C 语言程序设计教程

李书涛　编

北京理工大学出版社

(京)新登字 149 号

内 容 简 介

C 语言是一种结构化的通用程序设计语言，是目前国际上应用非常广泛的新型的现代主流程序设计语言。

本书以最新版本美国标准 C 语言(87 ANSI C)为基础，全面系统地介绍了 C 语言的基本概念、语法规则及 C 语言程序设计技术。

本书是学习 C 语言程序设计的基础教程，内容丰富、逻辑性强、循序渐进、深入浅出，并结合大量的例题分析，讲解了结构化程序设计方法和编程技巧，便于读者自学。另外，每章附有习题，书末附录给出了 C 语言程序设计常用的一些资料，具有较强的实用价值。

本书可作为大专院校计算机语言课教材，也可供计算机应用培训班和有关人员参考使用。

C 语言程序设计教程

李书涛 编

*

北京理工大学出版社出版发行

各地新华书店经售

秦皇岛市卢龙印刷厂印刷

*

787×1092 毫米 16 开本 19.25 印张 479 千字

1993 年 2 月第一版 1993 年 2 月第一次印刷

ISBN 7—81013—662—3/TP・75

印数：1—8000 册 定价：10.00 元

目　　录

第一章　C语言概述

§1.1　C语言的发展与特点

1.1.1　C语言的发展

早期的操作系统等系统软件，主要是用汇编语言编写的，它依赖于计算机硬件，程序的可读性和可移植性都很差。若改用高级语言来编写，又难以实现汇编语言能直接对硬件进行操作的某些功能。为此，人们开始寻求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。

1960年出现的ALGOL 60是一种面向问题的高级语言，它离硬件较远，不适宜用来编写系统程序。1963年英国的伦敦和剑桥大学在ALGOL 60的基础上推出了CPL(Combined Programming Language)语言，它接近硬件一些，但规模较大，难以实现。1967年英国剑桥大学的Martin Richards对CPL语言作了简化，推出了BCPL(Basic Combined Programming Language)语言。它是一种没有数据类型，或者说只有一种数据类型——机器字的单一数据型语言。1970年美国贝尔实验室的Ken Thompson以BCPL语言为基础，又作了进一步简化，设计出了很简单且很接近硬件的B语言(取BCPL的第一个字母)，并在PDP-7机上实现了用B语言写的第一个UNIX操作系统。1971年又在PDP-11/20机上实现了B语言，并写了UNIX操作系统。但B语言仍是单一数据型语言，过于简单，功能有限。1972年美国贝尔实验室的D. M. Ritchie在PDP-11机上实现了C语言。他以B语言为基础，引进多种基本数据类型：字符、整数和浮点数，并导出数组、指针、结构、联合和函数，既保持了BCPL和B语言精练、接近硬件的优点，又克服了它们过于简单、数据无类型等缺点。1973年，K. Thompson和D. M. Ritchie合作用C语言重写了UNIX操作系统，实现了UNIX第五版。它比原先的版本更易于理解、修改和扩充，并增加了多道程序设计功能，开创了UNIX系统发展的新局面。图1.1反映了C语言的发展历史。

从C语言发展历史来看，C属于ALGOL语言族系。C语言在发展过程中与UNIX相辅相成，C语言开创了UNIX的新局面，使其获得了巨大成功；UNIX促成了C语言的大发展，使其得到了迅速推广，从而形成一对繁荣与共的孪生兄弟。1978年以后，C语言逐渐独立于UNIX系统，独立于PDP—11机而蓬勃发展。以1978年发表的UNIX第七版中的C编译程序为基础，Brian W. Kernighan和Dennis M. Ritchie(合称K&R)合作出版了名著《The C Programming Language》，由它产生了C语言版本的基础，称为标准C。1983年，美国国家标准化协会(ANSI)将标准C作了扩充和发展，制定了新的标准，称为ANSI C。1988年K&R按照ANSI C标准又重写了他们的名著。目前人们常将1978年的标准C称为旧标准，将ANSI C称为新标准。随着C的发展，现在，C语言已风靡全世界，成为世界上应用非常广泛的新型的现代主流程序设计语言，成为微、小、超小、大、超大和巨型等各类计算机共同使用的语言。广泛应用于系统软件(如操作系统、编译系统等)、应用软件(如图形处理)、数据处理(如企业管理)以及数值计算等各个领域。

