



高职高专计算机系列规划教材

# C语言程序设计

胡 钢 主编      何华夏 夏 容 副主编



中国铁道出版社  
CHINA RAILWAY PUBLISHING HOUSE



清华大学出版社数字出版工程

# C语言程序设计

第2版

清华大学出版社



清华大学出版社

21 世纪高职高专计算机系列规划教材

# C 语言程序设计

胡 钢 主 编  
何华夏 夏 容 副主编

**中国铁道出版社**  
CHINA RAILWAY PUBLISHING HOUSE

---

## 内 容 简 介

本书是 21 世纪高职高专计算机系列规划教材之一,对知识点的讲解由浅入深,强调算法设计,突出编程思路,注重实例讲解和对动手能力的培养。

全书共分 11 章,内容主要包括:C 语言概述,算法,C 语言基础,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,结构体、枚举类型与共用体,文件。本书在每一章前面列出了本章的主要内容和学习重点,每一节后附有课堂练习,每一章后附有本章小结和习题,以有助于学生的预习和复习。

本书适合作为高职高专计算机程序设计基础教材,也适合作为中职及社会各类人士的自学参考书。

### 图书在版编目 (CIP) 数据

C 语言程序设计/胡钢主编. —北京:中国铁道出版社, 2008.6

(21 世纪高职高专计算机系列规划教材)

ISBN 978-7-113-08775-3

I. C… II. 胡… 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 (2008) 第 089551 号

书 名: C 语言程序设计

作 者: 胡 钢 主编

---

策划编辑: 严晓舟 秦绪好

责任编辑: 黄园园

编辑部电话: (010) 63583215

编辑助理: 王春霞

封面制作: 白 雪

责任印制: 李 佳

---

出版发行: 中国铁道出版社 (北京市宣武区右安门西街 8 号 邮政编码: 100054)

印 刷: 北京鑫正大印刷有限公司

版 次: 2008 年 8 月第 1 版 2008 年 8 月第 1 次印刷

开 本: 787mm×1092mm 1/16 印张: 15.75 字数: 366 千

印 数: 5 000 册

书 号: ISBN 978-7-113-08775-3/TP·2806

定 价: 23.00 元

---

版权所有 侵权必究

本书封面本书封面贴有中国铁道出版社激光防伪标签,无标签者不得销售

凡购买铁道版的图书,如有缺页、倒页、脱页者,请与本社计算机图书批销部调换。

# 前 言

C 语言是目前国内外广泛使用的程序设计语言之一。C 语言功能强、使用灵活、可移植性好、目标代码质量高，既具有高级语言的特点，同时也具有低级语言的特点；既适合编写应用程序，也适合开发系统软件。C 语言程序设计是高等院校普遍开设的一门计算机程序设计课程。

本书对知识点的讲解由浅入深，强调算法设计，突出编程思路，注重实例讲解对学生动手能力的培养。全书共分 11 章，主要内容包括：C 语言概述，算法，C 语言基础，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，结构体、枚举类型与共同体，文件等。

目前市面上已出版不少同类书籍，但作为高职高专教材，学生普遍感到其内容要么过于简单，要么过于深奥，内容与实际应用联系不紧密。本书作者根据多年教学经验，结合高职教育的特点，编写了这本适合我国高职学生的 C 语言程序设计课程的教材。本教材具有以下特点：

(1) 本书讲解的知识点坚持实用原则，重点讲解基础、必要的知识点，省略一些不常用的知识点。

(2) 贯穿以应用为主线原则，在例题和习题的选择上尽量考虑生活中的实例，使学生把 C 语言学习融入生活中。

(3) 本书注重对学生算法思想的培养，引导学生逐步提升编程能力，为后续课程打下坚实的基础。

(4) 在每一章的开始都列出了本章的主要内容和学习重点，在每一小节后都列出课堂练习，在每一章后都附有本章小结和习题，帮助学生进行预习和复习。

本书为四川省精品课程，精品课程网址为：[http://jpkc.scitc.com.cn/jpkc/2007/c/QH\\_INDEX.ASP](http://jpkc.scitc.com.cn/jpkc/2007/c/QH_INDEX.ASP)。

