



中等专业学校
电子信息类 规划教材



• 孙振业 史宝慧 编著

C语言及应用

西安电子科技大学出版社

中等专业学校 规划教材
电子信息类

C 语言 及 应 用

孙振业 史宝慧 编著

西安电子科技大学出版社

内 容 简 介

本教材主要介绍 C 语言的基本语法及结构化程序设计方法, 内容主要包括: C 语言的顺序程序设计、分支程序设计、循环程序设计、数组、函数、指针、结构、文件等, 并简单介绍了 C 语言的高级应用。本教材在注重介绍 C 语言的定义、概念和使用规则的同时, 突出介绍了结构化程序设计的方法和技巧。书中配有大量的例题、习题及程序分析。

本教材适合于中等专业学校计算机类专业, 也适合用作非计算机专业的 C 语言普及, 同时也是一本很好的自学读本。

中等专业学校
电子信息类 规划教材

C 语言及应用

孙振业 史宝慧 编著

责任编辑 梁家新

出版发行 西安电子科技大学出版社(西安市太白南路 2 号)

电 话 (029)8227828 邮 编 710071

http: /www.xduph.com E-mail: xdupfxb@pub.xaonline.com

经 销 新华书店

印 刷 铁一局印刷厂

版 次 1999 年 3 月第 1 版 2004 年 1 月第 5 次印刷

开 本 787 毫米×1092 毫米 1/16 印张 17.75

字 数 423 千字

印 数 18 001~22 000 册

定 价 18.00 元

ISBN 7-5606-0700-1/TP·0361

XDUP 0970001 5

*** 如有印装问题可调换 ***

本社图书封面为激光防伪覆膜, 谨防盗版。

出版说明

为做好全国电子信息类专业“九五”教材的规划和出版工作,根据国家教委《关于“九五”期间普通高等教育教材建设与改革的意见》和《普通高等教育“九五”国家级重点教材立项、管理办法》,我们组织各有关高等学校、中等专业学校、出版社,各专业教学指导委员会,在总结前四轮规划教材编审、出版工作的基础上,根据当代电子信息科学技术的发展和面向 21 世纪教学内容和课程体系改革的要求,编制了《1996—2000 年全国电子信息类专业教材编审出版规划》。

本轮规划教材是由个人申报,经各学校、出版社推荐,由各专业教学指导委员会评选,并由我们与各专指委、出版社协商后审核确定的。本轮规划教材的编制,注意了将教学改革力度较大、有创新精神、有特色风格的教材和质量较高、教学适用性较好、需要修订的教材以及教学急需、尚无正式教材的选题优先列于规划。在重点规划本科、专科和中专教材的同时,选择了一批对学科发展具有重要意义,反映学科前沿的选修课、研究生课教材列入规划,以适应高层次专门人才培养的需要。

限于我们的水平和经验,这批教材的编审、出版工作还可能存在不少缺点和不足,希望使用教材的学校、教师、学生和其他广大读者积极提出批评和建议,以不断提高教材的编写、出版质量,共同为电子信息类专业教材建设服务。

电子工业部教材办公室

前 言

本教材系按电子工业部的《1996—2000 年全国电子信息类专业教材编审出版规划》，由全国中专计算机专业教学指导委员会编审、推荐出版。本教材由北京无线电工业学校孙振业担任主编，史宝慧参编，北京计算机工业学校校长、高级讲师武马群担任主审，贵州无线电工业学校高级讲师黄大胜担任责任编委。

本教材的参考学时数为 80 学时，其主要内容是以微型计算机为背景，介绍 C 语言程序设计的基本概念和方法。具体有以下几部分：C 语言的结构化程序设计思想、C 语言的顺序程序设计、分支程序设计、循环程序设计、数组、字符串、结构、指针、文件，以及计算机绘图、工程文件、菜单的制作等。本教材在注重介绍 C 语言的定义、概念和使用规则的同时，突出介绍了程序设计的技巧和方法，并配置了大量的程序例题和习题，供读者参考。

本教材在编写与修改过程中，得到主审武马群、责任编委黄大胜、北京无线电工业学校副校长、高级讲师曾玉昆等老师的精心指导与帮助，北京无线电工业学校卢海、朱立波为本书作了大量的录入与程序的调试工作，这里一并表示诚挚的感谢。由于编者水平有限，书中难免还存在一些缺点和错误，殷切希望广大读者批评指正。

