中等职业教育计算机系列教材

陈宇姣　徐　卉 编著

C程序设计简明教程

机械工业出版社
China Machine Press

《中等职业教育计算机系列教材》编委会

序　言

深化教学改革，提高中等职业学校教育的教学质量，既是落实“十五”大提出的科教兴国战略目标的重要举措，也是职业教育自身发展的基本需要。教育部职教司有关人士曾在教育部职业教育改革座谈会上指出：“按照专业教育的要求，建立新的课程体系和与之配套的教材系列是职业教育改革的核心。”根据这个教改思想和教学要求，我们特地精心编写了这套中等职业教育计算机系列教材。

经过多年的办学实践，我们发现以往我们实施的职教课程是以学科为中心设计的。这种学科课程模式虽有它的长处，但用之于职业教育，其弊端十分明显：它不利于学生职业能力的形成，其专业教学不能很好地适应人才市场对学生的需求，特别是不能满足不断涌现的新行业、新工种的社会需求，同时也不符合当前接受中等职业教育学生知识水平的实际情况。为此，一批长期工作在职中、中专教学第一线的老、中、青计算机骨干教师，以他们饱满的热情、丰富的教学经验、积极的探索精神，以国家教育部“九五”重点课题“面向21世纪职业学校课程与专业教材体系的研究与实验”子课题阶段性成果为依托，以成都市电子计算机职业学校已形成的《面向21世纪“宽基础，活模块”课程改革方案》为指导，为我国中等职业教育(包括职中、技校、中专等)广大师生，提供了一套“立意科学、体系新颖，选材得当、结构合理，简明易懂、适合教学”的计算机系列教材。这套系列教材分为“宽基础”教材和“活模块”教材两大部分。其课程设置，宏观上体现了以实践能力为本位的教育观和新时期职业教育培养目标，符合经济“两个根本性转变”对高素质劳动者的需求。微观上，注重基础知识和通用能力的教学，有利于学生的深造和发展；注重多种应岗能力的培养，有利于学生的就业、生存和发展；注重因材施教，学生可以根据自己的兴趣和能力，选择主修模块，有利于学生的主动发展。

整套教材富有特色、注重实用、有利教学、配套成龙，其根本特点和主要优点，可体现并归结为“浅”、“用”、“新”：浅——“语言通俗，内容浅显”，适合中等职业教育学生的学习特点和接受能力；用——“符合实际，知识够用”，能满足社会对中等职业教育学生的知识结构需要和技能素质要求；新——“跟踪发展，技术较新”，能兼顾中等职业教育学生毕业后再学习和再提高的可持续发展需要。

本套教材首批出版以下分册，供各校作为计算机“宽基础”部分可组合的模块化教材使用，适用于各专业的基础课程。

1.《中文WORD 97简明教程》
2.《中文EXCEL 97简明教程》
3.《QBASIC程序设计简明教程》
4.《中文FOXBASE⁺程序设计简明教程》
5.《中文VISUAL BASIC简明教程》
6.《C程序设计简明教程》
7.《PHOTOSHOP简明教程》

8.《计算机软件维护简明教程》

这套计算机系列教材内容上涵盖了计算机基础知识、当前最流行的常用软件知识和基本技能以及公众社会最常用的计算机应用技能。今后，我们还将根据专业模块的课程设置，编写“计算机应用、文秘及电算化”、“计算机程序设计及应用”、“计算机维护管理与网络应用”、“计算机平面设计、动画制作、广告信息处理”等相关计算机专业知识和技能方面的教材。

本丛书的编写出版，得到了机械工业出版社的大力支持，得到了省市职业教育科研部门的指导和帮助。特别鸣谢周启海教授，他花费了大量时间对每一本教材进行了详细审阅，使该系列教材的编写质量有了较大的提高。在该系列教材的编写过程中，我们也参阅了国内外许多专家学者计算机方面的专著。这里，我们谨向他们表示诚挚的敬意。

祝这套新教材的问世和推广能为我国中等职业教育的发展做出积极贡献!