本书由胡钢任主编，何华夏、夏容任副主编。参与本书编写的有四川信息职业技术学院胡钢、王坤，四川省广元市广播电视大学何华夏，南昌航空大学科技学院夏容，四川省泸县一中钟学均，其中第 1、2 章由钟学均编写，第 4、5 章由王坤编写，第 6 章由何华夏编写，第 7~10 章由夏容编写，第 3、11 章由胡钢编写，全书由胡钢负责统稿。

编 者  
2008 年 5 月

# 目 录

<b>第 1 章 C 语言概述</b> .....	1
1.1 C 语言产生的历史背景 .....	1
1.2 C 语言的特点 .....	2
1.3 简单的 C 程序介绍 .....	3
1.4 C 程序的运行环境 .....	6
1.4.1 Turbo C 2.0 的运行环境 .....	6
1.4.2 Dev-C++ 运行环境简介 .....	9
1.4.3 Turbo C/C++ for windows 集成实验与学习环境简介 .....	11
1.4.4 Visual C++ 6.0 运行环境简介 .....	12
本章小结 .....	13
习题一 .....	13
<b>第 2 章 算法</b> .....	14
2.1 算法的概念 .....	14
2.2 算法的特性 .....	16
2.3 算法的表示 .....	16
2.3.1 用自然语言表示算法 .....	17
2.3.2 用传统流程图表示算法 .....	17
2.3.3 用 N-S 流程图表示算法 .....	20
2.3.4 用计算机语言实现算法 .....	21
2.4 结构化程序设计方法 .....	22
本章小结 .....	23
习题二 .....	23
<b>第 3 章 C 语言基础</b> .....	24
3.1 C 语言数据类型 .....	24
3.2 常量与变量 .....	25
3.2.1 常量 .....	25
3.2.2 变量 .....	27
3.3 运算符及表达式 .....	31
3.3.1 C 语言运算符简介 .....	32
3.3.2 算术运算符和算术表达式 .....	33
3.3.3 算术运算符的结合性与优先级 .....	34
3.3.4 关系运算符和关系表达式 .....	34
3.3.5 逻辑运算符和逻辑表达式 .....	35
3.3.6 赋值运算符和赋值表达式 .....	37
3.3.7 逗号运算符和逗号表达式 .....	39

3.3.8 条件运算符和条件表达式 .....	39
3.4 常用数学函数 .....	40
本章小结 .....	42
习题三 .....	42
<b>第 4 章 顺序结构程序设计 .....</b>	<b>46</b>
4.1 顺序结构程序设计基础 .....	46
4.2 输入函数 .....	47
4.2.1 字符输入函数 .....	47
4.2.2 格式输入函数 .....	48
4.3 输出函数 .....	51
4.3.1 字符输出函数 .....	51
4.3.2 格式输出函数 .....	51
4.4 顺序结构程序举例 .....	53
本章小结 .....	55
习题四 .....	55
<b>第 5 章 选择结构程序设计 .....</b>	<b>57</b>
5.1 关系运算符和关系表达式 .....	57
5.1.1 关系运算符 .....	57
5.1.2 关系表达式 .....	58
5.2 逻辑运算符和逻辑表达式 .....	59
5.2.1 逻辑运算符及其优先次序 .....	59
5.2.2 逻辑表达式 .....	60
5.3 选择结构实现: if 语句 .....	62
5.3.1 if 语句的三种形式 .....	63
5.3.2 if 语句的嵌套 .....	66
5.3.3 条件运算符 .....	68
5.4 选择结构实现: switch 语句 .....	69
5.5 选择结构程序设计举例 .....	71
本章小结 .....	73
习题五 .....	73
<b>第 6 章 循环结构程序设计 .....</b>	<b>76</b>
6.1 循环结构概述 .....	76
6.2 for 循环 .....	77
6.3 break 语句和 continue 语句 .....	79
6.4 while/do...while 语句 .....	81
6.4.1 while 语句 .....	81
6.4.2 do...while 语句 .....	83
6.4.3 goto 语句 .....	86