编者

1998 年 10 月

目 录

第 1 章 C 语言概述	1
1.1 C 语言的特点及发展历史	1
1.2 C 语言的基本符号	2
1.3 C 语言程序的结构	4
1.4 标准输入输出库函数	5
1.5 C 语言程序的编辑及运行	10
习题 1	13
第 2 章 数据	14
2.1 常量	14
2.2 变量	18
2.3 运算符和表达式	24
习题 2	33
第 3 章 C 语言程序的控制结构	35
3.1 结构化程序设计	35
3.2 顺序结构	37
3.3 分支结构	39
3.4 循环结构	51
3.5 辅助语句	57
3.6 应用举例	59
习题 3	62
第 4 章 数组	65
4.1 数组的概念	65
4.2 一维数组	66
4.3 二维数组	69
4.4 字符数组	74
4.5 应用举例	83
习题 4	87
第 5 章 函数	90
5.1 函数的定义	90
5.2 函数的调用	96
5.3 编译预处理	105
5.4 标准库函数	111
5.5 应用举例	115
习题 5	119
第 6 章 结构与联合	121
6.1 结构	121
6.2 联合	127

6.3 位字段	129
6.4 枚举	135
6.5 自定义类型	138
习题 6	139
第 7 章 指针	140
7.1 指针的概念	140
7.2 变量的指针与指针变量	142
7.3 指针与数组	147
7.4 指针与函数	156
7.5 指针与结构	162
7.6 结构与链表	166
7.7 应用举例	178
习题 7	185
第 8 章 文件	186
8.1 文件概述	186
8.2 缓冲型文件输入输出系统	187
8.3 非缓冲型文件输入输出系统	204
8.4 标准设备文件及 I/O 改向	209
8.5 工程文件	211
习题 8	214
第 9 章 C 语言应用	215
9.1 C 语言的命令行编译	215
9.2 C 语言的指针及编译模式	216
9.3 文本屏幕输出和文本窗口	219
9.4 菜单技术	225
9.5 Turbo C 的图形处理	231
9.6 C 语言与 BIOS 和 DOS 的接口	245
9.7 C 语言与汇编语言的混合编程	249
附录 1 ASCII 码字符表	257
附录 2 Turbo C 常用库函数	259
附录 3 Turbo C 集成环境介绍	271

第1章 C语言概述

随着计算机的发展和普及, C语言在各个领域的应用越来越广泛。几乎各类计算机都支持C语言的开发环境, 这也为C语言的普及及应用奠定了基础。

本章主要介绍C语言的发展历史及特点, C语言程序的结构, C语言程序的编辑及运行方法。

1.1 C语言的特点及发展历史

1.1.1 C语言的发展历史

C语言是一种编译性程序设计语言, 它的前身是BCPL(Basic Combined Programming Language)语言, 1967年由英国剑桥大学的Martin Richard推出。1970年, 贝尔实验室的K. Thompson以BCPL语言为基础, 开发了B语言, 并用B语言编写了UNIX操作系统, 在PDP-7上实现。1972年, 贝尔实验室的D. M. Ritchie在B语言的基础上设计出C语言, C语言既保持了BCPL语言和B语言的优点(精练、接近硬件), 又克服了它们的缺点(过于简单)。1973年, K. Thompson和D. M. Ritchie合作把UNIX的90%以上用C语言改写, 并加进了多道程序设计的功能, 称为UNIX第五版, 开创了UNIX系统发展的新局面。1975年UNIX第六版颁布后, C语言得到计算机界的普遍认可, 并与UNIX系统一起互相促进, 迅速发展。

最初的C语言只是为描述和实现UNIX操作系统提供一种工作语言而设计的。目前, C语言已独立于UNIX系统, 先后被移植到大、中、小型计算机上。1978年, B. Kernighan和D. M. Ritchie合作编写了经典著作“The C Programming Language”, 它是目前所有C语言版本的基础。1983年, 美国国家标准化协会(ANSI)对C语言问世以来的各种版本进行扩充, 制定了ANSI C。现在流行的C语言版本有: Microsoft C, Turbo C, Quick C, Borland C等。本书以介绍Turbo C为主, 并简单介绍C语言的高级应用。

1.1.2 C语言的特点

C语言有以下几个基本特点:

(1) C语言不但具有高级语言的特性, 还具有汇编语言的特点。既有高级语言面向用户、容易记忆、便于阅读和书写的优点; 又有面向硬件和系统, 可以直接访问硬件的功能。

(2) C语言是结构化程序设计语言。C语言程序的逻辑结构可以用顺序、选择和循环三种基本结构组成, 便于采用自顶向下、逐步细化的结构化程序设计技术。用C语言编制的程序, 具有容易理解、便于维护的优点。