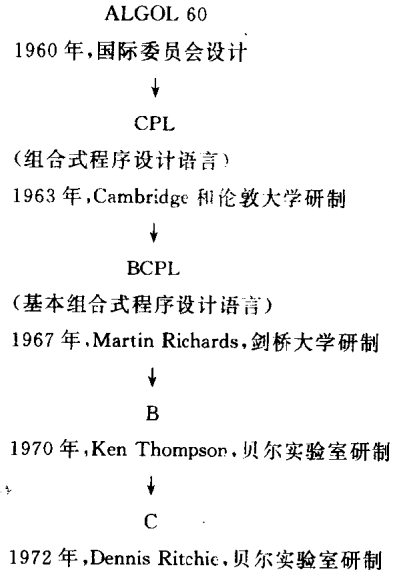


图1.1 C语言发展历史

C语言之所以起名为C，有人说因其是由BCPL经B演化而成，所以称C；另有人说这是按字母A、B、C、D……的顺序，B后该是C。C语言的开发者D. M. Ritchie对这两种说法均不置可否。C之后的发展是称P或D，也任人猜测。不论将来怎样，当前从它仅一个字母C的名字，常令人联想到它的简洁性。

1.1.2 C语言的特点

C语言之所以能具有强大的生命力，成为国际上公认的最重要的少数几种通用程序设计语言之一，固然与它产生的环境及历史背景有关，但起决定作用的是它自身的特点。概括起来，C语言的特点是：简洁、灵活，表达能力强，代码质量高，结构化程序，可移植性好。具体说可分成以下几点：

(1) 语言简洁。C是一种小型语言，在程序设计中，小即是美。C虽小却被公认为是比较强有力的语言。经精心选用，它总共只有32个关键字，9种控制语句，压缩了一切不必要的成分，基本组成部分紧凑，表示方法简洁，使用一些简单、规整的方法就可以构造出相当复杂、功能很强的数据类型、语句和程序结构。如用{、}代替begin、end作复合语句或函数体的“括号”；用++表示加1；--表示减1；运算符省写等。另外，从语言内部实现角度看，C语言本身不提供输入/输出机构，即没有在一般高级语言中常用的read、write语句和固定的文件存取方式，这些高级机构都通过显式函数调用来实现。

(2) 表达方式灵活实用。C语言提供了多种运算符和获得表达式值的方法，对问题的表达也可通过多种途径得到，使用户在程序设计中有更大的主动性；语法限制不太严格，程序设计自由度大，如对数组下标越界不作检查，对变量类型使用灵活(整型量与字符型数据及逻辑型数据可以通用)等；另外语言格式自由、限制少，编写程序较自由；程序以小写字母为基础，小写字母易读易写等，充分体现出C语言灵活、实用、方便的特点。

(3) 表达能力强。C语言有丰富的数据结构和运算符。包含了现代化语言所要求的各种数据结构：整型、实型、字符型、数组类型、指针类型、结构类型、联合类型等，能用来实现各种复

杂数据结构(如链表、树、栈等)的运算。运算符有34种,包含的范围很广,灵活使用各种运算符,可以实现其它高级语言难以实现的运算。C语言能直接访问物理地址,能进行位(bit)操作,能实现汇编语言的大部分功能,可以直接对硬件进行操作。因此,C语言兼有高级语言和低级语言的许多优点,故有人称C为"高级语言中的低级语言"或"中间语言",它既可用来编写系统软件,又可用来开发应用软件,成为一种成功的通用程序设计语言。

(4) 语言生成的目标代码质量高。相对汇编语言而言,许多高级语言生成的代码质量很低,所以,迄今汇编语言仍是编写系统软件的主要工具。但是,许多试验表明,针对同一问题,用C语言编写的程序,生成代码的效率仅比用汇编语言写的代码低10～20%。由于用C语言描述问题比用汇编语言描述编程迅速,工作量小,可读性好,易于调试、修改和移植,而在代码质量上可与汇编语言相媲美,因此,C语言迅速成为人们进行程序设计和软件开发得心应手的工具。