6.5 重要循环及循环嵌套 .....	87
6.6 循环结构程序设计举例 .....	88
本章小结 .....	91
习题六 .....	91
<b>第 7 章 数组 .....</b>	<b>93</b>
7.1 数组的概念 .....	93
7.2 一维数组 .....	95
7.2.1 一维数组的定义 .....	95
7.2.2 一维数组的使用 .....	96
7.2.3 一维数组应用举例 .....	98
7.3 二维数组 .....	102
7.3.1 二维数组的定义 .....	102
7.3.2 二维数组的使用 .....	103
7.4 字符数组 .....	106
7.4.1 字符数组的定义 .....	106
7.4.2 字符数组的初始化 .....	106
7.4.3 字符数组的输入/输出 .....	107
7.4.4 字符串函数处理函数 .....	108
7.5 数组的应用举例 .....	111
本章小结 .....	114
习题七 .....	114
<b>第 8 章 函数 .....</b>	<b>119</b>
8.1 函数概述 .....	119
8.2 函数的参数和函数的值 .....	121
8.2.1 形式参数和实际参数 .....	121
8.2.2 函数的返回值 .....	122
8.3 函数的定义 .....	124
8.3.1 无参函数的定义 .....	124
8.3.2 有参函数的定义 .....	125
8.3.3 空函数 .....	126
8.4 函数的调用 .....	126
8.4.1 函数调用的一般形式 .....	126
8.4.2 函数调用的方式 .....	127
8.4.3 被调用函数的声明和函数原型 .....	127
8.4.4 函数的嵌套调用 .....	128
8.4.5 函数的递归调用 .....	130
8.5 数组作为函数参数 .....	131
8.5.1 数组元素作为函数实参 .....	131



8.5.2 数组名作为函数参数 .....	132
8.6 变量作用域和存储类别 .....	136
8.6.1 局部变量和全局变量 .....	136
8.6.2 变量的存储类别 .....	139
8.7 函数应用举例 .....	143
本章小结 .....	145
习题八 .....	146
<b>第 9 章 指针</b> .....	<b>149</b>
9.1 指针变量 .....	149
9.1.1 地址及指针概述 .....	149
9.1.2 变量的指针和指向变量的指针变量 .....	151
9.1.3 指针变量的定义 .....	151
9.1.4 指针变量的运算 .....	151
9.1.5 指针变量的引用 .....	153
9.1.6 指针变量作为函数参数 .....	154
9.2 指针和数组 .....	157
9.2.1 指向数组元素的指针变量 .....	157
9.2.2 通过指针引用数组元素 .....	158
9.2.3 用数组的指针作为函数参数 .....	161
9.2.4 多维数组与指针 .....	162
9.3 字符串与指针 .....	165
9.3.1 字符指针的定义与引用 .....	165
9.3.2 字符指针作为函数参数 .....	167
9.4 指针数组 .....	168
9.4.1 用指针数组处理二维数组 .....	168
9.4.2 用字符指针数组处理字符串组 .....	169
9.4.3 用指针数组作为 main 函数的形参 .....	171
9.5 指向指针的指针 .....	172
9.6 指针与函数 .....	174
9.6.1 指针型函数 .....	174
9.6.2 函数指针变量 .....	175
9.7 程序应用举例 .....	177
本章小结 .....	178
习题九 .....	179
<b>第 10 章 结构体、枚举类型与共用体</b> .....	<b>182</b>
10.1 结构体的概念和结构体变量 .....	182
10.1.1 结构体的概念 .....	182
10.1.2 结构体类型的定义 .....	183