(3) C 语言是模块化程序设计语言。C 语言的函数结构、程序模块间的相互调用及数据传递和共享技术,为大型软件设计的模块化分解技术,及软件工程技术的应用提供了强有力的支持。

(4) C 语言具有丰富的运算能力。除具有一般高级语言所拥有的四则运算及逻辑运算功能外,还具有二进制的位(bit)运算、单项运算和复合运算等功能。

(5) C 语言具有预处理能力。能够对字符串或特定参数进行宏定义,能够对外部文本文件实现读取和合并。

(6) C 语言具有丰富的数据类型和较强的数据处理能力。不但具有整型、实型、双精度型,还具有结构、联合等构造类型,并为用户提供了自定义数据类型。

(7) C 语言具有较强的移植性。C 语言程序本身并不依赖于计算机的硬件系统,只要在不同种类的计算机上配置 C 语言编译系统,即可达到程序移植的目的。

(8) C 语言具有较好的通用性,它既可用于编写操作系统、编译程序等系统软件,也可用于编写各种应用软件。

1.2 C 语言的基本符号

1.2.1 基本符号集

C 语言的基本符号集是 ASCII 码字符集。包括:

(1) 大小写英文字母各 26 个。

(2) 阿拉伯数字 0~9。

(3) 特殊符号 39 个,包括运算符和操作符:

+	-	*	/	%	<
<=	>	>=	==	!=	&&
	!	&		~	=
++	--	?:	<<	>>	()
[]	.	->	^	#	sizeof
+=	-=	*=	/=	%=	&=
^=	=	,			

1.2.2 标识符

1. 标识符的作用

标识符的作用是表示常量、变量、函数及类型的名字。

2. 标识符的构成规则

(1) 标识符必须以英文字母开头,由字母、数字和下划线组成。例如:

abcd, y105, year_day, b20a

都是合法的标识符,而

1234a, a+b

则是不合法的。

(2) 大小写字母的含义不同。例如：

ABCD、Abcd 和 abcd

是完全不同的 3 个标识符。

(3) 一个标识符可以由多个字符组成，但一般只有前 8 个字符有效。例如：

integer_a、integer_b、integer_c

具有相同的意义，表示同一个标识符 integer_。

1.2.3 保留字

保留字是 C 语言编译系统固有的、有专门意义的标识符。C 语言的保留字包括全部的语句名和数据类型名。C 语言的保留字见表 1-1 所示。

表 1-1 C 语言保留字

描述数据类型定义	描述存储类型	描述数据类型	描述语句
typedef	auto	char	break
	extern	double	continue
	static	float	default
	register	int	case
		long	if
		short	else
		struct	do
		union	for
		unsigned	while
			goto
			sizeof
			return

保留字的使用说明：

(1) 所有保留字的字母都采用小写。

(2) 保留字不能再作为用户的常量、变量、函数和类型等的名字。

(3) 在语句中，保留字与数据或语句的其它部分，必须用空格或其它语法允许的专用字符分隔开。即，必须对保留字加以适当的分界，C 语言编译系统才能识别它们。

1.2.4 ASCII 码字符集

ASCII 码是美国标准信息交换码 (American Standard Coder for Information Interchange)。ASCII 码字符集中包含基本字符与控制字符两部分，见附录 1 所示。

基本字符在本节已作过介绍，ASCII 码字符集中，代码值为 32~127 的代码是基本字符。

控制字符一般是计算机发向外部设备的命令码,它们仅仅控制外部设备实现某些特定动作,并不是提供给用户的输出信息。在 ASCII 码字符集中,代码值为 0~31 的代码是控制代码。

C 语言中的字符代码采用 ASCII 码表示。

1.3 C 语言程序的结构

1.3.1 简单的 C 语言程序示例

例 1.1 求两个数中的大者。

```
/* 计算两个数的最大值 */
main()                               /* 主函数 */
{ float a,b;                         /* 变量说明 */
  scanf("%f%f",&a,&b);               /* 输入变量 a 和 b 的值 */
  if(a>b)                            /* 比较 a 和 b 的大小 */
    printf("%f\n",a);                /* 如果 a 大,则输出 a 的值 */
  else
    printf("%f\n",b);                /* 如果 b 大,则输出 b 的值 */
}
```