(5) 结构化程序设计。C语言是一种结构化语言,提供了编写结构化程序所需的基本控制流结构语句,如if…else、switch、while、do…while、for等;用函数作为程序设计的基本单位,实现程序的模块化;包含程序的源文件,可以分割为多个源文件,可以分别对各个源文件进行编译,再通过连接得到可执行的目标程序文件;提供多种存储属性,使得数据可以按需要在相应的模块中共享或隐藏。

(6) 可移植性好。汇编语言因为依赖于机器硬件,所以根本不可移植。而一些高级语言,如FORTRAN等的编译程序也不可移植,它们从一种机器搬到另一种机器,基本上都要根据国际标准进行重新编写。而C语言目前在不同机器上出现,大部分都是C语言编译移植得到的。统计资料表明,不同机器上的C编译程序80%的代码是公共的。C语言的编译程序便于移植,就使一个环境上用C语言编写的许多程序,从该环境不加或稍加改动就可以搬到另一个完全不同的环境上运行。

C语言的优点很多,但和其它程序设计语言一样,它也有其弱点,如运算符的优先级较多,有些还与常规约定不同,不便记忆;某些语法部分不易用形式化方法进行描述;各种C语言版本之间略有差别,缺乏统一的标准;C语言不是强类型的语言,它强调灵活、高效的同时,在一定程度上牺牲了某些安全性,如类型检验太弱,转换比较随便等。因此,C语言对程序设计员提出了较高的要求,尤其在使用C语言的某些高级手段时更是如此。但是,C语言的优点远远超过了它的弱点,这些优点使C语言具有强大的吸引力。实际经验表明,程序设计人员一旦接触了这种语言,并且有一定程序设计经验后,就会对它爱不释手。

§1.2 C语言程序结构

C语言程序一般由若干个函数组成。函数是完成某个算法的一个程序段,它是C语言的基本构成单位。组成一个程序的若干个函数可以保存在一个或几个源程序文件中,这些文件都以.c为文件扩展名。一个程序必须有且只能有一个名为main的函数,它是该程序的主函数,运行C程序时,总是从main函数开始执行。

函数由函数头和函数体两部分构成。函数头主要是函数名、函数标志和参数说明;函数体中是一组语句,语句可分为说明语句和执行语句两类。

下面是一个小而简单的C程序的例子。

例 1.1 已知 a=1,b=2,写一个程序,计算并输出和值:sum=a+b+c。

```
/*
  * file:simple.c
  * sum=a+b+c/
  */
main(  )                    /* 主函数 */
{ int  a,b,c,sum;           /* 定义变量 */
  a=1;b=2;                  /* 赋值 */
  scanf("%d",&c);           /* 输入 */
  sum=a+b+c;                /* 求和 */
  printf("sum=%d\n",sum);   /* 输出 */
}
```

上述是一个完整的 C 程序。其中,第一行至第四行以"/*"开头到"*/"结尾之间的内容是一个注释,它可在一行写完或分多行写完,可写在程序的任何部位。注释的作用是帮助阅读和理解程序,编译时,注释行被忽略掉,即它不产生代码行。注释在 C 程序中是很有用的,一个好的程序设计者应该在程序中多使用注释来说明整个程序的功能和注意事项,以及有关算法等。注释可以用英语、汉语、汉语拼音或数学式子等任意形式表示,关键是要简洁明了。

第五行是函数头。其中 main 是函数名,它表明本函数是该程序的主函数。紧跟其后的括号"( )"为函数标志,告诉编译程序这是一个函数。一个函数在其名字之后一定要有一对圆括号。圆括号中的参数可有可无,本函数这里没有参数。

第六行开头的花括号"{"和第十一行的"}"围住了函数体,分别表明函数体的开始和结束,它的作用类似于 PASCAL 中的"begin"和"end"。这里函数体中包含了六个语句。

第六行"{"后为一个说明语句,说明 a,b,c 和 sum 为四个整型(int)变量,它是程序的变量定义部分。

第七行是两个赋值语句,使 a 和 b 的值分别为 1 和 2。

第八行是一个函数调用语句行,调用名叫 scanf 的函数,它是 C 语言提供的标准输入函数,作用是输入 c 的值。scanf 是"输入函数"的名字,圆括号中是它的参数,"%d"是输入的"格式字符串",用来指定输入时的数据类型和格式。这里,"%d"表示"十进制整数类型",&c 中的"&"的含义是"取地址"。这里 scanf 函数的作用是:将键盘打入的一个十进制整数值输入到变量 c 的地址所标志的单元中,也就是输入给变量 c。这种形式与其它语言不同,它相当于 BASIC 语言中的 INPUT c 或 PASCAL 语言中的 Read(c)。