10.1.3	结构体类型变量的定义.....	184
10.1.4	结构体变量的引用.....	185
10.1.5	结构体变量的初始化.....	186
10.2	结构体数组和结构体指针.....	187
10.2.1	结构体数组的定义.....	187
10.2.2	结构体数组的初始化.....	188
10.2.3	结构体数组举例.....	189
10.2.4	结构体指针.....	189
10.3	枚举类型和共用体简介.....	191
10.3.1	枚举类型和枚举变量的定义.....	191
10.3.2	枚举变量的引用.....	192
10.3.3	共用体类型和共用体变量的定义.....	193
10.3.4	共用体变量的引用以及共用体类型数据的特点.....	194
	本章小结.....	195
	习题十.....	195
<b>第 11 章</b>	<b>文件.....</b>	<b>200</b>
11.1	文件的基本概念.....	200
11.2	文件的打开与关闭.....	201
11.2.1	文件的打开.....	201
11.2.2	文件的关闭.....	203
11.3	文件的读写.....	203
11.3.1	字符与字符串读写函数.....	204
11.3.2	数据块读写函数 fread 和 fwrite.....	208
11.3.3	格式化读写函数 fscanf 和 fprintf.....	210
11.3.4	文件的定位.....	211
11.3.5	文件检测函数.....	212
11.4	文件操作举例.....	213
	本章小结.....	215
	习题十一.....	216
附录 A	Turbo C 2.0 编译错误信息.....	217
附录 B	常用字符与 ASCII 代码对照表.....	223
附录 C	C 语言的 32 个关键字.....	224
附录 D	C 语言中的运算符及其优先级.....	226
附录 E	参考答案.....	228
参考文献	.....	240

# 第 1 章

## C 语言概述

任何事物的产生和发展都有其特定的历史背景，C 语言作为世界上应用最广泛、最有发展前景的一种计算机高级语言，它的产生和发展也有其历史背景。C 语言为何成为程序设计入门语言，它有何特点，与其他高级语言有何不同，怎样编写一个 C 语言程序，是本章要学习的内容。

### 主要内容

- C 语言产生的历史背景。
- C 语言的特点。
- 简单 C 语言程序的编写与运行。

### 学习重点

- 在 Turbo C 2.0 环境下调试 C 语言程序的方法。

## 1.1 C 语言产生的历史背景

当人们已经具有了解决某类问题的方法后，总是想再将这个方法加以改进、简化，以提高处理和解决问题的效率，C 语言就是在这样的思想下产生的。

在 C 语言产生之前的系统软件主要是用汇编语言编写的（包括 UNIX 操作系统）。由于汇编语言不是与计算机硬件独立的，因此所编写的程序的可移植性很差。要想提高程序的移植性，必须使用高级语言来编写，但以前的高级语言难以实现汇编语言的某些功能，所以人们就想找到一种既具有高级语言的移植性强，又能像汇编语言那样对硬件直接进行操作的语言。C 语言就是在这种思想的指导下产生的，它汇聚了高级语言和汇编语言的优点。

C 语言是在 B 语言的基础上发展起来的，其根源可以追溯到 ALGOL 60。1963 年，剑桥大学把 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，于是就产生了 BCPL 语言。1970 年，美国贝尔实验室的 Ken Thompson 又对 BCPL 语言进行了修改，并为它取了一个有趣的名字“B 语言”，意思是将 CPL 语言浓缩，提炼出它的精华。并且，他用 B 语言写了第一个 UNIX 操作系统。到了 1973 年，美国贝尔实验室的 Dennis M.Ritchie 在 B 语言的基础上最终设计出了一种新的语言，他取了 BCPL 的第二个字母作为这种语言的名字，这就是 C 语言。1977 年，Dennis M.Ritchie 发表了不依赖于具体机器系统的 C 语言编译文章《可移植的 C 语言编译程序》。1978 年，Brian W.Kernighan 和 Dennis