由于我们的水平有限，加之时间仓促，所编教材一定存在许多不足之处，希望选用我们教材的同行和同学给予批评指正。

《中等职业教育计算机系列教材》编委会　主任委员
中国计算机学会职业教育专业委员会常务委员　黄浩军

1999年7月

前　言

C语言近年来在国内外得到了迅速推广和使用。它既具有高级语言的功能，又具有低级语言的功能，是一种可以用于开发系统软件的程序设计语言。C作为一种基础语言，从实用性和现实性两方面，再结合职高计算机后续课程的开设来看，学习和掌握它是十分必要和有用的。

本教材在每章的篇首都提供了参考教学时数，其中包含授课学时和上机实习学时，总学时为一个学期。主要内容包括：C语言基础知识和在Turbo C编译系统下编写C程序的基本方法两部分。

本书面向职业高中和技校及中等专业学校的学生。作者根据在长期职高教学实践活动中的经验，将C语言的基本内容经过合理地组合，力求做到条理清晰、深入浅出，同时自己设计和精选了一些适合该层次学生水平的例题和习题，且全部例题和习题均在Turbo C环境下调试通过。本书着重于讲述语言的使用，同时也通过练习培养学生编写简单程序的能力。全书体现了浅、用、新的特点。

本书的第1、2、5、6、7、8章及附录由陈宇姣编写，第3、4、9章由徐卉编写。在编写过程中，得到了成都市电子职业中学及职教处、专业组的领导和同志们的大力支持和帮助，特表感谢。

由于成书匆忙，定有不完善之处，敬请广大师生批评指正。我们将在积累经验的基础上不断进行修订与补充。

作者

1999年6月

目　录

第1章　C语言的基本知识

[内容提要]

1) C语言的特点。

2) C程序的基本构造。

3) C语言的数据类型。

4) C语言的各种运算符及表达式。

5) Turbo C系统的安装及运用。

[教学要求]

1) 了解C的发展和程序的基本构成。

2) 掌握C的特点和上机实习的步骤。

3) 掌握常量、变量的数据类型及给变量赋值的方法。

4) 掌握C语言中各类表达式的运算及在运算中各类数据间的相互转换。

[课时建议]

讲授学时：6~8课时；

上机学时：2~4课时。

1.1　C语言的发展和特点

1.1.1　历史背景简介

电子计算机自1946年问世以来，随着其应用领域的迅速扩大，硬件和软件都有了长足的发展。作为计算机软件的基础，程序设计语言也得到不断的充实和完善。功能全面、使用方便的程序语言相继问世。

最早的操作系统等系统软件主要是用汇编语言编写的。由于汇编语言依赖于计算机硬件，所以程序的可读性和可移植性都比较差。为了提高可读性和可移植性，就需要找到一种既具有一般高级语言特性，又具有低级语言特性的语言，集二者的优点于一身。美国贝尔实验室的K.Thompson和D.M.Ritchie在开发UNIX操作系统时，迫于需要在1972～1973年由D.M.Ritchie设计出了C语言。后来，C语言多次做了改进，但主要还是在贝尔实验室内部使用。直到1975年UNIX第6版公布，C语言才引起了人们的注意。1977年出现了可移植C语言，1978年第7版的C 语言成了当时的标准，即标准C语言。1983年，美国标准化协会(ANSI)对C语言制定了新的标准，称为ANSI C。到1987年，ANSI又公布了新标准——87 ANSI C，即现行的C语言标准。

本书的内容基本上是以ANSI C为基础。目前微机上广泛流行的C语言编译系统有Microsoft C、Turbo C等，本书的上机实习以Turbo C 环境为基础。

1.1.2　C语言的特点

C语言自问世以来就表现出较强的生命力，它的主要特点如下:

1) 语言简捷、紧凑，使用方便灵活，程序书写形式自由。C 语言只有32个关键字(见附录B)，它们构成了C语言的全部指令。程序主要用小写字母表示，压缩了一些不必要的成分，便于阅读和书写。

2) 运算符种类丰富。除一般的算术运算(+、-、*、/)与逻辑运算(&&、||、！)外，还可以实现以二进制位(bit)为单位的位运算(—、|、～、∧、>>、<<)，并且具有自增、自减和复合运算功能(a++、b--、+=、-=、*=、/=等)。

3) 数据类型(结构)丰富。基本类型有整型(int)、实型(float)、字符型(char)，除此之外还具有构造类型，如数组、指针、结构体、共用体等。用它们可实现各种复杂的数据结构，因此，C语言具有很强的数据处理能力。