第九行赋值语句,使 sum 的值为 a+b+c,即求和赋值。

第十行也是一个函数调用,printf 是 C 语言提供的标准输出函数,printf 是"输出函数"的名字,函数中双引号内的"sum=%d\n",在输出时,"sum="原样输出,"%d"将由圆括号中最右端的 sum 的值取代,"\n"表示输出后换行。程序运行情况如下:

```
6↓          (输入 6 给变量 c)
sum=9       (输出 sum 的值)
```

本例程序比较简单,它只由一个主函数构成,其中用到了两个函数调用。scanf 和 printf 都是 C 语言提供的标准函数,称为库函数。通常,一个 C 程序除必须有一个主函数外,还可能有

若干个其它函数，它们被主函数调用。

C 语言中的函数等价于 FORTRAN 中的子程序或函数，等价于 PL/1、PASCAL 等中的过程。函数提供了一种方便方法，它把某些计算功能封装在黑盒子里，使用时无需考虑它的内部结构。正确地设计函数可能做到只要知道一个作业做什么就够了，而不需要知道它究竟怎么做。C 语言函数使用简单、方便，执行效率高。在 C 语言程序设计中，尽量用多个小函数构成一个大程序，是一种良好的程序设计风格。下面给出一个求大值程序的例子。

例 1.2 写一个程序，输入 a 和 b 两个数，求出其中较大者的值输出。

```
/*  *  *  *  *  *  *  *  *  *  *  *  *  *  */
/* 程序:1.2                                  */
/* 功能:求大值                               */
/*  *  *  *  *  *  *  *  *  *  *  *  *  *  */
main( )                    /* 主函数 */
{ int a,b,c;
  scanf("%d,%d",&a,&b);
  c=max(a,b);
  printf("max=%d\n",c);
}
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;                 /* 定义形参类型 */
  { int z;                 /* 定义函数中用到的变量 z */
    if (x>y) z=x;
    else z=y;
    return(z);             /* 将 z 的值返回 */
  }
```

本例程序由两个函数构成：主函数 main 和被调用函数 max。max 函数的作用是将 x 和 y 中较大者的值赋给变量 z。返回语句 return 将 z 的值返回给主函数 main。返回值是通过函数名 max 带回到主函数的调用处。

程序中第八行为主函数调用 max 函数，调用时先将实际参数 a 和 b 的值分别传递给 max 函数中的形式参数 x 和 y，执行 max 函数后得到一个返回值 max，主函数通过赋值语句将该返回值 max 赋给变量 c。

第九行输出 c 的值。程序运行情况如下：

```
54,8↓        (输入 54 和 8 给 a 和 b)
max=54       (输出 c 的值)
```

本例第一行至第四行与例 1.1 的第一行至第四行类似，都是注释，用来说明程序的名字和功能，它们的作用相同，但表达的形式不一样。对于一个大系统的开发，它由许多程序组成，其程序名和功能的注释，说明或多或少，形式或繁或简，关系都不大，但要求简洁明了，形式统一。以上只不过是列举两种形式供读者参考。此后本书程序举例为节省篇幅，一般不再写出程序的名字和功能的注释。但在大系统的开发设计中，给每个程序写出程序名字、功能，甚至包含编程时间等统一形式的注释，仍为一种程序设计的良好风格。

综上所述，归纳得C语言程序的结构如下：

(1) C语言程序的基本单位是函数。一个C程序由一个或多个函数构成，其中必须有且只能有一个名为main的主函数。主函数可以调用其它函数，被调用的函数可以是系统提供的库函数，也可以是用户自己编写的函数。C的函数库十分丰富，标准C提供一百多个库函数，Turbo C和MS C 4.0提供三百多个库函数。主函数与被调函数在程序中的位置，不论前后，可以任意放置。

