使 UNIX 操作系统向其他类型的机器上移植变得相当简单,此时 C 语言

开始受到人们的关注。

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《The C Programming Language》一书。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,被称为 K&R C 语言。其后的十几年中,适用于不同机种和不同操作系统的 C 编译系统相继问世,从而把 C 语言的应用推向了更加广泛普及的阶段。然而,尽管 C 语言源程序兼容性很强,但由于没有统一的标准,各种版本之间略有差异。为此,美国国家标准局(ANSI)于 1983 年制定了 C 语言标准,并于 1987 年开始实施,称为 ANSI C。

1988 年,K&R 按 ANSI C 标准重编写了《The C Programming Language》一书。此后陆续出现的各种 C 语言编译版本,如 Microsoft C、Turbo C、Quick C 等都是与 ANSI C 兼容的新版本。它们的语法和语句功能是一致的,其差异表现在各自的标准函数库中收纳的函数种类、格式和功能上。

自 C 语言开发成功以来,使利用高级语言编写系统软件成为可能。目前,C 语言已成为流行的通用的程序设计语言之一。

### 1.1.2 C 语言的特点

由于 C 语言集中了一般高级语言和汇编语言的优点,既能用它方便地编写不依赖于计算机硬件设施的各种应用程序,又能用它编写包括操作系统在内的各种系统程序。C 语言具有多方面的特点,可大致归纳如下。

(1) 语言表达能力强。C 语言是一种通用的程序设计语言,不局限于某个机器或某个操作系统。C 语言具有丰富的运算符,除了具有一般高级语言所能处理的算术运算和逻辑运算外,还可以直接进行通常由硬件实现的对位、字节、地址及寄存器等的操作,因此在系统程序的设计中很有效,有“高级汇编语言”之称,以致足以取代汇编语言来编制各种系统软件和应用软件。

(2) 语言简洁。用 C 语言编写的程序通常比用其他高级语言编写的程序更简练,代码行少。

C 语言中类型说明符采用缩写形式,如整型可用 `int`,而不用 `integer`;字符型用 `char` 而不用 `character` 等。

C 语言中关键字较少,只有 32 个标准的关键字、44 个标准运算符及 9 种控制语句,而有些关键字又可用简单的符号代替,例如,循环语句中的循环体可以采用花括号“`{ }`”作为定界符。

C 语言没有专门的输入/输出语句,输入/输出操作通过函数调用完成,它的许多功能都可以通过函数调用来完成,因此,语言的组成精练、简洁,使用方便、灵活。

另外,C 语言程序在运行时所需要的支持少,占用的存储空间也小。

(3) 数据类型丰富。C 语言提供了丰富的数据类型,除了基本的数据类型(如整型、实型、字符型等)之外,还提供了构造的数据类型(如数组、结构体、共用体和枚举等)。这些构造数据类型是在基本类型基础上按结构化的层次构造方法构造出来的,使用这些构造数据类型可以很方便地实现各种复杂的数据结构(如链表、栈、树、图等)的操作。

(4) 代码执行效率高。许多高级语言相对汇编语言而言其代码的执行效率要低得多,但 C 语言则不然,用 C 语言描述一个问题,其代码执行效率比用汇编语言描述同一个问题

只低约 10%~20%左右。而且用 C 语言编程要比用汇编语言编程快得多。

(5) 程序的可移植性好。程序的可移植性是指在一个环境上运行的程序可以不加或稍加改动后在另一个完全不同的环境上运行。汇编语言是依赖于机器硬件的,用汇编语言编写的程序一般不可移植,而有些高级语言,因它们的编译程序不可移植,进而影响了用它们编写的程序的可移植性。而 C 语言编译程序可移植,因此用 C 语言编写的程序也可移植。

(6) C 语言是一种结构化的程序设计语言。C 语言具有顺序、选择、循环三种基本控制结构,它的各种控制语句如 if、while、for、switch、break 等,功能灵活,足以描述结构良好的程序。C 语言的代码和数据分隔特性(使用全局变量和局部变量)、代码块分隔特性(将一组语句放在一对花括号中)在功能上有助于数据隐蔽;C 函数的独立性有助于进行模块化程序设计。

(7) 绘图功能强。C 语言具有庞大的图形函数库,利用 C 提供的丰富的图形函数可进行二维、三维图形及动画设计和漂亮的屏幕界面设计。

虽然 C 语言有许多优点,但也存在一些不足之处。

(1) 用 C 语言编程,自由度大。如对变量的类型约束不够严格,整型、字符型及逻辑型数据的通用,指针和数组的通用等,使用户不易掌握。

(2) C 语言的运算符优先级太多,不便于记忆,有些还与常规约定有所不同。

(3) 检测手段较弱。如 C 语言不检查数组下标越界和变量类型的可兼容性等。

(4) 数据类型转换比较随便,进而影响了程序的安全性。

这些都要求程序员对程序设计更熟练、查错能力更强。但是,瑕不掩瑜,多种程序设计的应用已经证明 C 语言是一种极为有效且表达能力很强的语言。

总之,由于 C 语言的上述特点,使得 C 语言越来越受到程序设计人员的重视,并且已经在广泛的领域得到了应用。

## 1.2 C 程序基本结构

C 程序的基本结构是函数,一个 C 程序是由一个或多个 C 函数组成的,C 函数的实质是实现一个特定功能的程序段,一个 C 函数一般由若干条 C 语句组成。C 语句是完成某种程序功能的最小单位。