4) 是一种结构化程序设计语言。即程序的逻辑结构可以用顺序(图1-1)、选择(图1-2)和循环(图1-3)三种基本结构组成。具有诸如if…else、for、do…while、while、switch…case等结构化语句，便于采用由顶向下、逐步细化的结构化程序设计技术。因此C程序具有容易理解、便于维护的优点。

三种结构分别用传统流程图和N-S结构化流程图描述。

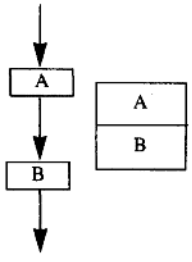

图 1-1

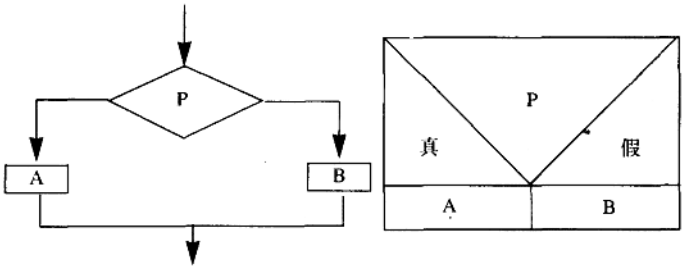

图 1-2

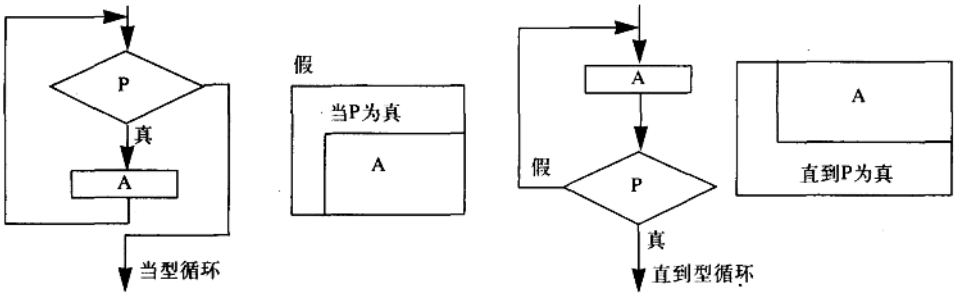

图 1-3

5) 允许直接访问物理地址，能进行位操作，能实现汇编语言的部分功能，可以直接对硬件进行操作。因此C既具有高级语言的功能，又具有低级语言的许多功能，既是成功的系统描述语言，又是通用的程序设计语言。

6) 生成的目标代码质量高。用C语言编写的程序，其代码效率可与汇编语言媲美，同一问题仅比汇编语言写的代码少10%~20%。

7) 具有较高的可移植性。C语言的语句中，没有依赖于硬件的输入与输出语句，程序的输入与输出是通过调用系统提供的输入与输出函数实现的。因此，C程序本身并不依存于机器硬件系统，从而可在不同机种间实现程序的移植。

总之，C语言的优点很多，但也有一些不足。如语法限制不太严格，因此不太安全；运算符优先级太多，不便于记忆等。这就要求使用C语言的人对程序设计更熟练一些。能熟练地用C语言编写程序会感到限制少、灵活性大、功能强。它的突出优点吸引了国内外越来越多的人使用和研究它。优秀的C语言版本和配套工具软件不断涌现。

1.2 简单的C程序介绍

1.2.1 源程序和书写格式

我们先看下面一个简单的C程序，然后从中分析C语言的程序特性。

例1.1

```
main( )                                  /*主函数*/
{int a,b,c;                              /*定义变量*/
scanf("%d,%d",&a,&b);                    /*输入变量a和b的值*/
c=max(a,b);                              /*调用max函数，将得到的值赋予c*/
printf("a=%d\nb=%d\n\max=%d",a,b,c);     /*输出c的值*/
}
int max(x,y)                             /*定义max函数，函数值为整型，x、y为形式参数*/
int x,y;                                 /*对形参x、y作类型定义*/
{int z;                                  /*max函数中用到的变量z，也要加以定义*/
if (x>y) z=x;
else z=y;
return(z);                               /*将z的值返回，通过max带回调用处*/
}
```

这个程序的目的是显示输出a、b两个整数中的最大值。

通过程序我们可以看出：

1) 程序是由函数构成的。函数是构成C语言程序的基本单位，即C语言程序由一个或多个函数组成，组成函数的若干函数中有且只能有一个名为main的主函数。C的函数相当于其它语言中的子程序，所以可以说C是函数式的语言。C的库函数十分丰富，Turbo C提供了三百多个库函数，需要时直接调用即可。在后续章节中我们将陆续详细介绍库函数。