M.Ritchie 出版了名著《The C Programming Language》，从而使 C 语言成为当前世界上最流行的一种高级程序设计语言。



### 课堂练习 1.1

了解 C 语言的由来。

## 1.2 C 语言的特点

C 语言之所以能够迅速地成为风靡全世界和应用最广泛的计算机语言之一，这与它的特点是密不可分的。C 语言的特点主要有以下几点：

### (1) C 语言简洁紧凑，使用方便

C 语言非常简洁，使用起来非常方便；只有 32 个关键字，9 种控制语句，程序书写形式灵活、自由；关键字采用小写字母表示，压缩了一些不必要的成分，程序简洁明了。下面将 C 语言和 Pascal 语言进行比较，如表 1-1 所示。

表 1-1 C 语言和 Pascal 语言比较

C 语言	Pascal 语言	语句功能
int i;	VAR i:INTEGER	定义 i 为整型变量
int a[10];	VAR a:ARRAY[1..10] OF INTEGER	定义 a 为整型一维数组
int *m;	VAR m:↑INTEGER	定义 m 为指向整型变量的指针变量

从以上的比较可以看出，C 语言比 Pascal 语言的表达要简洁得多，因此 C 语言程序短，减少了录入程序的工作量，极大地提高了人们的工作效率。

### (2) C 语言的运算符丰富

C 语言除了具有其他高级语言的运算符外，还增加了很多其他运算符，共有 34 种。C 语言把赋值、强制类型转换等都作为运算符处理。灵活地使用各种运算符，可以实现其他高级语言难以实现的运算，这就为 C 语言实现强大、复杂的功能提供了坚实的基础。

### (3) C 语言的数据类型丰富。

C 语言具有现代化语言的各种数据结构，其数据类型有整型、实型、字符型、数组型、指针型、结构体、共用体等。这使得 C 语言能够实现各种复杂的数据结构的运算，特别是指针类型可以实现直接对内存地址和寄存器进行操作。

### (4) C 语言程序设计自由度大，语法限制少

C 语言的语法限制较少，这就增大了程序设计的自由度，使程序设计更灵活。例如，在 C 语言中虽然事先定义了数组的大小，但它并不对数组下标进行检查。而且，C 语言对变量的数据类型使用比较灵活，例如整型、字符型和逻辑型数据可以通用，不像许多高级语言那样只有相同类型的数据才能进行运算，这就极大地方便了程序编写。

### (5) C 语言是一种结构化控制语言

C 语言具有结构化的控制语句，如 if...else 语句、switch 语句、for 语句、do...while 语句等，并用函数作为模块实现程序的模块化，因此是一种符合结构化程序设计思想的语言。这一特点为大型软件模块化、集体共同开发软件提供了有力的支持。

### (6) C语言程序具有良好的移植性

C语言的输入/输出是通过调用C语言函数库中的输入/输出函数来实现的,这些函数与计算机硬件是无关系的,因此C语言程序可以很好地在不同硬件的计算机之间移植。也就是说,C语言程序可以在不同硬件的计算机上很好地运行,而无须改动程序。

### (7) C语言可直接对硬件进行操作

C语言能够直接访问物理地址,可以进行位(bit)操作,可直接对硬件进行操作,能够实现汇编语言的很多功能。因此,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,可用于编写系统软件,例如著名的FoxPro系统软件就是用C语言编写的。

### (8) C语言编译后生成的目标代码体积小、质量高、速度快

C语言程序可以完全脱离原编译环境执行。因此,C语言特别适用于过程控制、智能仪表、家用电器等嵌入式编程,应用领域广泛。

C语言也有其不足之处:

(1) 由于C语言语法灵活,在某种程度上降低了程序的安全性,因此对程序员也提出了更高的要求。

(2) C语言适用于底层开发和小型精巧程序的开发(如硬件驱动、手机应用软件等),不适合用于开发大型应用软件。



## 课堂练习 1.2

1. C语言的特点主要有哪些?
2. C语言有哪些不足之处?

## 1.3 简单的C程序介绍

我们认识一个人,总是先认识他的外貌特征,对他有个大概的了解,然后再通过进一步的接触、交流,认识他的思想行为和道德品质等深层的东西。也就是说,认识人和事物总是一个由简单到复杂的循序渐进的过程,学习一门语言也是如此,所以下面介绍两个简单的C程序,从中了解C程序的特性。

**【例 1-1】**用C语言实现两个数求和。

参考源代码:

```
#include <stdio.h>
main()
{
    int x,y,sum;           /*定义 x,y,sum 为整型变量*/
    x=120;
    y=230;
    sum=x+y;
    printf("%d",sum);     /*输出变量 sum 的值*/
}
```

下面是以上程序中每一行的说明：

(1) 第 1 行是输入/输出函数编译预处理命令。

(2) 第 2 行 `main()` 表示主函数，每一个 C 语言程序都必须有一个 `main()` 函数。函数体是用一对大括号“{}”括起来的。

(3) 第 4 行定义变量 `x`、`y`、`sum` 为整型变量。“/\* \*/”是注释符号，注释符号中间的是注释内容，注释对程序的运行不起作用，只是给阅读程序的人带来方便，一般用来注明语句或程序段的功能是什么。

(4) 第 5 行和第 6 行是赋值语句，将整数 120 和 230 分别赋给变量 `x` 和 `y`。

(5) 第 7 行将变量 `x` 和 `y` 的值相加后赋给变量 `sum`，即 `sum` 中存放的是 `x` 和 `y` 之和。

(6) 第 8 行是输出语句，其中“%d”表示输出数值的类型是十进制整数。整个语句的作用是输出变量 `sum` 的值。

说明：C 程序的函数体中的每个语句之后都要用结束符“;”。

**【例 1-2】**输入两个整数，输出两个数中的最大数。

参考源代码：

```
#include <stdio.h>           /*输入/输出函数编译预处理命令*/
main()                       /*主函数*/
{
    int x,y,z;               /*定义整型变量 x,y,z*/
    scanf("%d,%d",&x,&y);    /*输入变量 x,y 的值*/
    z=max(x,y);             /*调用函数 max,并将函数值赋给变量 z*/
    printf("max=%d",z);     /*输出变量 z 的值*/
}
int max(a,b)                /*定义函数 max,函数值为整型,a,b为形参*/
int a,b;                    /*定义形参 a,b的数据类型为整型*/
{
    int c;                  /*定义整型变量 c*/
    if (a>b) c=a;          /*比较 a,b 的大小,如果 a 大则把 a 赋给变量 c*/
    else c=b;              /*否则把 b 赋给变量 c*/
    return(c);             /*返回 c 的值,通过 max 函数返回到调用处*/
}
```

程序说明：

(1) 第 5 行的 `scanf()` 函数的作用是输入变量 `x`、`y` 的值。`&x,&y` 中的“&”的意思是“取地址”。整个语句的含义是输入两个整数分别存储到变量 `x`、`y` 所在的存储单元中，简单来说就是分别给变量 `x`、`y` 输入一个整数。

(2) 第 6 行是调用用户自己定义的函数 `max()`，并将函数值赋给变量 `z`。程序执行到该语句时，将实际参数 `x` 和 `y` 的值分别传递给 `max()` 函数的形式参数 `a` 和 `b`，`max()` 函数执行后得到一个返回值（即 `max()` 函数中变量 `c` 的值），然后把这个值赋给变量 `z`。