**例 1.1** C 程序实例 1——输出一行文字。

```
main( )                               /* 主函数 */
{
    printf("This is a C program. \n"); /* 输出函数调用 */
}
```

运行结果如下:

```
This is a C program.
```

本程序的作用是输出一行信息,其中 main 表示“主函数”,每一个 C 程序都必须有一个 main 函数。函数体由花括号“{ }”括起来。本例中主函数内只有一个输出语句,printf 是 C 语言中的输出函数,双引号内的字符串原样输出,“\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。

/\* ... \*/表示注释部分,用于解释该程序或该语句的作用。注释只是给用户看的,目的

是为了使程序更易于理解，对系统编译和运行不起任何作用。因此注释可以出现在任何空格、制表符或换行符可以出现的地方，其内容可以任意。

### 例 1.2 C 程序实例 2——计算两个整数之和。

```
#include "stdio.h"          /* 命令行，指明本程序包含 stdio.h 头文件 */
main( )
{ int a,b,s;                /* 声明 a, b, s 三个整型变量 */
  a=2;   b=3;               /* 给 a, b 赋值 */
  s=a+b;                    /* 计算 a+b 的和，并赋给变量 s */
  printf("a=%d, b=%d, s=%d \n",a,b,s); /* 输出 a, b 及 s 值 */
}
```

运行结果如下：

```
a=2, b=3, s=5
```

本程序的作用是求两个整数  $a$  和  $b$  之和。第 3 行是声明部分，定义了变量  $a$ 、 $b$  和  $s$ ，且其类型均为整型 ( $\text{int}$ )，第 4 行是两个赋值语句，使变量  $a$  和  $b$  的值分别为 2 和 3，第 5 行使  $s$  的值为  $a+b$  之和，第 6 行是输出函数调用语句，用来输出  $a$ 、 $b$  和  $s$  的值。其中，“ $a=\%d, b=\%d, s=\%d \backslash n$ ”是输出的“格式控制字符串”，“ $\%d$ ”用来指定输入输出时的数据类型及格式（详见项目 4），表示“十进制整数类型”，在执行输出时，此位置上代以一个十进制整数值，其他字符原样输出， $\text{printf}$  函数中括号内最右端  $a$ 、 $b$  和  $s$  是要输出的变量，因此有以上运行结果。

### 例 1.3 C 程序实例 3——求两个数中的最大值。

```
#include "stdio.h"
main( )
{ int a,b,ma;                /* 定义变量 a 和 b */
  scanf("%d,%d",&a,&b);      /* 从键盘输入 a 和 b 的值 */
  ma=max(a,b);
/* 调用 max 函数，并将 a 和 b 的值传给 x 和 y，将得到的函数结果赋给 ma 变量 */
  printf("max=%d\n",ma);    /* 输出 ma 的值 */
}
int max(int x, int y)        /* 函数首部 */
/* 定义 max 函数，函数值为 int 型，两个形式参数 x, y 均为 int 型 */
{ int m;                    /* 定义 max 函数中的变量 m */
  if(x>y) m=x;              /* 条件判断语句，如果 x>y 成立，则将 x 的值赋给变量 m */
  else m=y;                 /* 如果 x>y 不成立，则将 y 的值赋给变量 m */
  return m;                 /* 将 m 值从 max 函数带回到主函数 */
}
```

运行结果如下：

```
8, 5 ✓ (输入 8 和 5 给 a 和 b)
max=8
```

本程序包括两个函数：主函数  $\text{main}$  和被调用的函数  $\text{max}$ 。 $\text{max}$  函数的作用是将变量  $x$  和  $y$  中较大者的值赋给变量  $m$ ，然后由  $\text{return}$  语句将  $m$  的值返回给主调函数  $\text{main}$ 。返回值是通过函数名  $\text{max}$  带回到  $\text{main}$  函数的调用处。

$\text{main}$  函数中的  $\text{scanf}$  是“输入函数”，其作用是输入变量  $a$  和  $b$  的值。 $\&a$  和  $\&b$  中的“ $\&$ ”的含义是“取地址”，即将输入的两个数值分别存放到变量  $a$  和  $b$  的地址所标志的单元中，也就是输入给变量  $a$  和  $b$ （详见项目 10）。

main 函数中第 4 行为调用 max 函数，在调用时将实际参数 a 和 b 的值分别传送给 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 函数中由 return 语句返回的变量 m 的值），赋给变量 ma（关于函数参数详见项目 8）。