2) 各函数的位置无关紧要。程序会自动从main函数开始执行，通常我们总是把main函数放在其它函数的前面。

3) C程序书写格式自由。一行内可以写几个语句，一个语句也可以写在多行上，用“\”作续行符。如例1.1的第5行和第6行。

4) 程序的每个语句后必须有一个分号“；”。如例1中的int a，b，c;

5) 可以用/*……*/对C程序中的任何部分作注释。注释可为若干行，但不允许嵌套。它使程序变得清晰，能帮助我们阅读和理解。一个好的、有价值的源程序都应加上必要的注释，以增加程序和可读性。

1.2.2 函数的基本形式

C语言的函数由如下两部分组成。

1. 函数的说明部分

这部分包括函数名、函数类型、参数名、参数类型。

如例1中max函数的说明部分:

int max(x, y)——►函数类型 函数名 函数参数

int x, y;——►参数类型 参数名

一个函数名后面必须跟一对圆括弧，函数参数可以没有，如main( )。

参数类型的说明也可以放在圆括号内，如 int max(int x, int y)。

2. 函数体

即大括号{……}内的部分。如果一个函数内有多个大括号，则最外层的一对为函数体的范围。

这部分由两部分组成:

1) 变量定义。如例1.1中的 int a, b, c; 。

2) 执行部分。由若干个语句组成，如例1中main函数中变量定义下面的三行。

注意 在某些情况下可以没有变量定义部分，甚至还可以没有执行部分。

如 donull( )

{ }

它是一个空函数，什么也不做，但这是一个合法的C程序。

1.3 常量

前面我们提到C的数据类型相当丰富，包含基本类型和构造类型等，但从使用的角度来看，可分为常量和变量。

在程序的运行过程中，其值不能被改变的量称为常量。常量包括整型常量、实型常量、字符型常量。

1.3.1 整型常量

C语言中的整型常量(常数)有三种表示形式:

1) 十进制整数，如23、-456、0等。

2) 八进制整数，如0123、-0456等，八进制整数以0(零)开头，0123等于$(123)_8$，-0456等于$(-456)_8$。

3) 十六进制整数，如0x123、-0x456等，十六进制整数以0x开头，0x123代表$(123)_{16}$，-0x456代表$(-456)_{16}$。

三种表示形式的使用如下例:

例1.2

```
main( )
{int x=123, y=0123, z=0x123;
printf("%d %d %d\n",x, y, z);   /*%d是printf函数的十进制输出格式*/
}
```

运行结果: 123 83 291

1.3.2 实型常量

实型常量也叫做实型数据，在C语言中有两种表示形式:

1) 十进制数形式。它由数字和小数点组成，书写时小数点不可缺省。如45.34、-0.129、0.0等都是合法的。

2) 指数形式。它用E或e来代表“ × 10”，如26e3和0.78E-5分别代表26×10^3和0.78×10^{-5}。

注意，字母E或e之前必须有数字，指数部分必须为整数，如e、e5、26e0.3 等都是非法的。

1.3.3 字符常量

字符常量是用单引号引起来的单个字符，如'B'、'b'、'8'、'*'、' '等都是字符常量，其中'B'与'b'代表不同的字符常量。

除此之外，C还允许使用一些特殊的字符常量，称为“转义字符”，如表1-1所示。它们用来代表一些非显示字符，如换行、回车等，主要用在printf函数中。

表 1-1

字符形式	功 能
\n	换行
\t	横向跳格(即跳到下一个输出区)
\v	竖向跳格
\b	退格
\r	回车
\f	走纸换页
\\	反斜杠字符"\"
\'	单撇号字符
\a	报警
\ddd	1~3位8进制数所代表的字符
\xhh	1~2位16进制数所代表的字符

例1.3

```
main( )
{printf("ab\'E\'␣ \t\bcd\n");
printf("efg␣ ");
}
```

例1.3中用printf( )函数直接输出双引号中的各个字符,其中第一个输出语句中的“\”,作用是输出一个单引号，“␣”的作用是输出一个空格(注意，在显示屏和打印机上，空格是不会显示出来的，只是留空一格)，“\t”的作用是跳到下一输出区(注：IBM-PC及其兼容机的一个输出区为8个字符位,即下一输出位为第9列)，“\b”的作用是向后退一格，“\n”的作用是换到第二行。