(3) 第 7 行是输出语句，`printf()` 函数中的“`max=%d`”，在输出时“%d”将被 `z` 的值代替，而“`max=`”原样输出。如果在程序运行时，输入的是 25、30，则输出结果是：

```
max=30
```

说明：本程序涉及函数调用、实际参数、形式参数等概念，这些内容将在后面的章节中陆续学习，读者现在可以不予以深究，只需大概地了解即可。

通过前面两个例子可以看出：

(1) C程序是由函数构成的

函数是C程序的基本单位，一个C程序由若干个函数构成（至少有一个main()函数）。构成C程序的函数可以是系统提供的库函数（如scanf()和printf()函数），也可以是用户自己根据需要设计的函数（如例1-2中的max()函数，这种函数称为自定义函数）。

(2) 一个函数由函数的说明部分和函数体两部分构成

① 函数的说明部分。包括函数名、函数类型、函数属性、函数参数名、函数参数类型等。例如，例1-2中的函数max()的说明部分如图1-1所示。

说明：函数名后面必须跟一对圆括号，函数参数可以没有，如main()。

② 函数体。函数体即函数说明部分下面的大括号{...}内的部分，若一个函数有多个大括号，则最外层的一对大括号为函数体的范围。

函数体一般包括以下两部分：

- 定义变量。例如，例1-1中main()函数中的int x,y,sum。
- 执行部分。执行部分由若干个语句组成，每个语句之后要用“;”表示语句结束。

说明：一个C程序总是从主函数main()开始执行的，与main()函数的位置无关。

之所以把上面的代码称为“参考源代码”，是因为程序无定式，无绝对写法，只要完成相应功能即可。好像一篇作文，虽然题目一样，但各人写法不同。

由以上两个例子可以看到，C程序的一般组成形式如下：

```
main()                /*主函数说明*/
{
    变量定义;          /*主函数体*/
    执行语句组;
}
函数类型 函数名 1(参数) /*子函数说明*/
{
    变量定义;          /*子函数体*/
    执行语句组;
}
...
函数类型 函数名 N(参数) /*子函数说明*/
{
    变量定义;          /*子函数体*/
    执行语句组;
}
```

其中，函数名1~函数名N是用户自定义的函数。

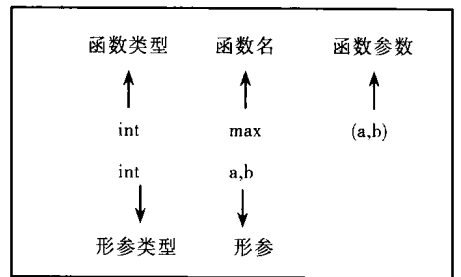


图 1-1 函数 max 说明



### 课堂练习 1.3

依照例题，编写一个求长方形面积的 C 语言程序。

## 1.4 C 程序的运行环境

通过前面的学习，我们对 C 语言已有了初步了解。对 C 语言源程序结构有了总体的认识，那么如何在机器上运行 C 语言源程序？任何高级语言源程序都要“翻译”成机器语言才能在机器上运行。“翻译”的方式有两种，一种是解释方式，即对源程序解释一句执行一句；另一种是编译方式，即先把源程序“翻译”成目标程序（用机器代码组成的程序），再经过连接装配后生成可执行文件，最后执行可执行文件而得到结果。

C 语言是一种编译型的程序设计语言，它采用编译的方式将源程序翻译成目的程序（机器代码）。运行一个 C 程序，从输入源程序开始，要经过编辑源程序文件（.c）、编译生成目标文件（.obj）、连接生成可执行文件（.exe）和执行四个步骤。

C 语言有比较多的版本，不同版本的运行环境有所不同。本书着重介绍 Turbo C 2.0 环境下 C 语言源程序运行的方法（本书的所有程序都是在 Turbo C 2.0 下调试运行的），并简单介绍 Dev-C++、Turbo C/C++ for windows 和 Visual C++ 6.0 的运行环境。