例 1.1 是一个简单的 C 语言程序。其中 main() 表示主函数,由大括号{}括起来的部分是函数体。/*...*/ 表示对程序的注释。scanf() 和 printf() 是标准输入输出库函数。main() 函数中定义了两个实型变量 a 和 b,并用 scanf() 函数从键盘输入 a 和 b 的值,if—else 是一个条件判断语句,如果 a>b 成立,则用 printf() 函数输出 a 的值,否则输出 b 的值。

例 1.2 求三个数的平均值。

```
float average(x,y,z)                 /* 定义 average 函数的值为实型 */
float x,y,z;                         /* 定义 average 函数的参数 x、y、z 为实型 */
{ float aver;                        /* 定义函数中的变量 aver 为实型 */
  aver=(x+y+z)/3;                    /* 计算 x、y、z 的平均值并赋给 aver */
  return(aver);                      /* 将 aver 的值返回主函数的调用处 */
}
main()                               /* 主函数 */
{ float a,b,c,ave;                  /* 定义 a、b、c、ave 为实型变量 */
  scanf("%f%f%f",&a,&b,&c);           /* 输入变量 a、b、c 的值 */
  ave=average(a,b,c);                /* 调用 average 函数并将返回值赋给 ave */
  printf("average=%f",ave);          /* 输出 ave 的值 */
}
```

例 1.2 程序包括两个函数:主函数 main() 和被调用函数 average()。主函数 main() 从键盘接收 a、b、c(称实际参数)的值,并传递给 average() 函数的 x、y、z(称形式参数);average() 函数计算 x、y、z 的平均值并赋给变量 aver,return 语句将 aver 的值返回给主函

数 main() 的调用处。

1.3.2 C 语言程序的结构特点

从上述两个例子可以看到，C 语言程序具有以下结构特点：

1. C 语言程序由函数构成

(1) 函数是 C 语言程序的基本单位。一个 C 语言程序至少由一个函数 main() 组成，也可以由一个 main() 函数和多个其它函数构成。例如，在例 1.1 中只包含一个 main() 函数，在例 1.2 中则包含两个函数 main() 和 average()，通常称 main() 为主调函数，average() 为被调函数。

(2) 被调用函数可以是系统提供的库函数(例如 scanf() 和 printf() 函数)，也可以是由用户根据需要自己编制的函数(例如例 1.2 中的 average() 函数)。

(3) 一个 C 语言程序总是从 main() 函数开始执行。这与 main() 函数在程序中的位置无关，因此，main() 函数可以放在程序的任何位置。

2. 一个函数由函数说明和函数体两部分组成

(1) 函数说明：包括对函数名、函数类型、函数参数名、函数参数类型的说明。

例如：例 1.2 中

```
float average(x,y,z)
float x,y,z;
```

前者是对函数名、函数参数名及函数类型的说明，后者是对函数参数类型的说明。

(2) 函数体：函数体是指由大括号括起来的部分。主要包括变量说明和执行语句两部分。

3. 一个函数由若干行组成

(1) 一行可由一个或多个语句组成，一个语句也可以分别写在若干行上。

(2) 语句或变量说明的最后必须有一个分号，分号是语句或变量说明的必要组成部分。

(3) 为了增强可读性，可以利用 /* ... */ 对 C 语言程序中的任何部分做注释。

4. C 语言程序的书写习惯是用小写英文字母

1.4 标准输入输出库函数

数据的输入输出是程序的重要组成部分，而 C 语言程序的输入输出是由系统提供的标准输入输出库函数实现的。本节首先介绍两种最常用的、简单的标准输入输出库函数。

1.4.1 编译预处理

在使用标准输入输出库函数时，要用编译预处理命令“#include”将“stdio.h”文件包含到用户程序中。一般形式如下：

```
#include "stdio.h"
```

stdio.h 文件包含了与标准输入输出库函数有关的变量定义、宏定义和函数定义(详细

内容见第 5 章)。在需要使用标准输入输出库函数时,应在程序之前使用上述编译预处理命令。

1.4.2 字符输入输出函数

1. 字符输出函数 putchar()

(1) putchar()的作用:向标准输出设备输出一个字符。

(2) putchar()的一般形式:

putchar(输出项);

其中,输出项可以是整型变量或字符型变量。

例 1.3 输出字符 B。

```
#include "stdio.h"      /* 编译预处理 */
main()
{ int ch;               /* 说明 ch 为整型变量 */
  ch=66;                /* 将 B 的 ASCII 码值赋给变量 ch */
  putchar(ch);          /* 输出 ch 的值 B */
}
```