程序运行结果如下:

```
ab'E' ␣␣ cd
efg␣
```

1.3.4 字符串常量

1. 字符串常量

C语言除了允许使用字符常量外，还允许使用字符串常量。字符串常量是一对双引号括起

来的字符序列。

如：“I am a student”、“2.13”，“a”等，都是字符串常量。

2. 字符串常量与字符常量的区别

C规定：每一个字符串的结尾，系统都会自动加一个字符串结束标志“\0”，以便系统据此判断字符串是否结束。“\0”代表空操作字符，它不引起任何操作，也不会显示到屏幕上。例如，字符串“I am a student”在内存中存储的形式如下：

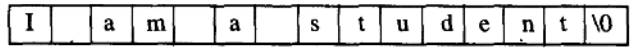

图 1-4

它的长度不是14个，而是15个，最后一个字符为‘\0’。但输出时不输出，系统在遇到它后就停止输出。注意，在写字符串时不能加上“\0”。

所以，字符串“a”与字符‘a’是不同的两个数常量。前者由字符‘a’和‘\0’构成，而后者仅由‘a’构成。

1.3.5 符号常量

用一个标志符代表一个常量称为符号常量，即标识符形式的常量。但是它必须先用#define加以定义。

如：
```
#define PRICE 30
main( )
{ int num=10, total;
total=num*PRICE;
printf("total=%d", total);
}
```

程序中用#define命令行定义PRICE代表常量30，在后面的程序中凡出现的PRICE都代表30，与常量的用法完全相同。(有关#define的用法在第9章详细讲解)

程序的运行结果为total=300

注意 习惯上，符号常量名用大写。

1.4 变量

在程序运行中，其值可以改变的量称为变量。

和其它语言一样，C语言中也有各种类型的变量。从用户的角度来说，变量代表某个具体的数字、字符等。从系统的角度来说，变量在计算机内存中占用了一定大小的存储空间，它是计算机高级语言中用以存放各种数据的机构。

在C语言中，我们把变量分为整型变量、实型变量(单精度、双精度)、字符变量、指针变量及在后面的章节中我们将详细介绍的数组、结构体、共用体等构造类型变量。

标志变量名、符号常量名、函数名、数组名、类型名、文件名的有效字符序列称为标志符。它们只能由字母、数字和下划线三种字符组成，且第一个字符必须为字母或下划线(注意C语言要区分大小写字母)。

如：total、sum、student name、1ab 是合法的变量名，

4seed、class.2、#ab 是不合法的变量名。

1.4.1 整数型变量

1. 整数型变量的分类

整型变量通常分为四类：

1) 整型(int)。

2) 短整型(short或short int)。

3) 长整型(long或long int)。

4) 无符号型 { 无符号整型(unsigned int)。
无符号短整型(unsigned short)。
无符号长整型(unsigned long)。

其中，无符号型变量只能存放不带符号的整数，如543、2314等，而不能存放负数，如-654、-1329。

表1-2列出了IBM PC及其兼容机上各类整型变量的字节长度和取值范围。

表 1-2

数据类型	字节长度	数的范围	
int	2	-32768~32767	即-2^{15}~$(2^{15}-1)$
short[int]	2	-32768~32767	即-2^{15}~$(2^{15}-1)$
long[int]	4	-2147483648~2147483647	即-2^{31}~$(2^{31}-1)$
unsigned[int]	2	0~65535	即0~$(2^{16}-1)$
unsigned short	2	0~65535	即0~$(2^{16}-1)$
unsigned long	4	0~4294967295	即0~$(2^{32}-1)$

2. 整型变量的定义及赋值

C程序中所有用到的变量都必须在使用之前指定其类型，即定义。

例1.4

```
main( )
{
int a, b;                    /*定义a, b为整型变量*/
long c, d;                   /*定义c, d为长整型变量*/
unsigned e, f;               /*定义e, f为无符号整型变量*/
a=32746; b=10;
c=1534785325; d=10;
e=6546; f=10;
printf("%d, %d\n", a, a+b);  /*%d输出整型数据的格式转换控制符*/
printf("%ld, %ld\n", c, c-d); /*%ld输出长整型数据的格式转换控制符*/
printf("%u, %u\n", e, e*f);  /*%u输出无符号整型数据的格式转换控制符*/
}
```