C 程序的基本构成如下。

(1) C 程序是由函数构成的。一个 C 源程序至少包含一个 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 程序的基本单位。被调用的函数可以是系统提供的库函数（如 scanf 和 printf 函数），也可以是用户根据需要自己编写的函数（如例 1.3 中的 max 函数）。

(2) 一个函数由首部和函数体两部分组成，具体如下。

① 函数的首部，即函数的第一行。包括函数名、函数类型、函数参数（形参）名、参数类型等。

例如，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

一个函数名后面必须跟一对圆括号，函数参数可以没有，如 main()，但圆括号不能省略。

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“{ }”为函数体的范围。

函数体一般包括以下两部分：声明部分和执行部分。

声明部分用于定义本函数中将要用到的变量，如例 1.2 中 main 函数中的“int a,b,s;”语句，例 1.3 中 main 函数中的“int a,b,m;”语句等，在某些情况下可以没有声明部分，如例 1.1。

执行部分是由若干条语句组成的，用以实现该函数的功能，如例 1.1 中的“printf(“This is a C program. \n”);”语句等。

(3) C 程序语句末尾都必须有一个分号，分号是 C 语句的必要组成部分，不可缺少。

如：

```
s=a+b;
```

分号不可少。即使是程序中最后一个语句也应包含分号。

(4) 一个 C 程序总是从 main 函数开始执行，再由 main 函数结束，而不论 main 函数处在程序的何处。

(5) C 程序书写格式自由，一行内可以写几个语句，一个语句也可以分写在多行上。

(6) C 语言本身没有输入输出语句。输入和输出的操作是由 C 提供的库函数完成的。

(7) 可以用 /\*...\*/（注意 / 与 \* 之间不能有空格）对 C 程序中的任何部分作注释。一个好的，有使用价值的源程序都应当加上必要的注释，以增加程序的可读性。

## 1.3 C 程序运行的上机步骤

所谓程序，就是一组计算机能识别和执行的指令。每一条指令使计算机执行特定的操作，程序可以用高级语言编写，用高级语言编写的程序称为“源程序”。从根本上说，计算机只能识别和执行由 0 和 1 组成的二进制的指令，而不能识别和执行用高级语言编写的指



令。为了使计算机能执行高级语言源程序，必须先用一种称为“编译程序”的软件，把源程序翻译成二进制形式的“目标程序”。然后将该目标程序与系统的函数库和其他目标程序连接起来，形成可执行的目标程序。

现在，实用的 C 语言编译系统种类繁多，适用于 IBM PC 机运行的有 Microsoft C(MSC) 和 Turbo C (TC) 等。

Turbo C 是美国 Borland 公司的产品，Borland 公司是一家专门从事软件开发、研制的公司。该公司相继推出了一套 Turbo 系列软件，如 Turbo BASIC, Turbo Pascal, Turbo C 等，这些软件很受用户欢迎。该公司于 1987 年首次推出 Turbo C1.0 产品，其中使用了全然一新的集成开发环境 (IDE)，使用了一系列下拉式菜单，将文本编辑、程序编译、连接及程序运行一体化，大大方便了程序的开发。1988 年推出 Turbo C1.5 版，增加了图形库和文本窗口函数库等，1989 年推出了 Turbo C2.0 版，该版本在原来集成开发环境基础上增加了查错功能等。

现以 Turbo C 2.0 为例讲述上机步骤。

首先将 Turbo C 2.0 系统装入某一磁盘某一目录下，例如，放在 C:\TC 子目录中。

### 1. 启动 Turbo C 2.0

如果用户的当前目录是 Turbo C 编译程序所在的子目录（如 TC 子目录），只需从键盘输入“tc”命令即可，如

C: \TC> tc ✓ (带下划线部分为键盘输入的内容，以下同)

屏幕上显示出 Turbo C 2.0 的集成开发环境，如图 1-1 所示。其中最上一行为 Turbo C 2.0 主菜单，中间窗口为编辑区，接下来是信息窗口，最下面一行为参考行。这 4 个部分构成了 Turbo C 2.0 的主屏幕。



图 1-1 Turbo C 2.0 的集成开发环境

“主菜单”中包括 8 个菜单项，分别是 File、Edit、Run、Comile、Project、Options、Debug、Break/Watch。

用户可以通过这 8 个菜单项来选择使用 Turbo C 2.0 集成环境所提供的各项主要功能。这 8 个菜单项分别代表文件、编辑、运行、编译、项目、选项、调试、中断/观察等功能。

按 F10 键可以激活主菜单，也可以同时按下 Alt 键和菜单项的首字母键来激活主菜单的某一菜单项，然后就可采用键盘上的“→”和“←”键来选择主菜单中所需要的菜单项，