例 1.4 输出字符 B。

```
#include "stdio.h"      /* 编译预处理 */
main()
{ char ch;              /* 说明 ch 为字符型变量 */
  ch='B';               /* 将字符'B'赋给变量 ch */
  putchar(ch);          /* 输出 ch 的值 B */
}
```

以上两个程序的运行结果都是输出字符 B。

2. 字符输入函数 getchar()

(1) getchar()的作用:从标准输入设备上接收一个字符。

(2) getchar()的一般形式:

getchar()

getchar()函数没有参数,函数值即是从输入设备接收的字符。

例 1.5 从键盘接收一个字符并输出。

```
#include "stdio.h"      /* 编译预处理 */
main()
{ char ch;              /* 说明 ch 为字符型变量 */
  ch=getchar();         /* 从键盘接收一个字符赋给变量 ch */
  putchar(ch);          /* 输出 ch 的值 */
}
```

程序运行结果如下:

a<CR> (键盘输入 a 后,按回车键,字符 a 送到变量 ch 中)

a (输出变量 ch 的值)

例 1.5 程序的输出结果中,“<CR>”表示“按回车键”,这是一种习惯写法,本书将采

用这种写法。

(3) getchar()的使用说明:

getchar()接收的字符可以赋给一个字符变量或整型变量,也可以不赋给任何变量,直接作为表达式的一部分使用。

例 1.6 改写例 1.5。

```
#include "stdio.h"
main()
{ putchar(getchar());
}
```

例 1.6 程序中, getchar()函数作为 putchar()函数的输出变量使用,输出结果与例 1.5 相同。

1.4.3 格式化输入输出函数

格式化输入输出函数 scanf()和 printf(),也是标准输入输出库函数。它们可以在程序中直接引用,不需要进行编译预处理说明,标准输入输出库函数中只有 scanf()和 printf()是例外的。

格式化输入输出库函数,是按指定的格式完成输入输出操作的。

1. 输出函数 printf()

(1) printf()的作用:将用户指定的输出项按规定格式送到标准输出设备上。

(2) printf()的一般形式:

printf("输出格式",输出项表);

其中,输出项表由若干个输出项组成,当输出项的个数多于一个时,输出项之间用逗号分隔。输出项可以是常量、变量或表达式。

输出格式由普通字符、格式说明和换码序列字符三部分组成。

普通字符是指需要按原样输出的字符。例如:

```
printf("v=%6.2f\n",v);
```

其中,v 是输出项;v=是普通字符,它将原样输出;%6.2f 是格式说明;\n 是换码序列字符。

格式说明由%、附加格式字符(例如 6.2)和格式字符(例如 f)组成,它规定了输出项的输出形式。常用的格式字符见表 1-2 所示,常用的附加格式字符见表 1-3 所示。

表 1-2 格式字符

格式字符	功 能	说 明
d	以带符号的十进制形式输入输出整数	正数不输出符号
o	以八进制无符号形式输入输出整数	不输出前导符 0
x	以十六进制无符号形式输入输出整数	不输出前导符 0x
c	以字符形式输入输出单个字符	
s	输入输出字符串	输入时,以非空字符开始,以第一个空白字符结束
f	以小数形式输入输出单精度、双精度数	隐含输出 6 位小数

续表

格式字符	功 能	说 明
e	以标准指数形式输入输出单精度、双精度数, 数字部分小数位数为 6 位	
u	以无符号十进制形式输出整数	
g	选用 %f 或 %e 格式中输出宽度较短的一种格式, 不输出无意义的 0	

表 1-3 附加格式字符

格 式 字 符	功 能 及 说 明
l	用于输入输出长整型整数, 可加在 d、o、x、u 之前
m	输出数据占 m 位
.n	对实数输出 n 位小数, 对字符串截取 n 个字符
0	数据左边补 0
+	输出后的数据右对齐
-	输出后的数据左对齐

换码序列字符是以反斜杠“\”打头后跟一个字母或数字组成的, 它的作用是输出控制代码和特殊字符。例如, \n 表示输出一个回车换行。详细内容将在第 2 章介绍。

(3) printf() 的使用说明:

① 格式字符必须用小写英文字母。例如, %f 不能写作 %F。

② 如果想原样输出一个 % 号, 则应在输出格式中用两个 % 表示。例如:

```
printf("%f%%", 0.777777);
```

的输出结果如下:

```
0.777777%
```

③ 选用格式字符“e”时, 以标准指数形式输出单精度、双精度数。所谓标准指数形式是指小数点前必须有并且只能有一位非零数字。例如:

```
printf("%e\n", 567.123);
```