运行结果：

```
32746, 32756
1534785325, 1534785315
6546, 65460
```

给整型变量赋值时应注意到变量的存储范围：

1) short int、int和long int型变量可保存-32768~+32767范围内的整型常量。

2) long int 型变量可保存-2147483648~+2147483647范围内的整型常量。

3) unsigned型变量可保存0~4294967295范围内的整型常量。

1.4.2 实型变量

实型数据也称为浮点型数据，有单精度和双精度之分。实型变量定义形式如下:

```
float x, y;          (定义x, y为单精度型实数)
double z;            (定义z为双精度型实数)
```

IBM-PC及其兼容机上浮点型变量的字节长度和取值范围见表1-3。

表 1-3

数据类型	字节长度	取值范围	有效位
float	4	1e-38~1e+38	7
double	8	1e-308~1e+308	15

两种类型的实型变量都可用来存放同一个实型常量，只是截取的有效位位数不同。double型比float型精度要高。

例1.5 有下面一段程序

```
float x;
double y;
x=123456.789
y=123456.789
```

其中x为单精度型变量，被赋值为9位，但只能接收前7位有效位，因此最后两位小数不起作用。而y为双精度型变量，它能接收全部9位数字并存储起来。

1.4.3 字符变量

1. 字符数据的存储形式及变量定义

一个字符变量用来存放一个字符(注意，不是一个字符串)，在内存中占一个字节。但是在存放时并不是存放的字符本身，而是将字符所对应的ASICII码放到存储单元之中。由于字符型数据与整数的存储形式相同，因此，C语言规定：字符型数据和整型数据之间可以通用。字符数据可以像整型数据那样使用，可以用来表示特定范围内的整数。字符型变量定义形式如下:

```
char c1, c2;   /*定义c1, c2为字符型变量*/
```

字符数据在IBM-PC及兼容机上的字节长度和取值范围见表1-4。

表 1-4

数据类型	字节长度	取值范围
char	1	0~255的整数或所有对应字符

2. 字符数据的输入、输出及算术运算

字符数据既可以以字符形式输出，也可以以整数形式输出。以字符形式输出时，首先将存储单元中的ASICII码转换成相应字符，然后输出。以整数形式输出时，直接将ASICII码作为整数输出。

例1.6

```
main( )
{char c1, c2;
c1=97; c2=98;
printf("%c, %c\n", c1, c2);      /*将97和98转换成字符'a'和'b'输出*/
printf("%d, %d\n", c1, c2);      /*直接将ASICII码值97和98作为整数输出*/
}
```

运行结果:

```
a, b
97, 98
```

例中，c1和c2为字符型变量，并将整数97和98分别赋予c1、c2，由于ASICII码97和98对应的字母是a和b，因此,第三行相当于赋值语句c1='a'; c2='b'。

例1.7

```
main( )
{char c1='a';
int c2='b';
c1=c1-32; c2=c2-32;
printf("%c %d\n", c1, c1);
printf("%c %d\n", c2, c2);
}
```

运行结果:

```
A 65
B 66
```

例中利用小写字母比它对应的大写字母ASICII值大32的规律，通过减运算进行字符转换，C语言这种对字符数据的处理方式使程序设计增大了自由度，具有很大的优越性。

注意 在C语言中没有专门的字符串变量，字符串如果需要存放在变量中，则要用一个字符型数组来存放，这将在第4章中详细介绍。

1.4.4 指针变量

要讲指针变量，首先要知道什么是地址。在C语言中，每一个变量都有一个存储单元来保存该变量的值，当计算机处理这个变量时，就到相应的存储单元去取值。每个存储单元在计算机中都用一个序号来表示，这个序号就称为存储单元的地址。每个变量都分配有一个或几个存储单元，这些存储单元的首地址就被称为“变量的地址”。

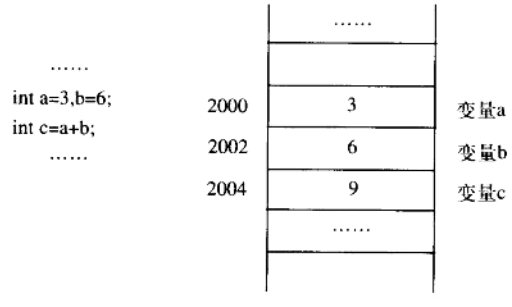

图 1-5