(2) 一个函数由函数头和函数体两部分构成。每个函数都有相同的形式，如：

```
函数名(参数表)
参数说明
{
  说明语句
  执行语句
}
```

① 函数头。又称函数说明部分。它包括函数名，函数标志(一对圆括号)，参数表(函数形式参数名)和参数说明(定义形式参数类型)。其中前两项必须有，后两项可有可无；函数名还可包含函数类型、函数名字等。

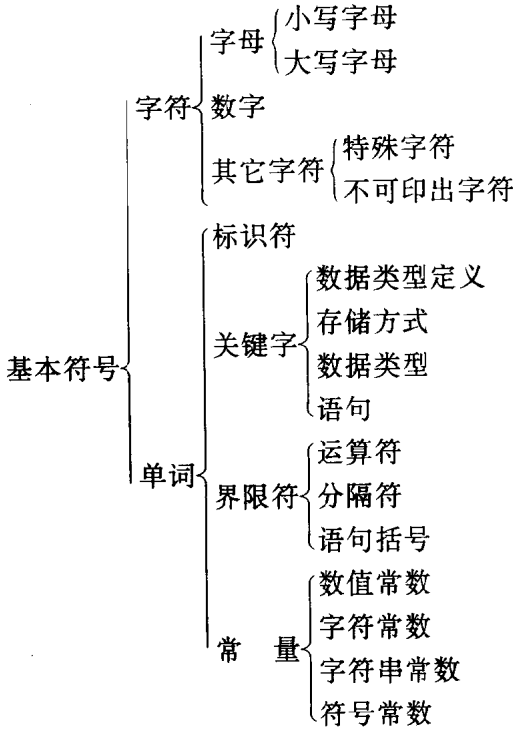
② 函数体。由一对花括号括起来的若干语句组成。通常这些语句分为两类：一类为说明语句，又称变量定义，作用是定义函数中用到的变量；另一类为执行语句，又称执行部分，作用是完成一定的算法处理。其中有些函数可以没有变量定义部分，但常有若干执行语句构成执行部分。特殊情况下，可以既无变量定义也无执行部分，只有一对花括号构成一个空函数体，它与函数头一起组成一个空函数，即什么也不做，这也是合法的。

(3) C语言程序是用小写字母按自由格式书写的程序。通常一个语句写成一行，也可写成几行，一行内也可写多个语句。语句没有行号，也没有象FORTRAN或COBOL那样严格规定书写格式的限制。但是每个语句最后必须有一个分号，该分号是语句的终止符，它属于语句的一个组成部分，即使是程序的最后一个语句也应包含分号，这是与PASCAL不同的。

(4) 可以用/*……*/对C程序中的任何部分作注释。对程序的关键部位写上简明必要的注释，是一种良好的程序设计风格。

§1.3 C语言的基本符号

如前所述，C语言程序的基本单位是函数，函数由语句组成，而语句又由一系列的单词和字符构成。由一个字符序列构成的C语言程序，需要C编译程序编译成目标代码程序才能执行。编译程序首先将源程序字符序列按照C语言规则组成基本词法结构——单词，然后再依据C语言语法规则处理。C编译程序不能编译语法上不正确的程序。符合C语言语法规则的单词和字符组合成C语言的基本符号。C语言的基本符号如下：



1.3.1 字符

C 语言中使用的合法字符有：

小写字母 abc……z

大写字母 ABC……Z

数字 012……9

特殊字符 十 一 _ < = > ! # $ % & * / ? ( ) [ ] { } • , ; : " ' \ ^ ¦ ~

不可印出字符(空白格) 包括空格、换行和制表符。

由上述字符可以按照语法规则构成 C 语言的基本词法结构——单词。

1.3.2 单词

一、 标识符

标识符是用来标识 C 语言程序中一个对象的名字。这些对象有：函数、变量、常量、数组、数据类型、存储方式及语句等。其中有些特殊的标识符具有固定的名字和特定的含义。对于一般的标识符，命名时应注意以下几点：

(1) 一个标识符是一串由字母、数字和下线符(_)组成的字符串。标识符的第一个字符必须是字母或下线符。习惯约定，以下线符领头的标识符作为系统程序专用，程序员最好不要使用以下线符领头的标识符，一般用下线符作为分段符，如 sum __ 1。

标识符的长度按规则可以是任意的，但在实际的 C 语言编译系统中，可识别的标识符长度常限制为 8 个字符。如有标只符 lengthword1 和 lengthword2，编译系统将它们认为是同一标识符，可改为 word1length 和 word2length，系统就可将它们区分为两个不同的标识符。