的输出结果如下:

```
5.67123e+02
```

④ 附加格式字符“m.n”用于控制输出数字的位数。表示输出数字共占 m 位, 其中含有 n 位小数。如果数值长度小于 m 则左端补空格。例如:

```
printf("%6.3f\n", 34.56);
```

的输出结果如下:

```
34.560
```

例 1.7 printf() 的应用。

```
main()
{ int a,b;
  a=10,b=25;
```

```
printf("a=%d b=%d\n",a,b);
printf("a+b=%4d\n a-b=%4d\n",a+b,a-b);
}
```

程序的运行结果如下:

```
a=10 b=25
a+b= 35
a-b= -15
```

例 1.8 printf()应用。

```
main()
{ int a,b;
  a=100;b=-200;
  printf("%+10d,%+10d\n",a,b);
  printf("%-10d,%-10d\n",a,b);
  printf("%010d,%010d\n",a,b);
}
```

程序的运行结果如下:

```
      +100,      -200
100      , -200
0000000100, -000000200
```

例 1.8 程序中, 选用“+”时, 输出数据右对齐, 左边补空格, 并输出符号; 选用“-”时, 输出数据左对齐, 右边补空格; 选用“0”时, 输出数据右对齐, 左边补 0。

2. 输入函数 scanf()

(1) scanf()的作用: 从标准输入设备上按一定格式接收一批数据, 并赋给指定的变量。

(2) scanf()的一般形式:

```
scanf("输入格式",输入项地址表);
```

其中: 输入项地址表由若干个地址组成, 当地址个数多于一个时, 地址之间用逗号分隔。地址可以是变量的地址或字符串的首地址。C 语言规定: 用“&”加上变量名表示变量的地址。例如, &a 表示变量 a 的地址。

输入格式中一般只使用格式说明, 格式说明规定了输入项的输入格式, 常用的格式说明见表 1-2 所示。例如:

```
scanf("%d%d",&y,&k);
```

其中, %d 是格式说明, &y 和 &k 表示变量 y 和 k 的地址。该函数的功能是从键盘输入两个十进制数, 分别赋给变量 y 和 k。

(3) scanf()函数的使用说明:

① 在使用多个输入项的 scanf()函数时, 键盘输入的各项数据之间可以用空格、TAB 或回车作为分隔符。

② 如果输入格式中出现普通字符, 则在输入数据时必须输入与这些字符相同的字符。例如:

```
scanf("a=%d,b=%d",&a,&b);
```

键盘输入时应按如下形式:

a=3,b=4<CR>

其中,“a=”,“,”和“b=”是要求用户必须原样输入的字符串。

③ 在用%c 格式输入字符时,字符间不能加分隔符。例如:

```
scanf("%c%c%c",&a1,&a2,&a3);
```

当键盘输入

abc<CR>

时,字符“a”赋给变量 a1,字符“b”赋给变量 a2,字符“c”赋给变量 a3。而当键盘输入:

a b c<CR>

时,字符“a”赋给变量 a1,字符“ ”(空格)赋给变量 a2,字符“b”赋给变量 a3。

例 1.9 从键盘任意输入两个整数,输出其和值与差值。

```
main()
{
    int i,j;
    printf("输入第一个整数 i:");
    scanf("%d",&i);
    printf("输入第二个整数 j:");
    scanf("%d",&j);
    printf("i+j=%d\n",i+j);
    printf("i-j=%d\n",i-j);
}
```

例 1.9 程序是以人机对话的方式向变量 i 和 j 赋值,运行结果如下:

输入第一个整数 i: 25<CR>

输入第二个整数 j: 12<CR>

i+j=37

i-j=13

其中,有底划线的部分是用户从键盘输入的字符。

1.5 C 语言程序的编辑及运行

本节主要介绍 C 语言程序的开发步骤和 Turbo C 集成环境的使用。

1.5.1 C 语言程序编辑及运行的一般步骤

C 语言采用编译方式将源程序翻译为二进制代码,其一般过程分为四个步骤:

1. 程序编辑

程序编辑包括以下内容:

(1) 将源程序输入到计算机内存;

(2) 修改源程序,并将修改好的源程序存入磁盘。源程序文件的扩展名为 .C。

2. 程序编译

C 语言编译程序对源程序进行语法检查,如果发现语法错误时,将在屏幕上显示出错信息。此时,应重新编辑源程序,编辑结束后再重新编译。正确的源程序文件经过编译后,