### 1.4.1 Turbo C 2.0 的运行环境

Turbo C 是美国 Borland 公司推出的 C 语言编译程序。它具有方便、直观、易用的界面和丰富的库函数。它向用户提供了集成环境，把程序的编辑、编译、连接和运行等操作全部集中在一个界面上进行，使用十分方便。

#### 1. Turbo C 2.0 的工作环境简介

C 程序可以在任意文字编辑软件中编辑，例如 Edit、Word、WPS、记事本等。但要注意，如果用 Word、WPS 文字处理软件编辑时，保存文件的类型应为纯文本，并将文件的扩展名命名为.c。

一般情况下，往往都直接用 Turbo C 或 Borland C 自带的编辑器直接编辑 C 语言源程序。

启动 Turbo C 的编辑器有两种方式，假设 Turbo C 安装在 E 盘的 TC 目录下，则启动 Turbo C 编辑器有如下两种方式。

#### （1）从 DOS 环境进入

在 DOS 命令行上键入：

```
C:\>E:
```

```
E:\>CD TC      (指定当前目录为 TC 子目录)
```

```
E:\TC>TC      (进入 Turbo C 2.0 环境)
```

这时，进入 Turbo C 2.0 集成环境的主菜单窗口，如图 1-2 所示。

#### （2）从 Windows 环境进入

在 Windows 环境中，打开“我的电脑”窗口，进入 E 盘，进入 TC 文件夹，双击 tc.exe 图标即可进入 C 语言环境。





图 1-2 Turbo C 2.0 编辑窗口

## 2. 认识 Turbo C 2.0 编辑器

Turbo C 2.0 编辑器由主菜单、编辑窗口、信息窗口、功能键提示区 4 个部分组成，下面介绍各个部分的功能。

### (1) 主菜单

编辑器的顶部显示的就是主菜单，它提供了 8 个选择项，每个菜单项的主要功能如下：

① File：处理文件（装入、存盘、选择、建立、换名存盘、写盘），目录操作（列表、改变工作目录），退出 Turbo C，返回 DOS 状态。

② Edit：建立、编辑源文件。

③ Run：自动编辑、连接并运行程序。

④ Compile：编辑、生成目标文件组合成工作文件。

⑤ Project：将多个源文件和目标文件组合成工作文件。

⑥ Option：提供集成环境下的多种选择和设置（如设置存储模式、选择编参数、诊断及连接任选项）以及定义宏；也可记录 Include、Output 及 Library 文件目录，保存编译任选项和从配置文件加载任选项。

⑦ Debug：检查、改变变量的值、查找函数，程序运行时查看调用栈。选择程序编译时是否在执行代码中插入调试信息。

⑧ Break/Watch：增加、删除、编辑监视表达式，及设置、清除、执行至断点。

在主菜单中，Edit 仅仅是一条进入编辑器的命令。其他选项均为下拉式菜单，包含许多命令选项，使用方向键移动光带来选择某个选项时，按【Enter】键，表示执行该命令，若屏幕上弹出一个下拉菜单，可以提供进一步选择。

### (2) 编辑窗口

编辑窗口是在主菜单下，信息窗口之上的区域，其顶行中间有 Edit 标志。在此窗口中可以建立、编辑一个源文件。按【F5】功能键可以扩大编辑窗口到整个屏幕，或恢复分屏式环境。

进入编辑窗口的方式有两种：

① 按【F10】功能键，激活主菜单，然后用光标移动键将光带移到 Edit 上，按【Enter】键，或者在激活主菜单后直接按字母键【E】，均可进入编辑窗口。

② 按【Alt+E】键无条件地进入编辑窗口。

进入编辑窗口后，编辑窗口的名字是高亮度的，表示它是活动窗口。窗口的顶部第一行是状态行，给出有关正在被编辑文件的信息，如当前光标所在的行、列，编辑模式（插入/改写），正