(2) 标识符大小写字母含意不同，但习惯约定，除符号常量名和宏名标识符用大写字母外，其它如函数名、变量名和数据类型名等标识符都用小写字母。

(3) 标识符命名应简洁明了，含意清晰，便于阅读。如用 sum 表示“求和变量”，用 ave 表示“平均值变量”等。

二、关键字

关键字是一类具有固定名字和特定含义的特殊的标识符，又称为保留字或基字，不允许程序员在程序中将它们另作别用。C 语言常用的关键字有：

数据类型定义：typedef

数据类型：char，double，enum，float，int，long，short，struct，union，unsigned

存储方式：auto，extern，register，static

运算符：sizeof

语句：break，case，continue，default，do，else，for，goto，if，return，switch，while

在某些 C 版本中，还包含下列关键字：

asm，fortran，pascal，ada

下面几个单词严格说来不属于关键字，但是建议程序员把它们看作关键字，不要在程序中随意使用，以免造成混淆，它们是：

define，undef，include，ifdef，ifndef，endif，line

它们通常用在 C 语言的预处理程序中。

Ada 语言具有 62 个关键字。C 的关键字比 Ada 少，比 PASCAL 也少，但却具有强大的功能，这也是 C 的一个特点。

三、界限符

C 语言中起界限作用的字符包括运算符、分隔符和语句括号。

1、 运算符 从广义上说，C 的运算符分隔运算分量，也都是界限符。C 的运算符非常丰富，主要有：

( ) [ ] . → ! ~ ++ -- & * sizeof / %

\+ - << >> >= > <= < == != ^ | && || ?: =

+= -= *= /= ,

其中有些运算符在不同的情况下可能有几种不同的含义，如%、* 及 & 等，这要根据程序中上下文来确定其功能。

2、 分隔符 C 的分隔符主要有：空格符、制表符、换行符和注释。C 语言中注释不可嵌套。

3、 语句括号 C 的语句括号为花括号：{ }

四、 常量

C 语言的常量有：

(1) 数值常量：包括无正负号整常量和整常量(又分十进制常量、八进制常量和十六进制常量)，以及实常量和双精度实常量(又分浮点常量和定点常量)。

(2) 字符常量。

(3) 字符串常量。

(4) 符号常量。

1.3.3 语法图

编写一个高级语言源程序，其是否合法，完全取决于它是否遵守约定的语法规则。通常，表示语法规则的形式有两种方法：巴科斯—诺尔范式即 BNF(Backus－Naur Form)和语法图。BNF 第一次在 1960 年用来描述 ALGOL 60 的语法规则，这种方法的优点是清晰、严谨，在语言形式化描述、编译程序自动生成等研究领域中是一种最为有效的工具。语法图最早盛行于描述 PASCAL 的语法规则，这种方法的长处是直观形象，在语言教学中是一个得力的工具。本书采用语法图形式。如前所述，在 C 语言中，构造一个标识符的语法图如图 1.2 所示。

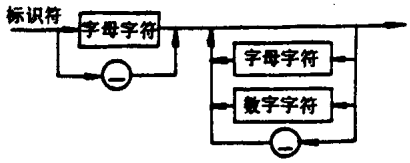


图 1.2　标识符语法图

由此可知，在语法图中，从箭头的入口到出口，沿箭头指向有一条以上的路线，按每条路线所定义的符号、语句或程序都是合法的。

习　　题

1. C 语言的简洁性表现在哪些地方？
2. C 语言的主要用途是什么？C 和其它高级语言比较有什么特点？
3. C 语言程序由哪几部分组成？C 语言程序结构有什么特点？
4. 编写一个 C 程序，输入 a、b、c 三个值，计算并输出其平均值
   ave＝(a＋b＋c)/3
5. 指出哪些不是标识符，为什么？
   3id　_ _ yes　o _ no _ o _ n　00 _ go　star * it
   1 _ i _ am　one _ i _ aren't　me _ to _ 2　xYshouldI
   int
6. 下面一行字符串能作为一条注释吗？为什么？
   /* This is an attempt /* to next */ a comment */
7. 什么是关键字？它与标识符有什么异同？