

C语言程序设计

钟廷志 赵洪波 著

人民邮电出版社

书 名：C语言程序设计
作 者：钟廷志 赵洪波
出 版 社：人民邮电出版社
出版日期：2004年1月
ISBN：7-115-12153-2/G634.67
定价：22.00元

第1章 C 语言概述

计算机语言经历了较长的发展阶段，各类语言的出现为计算机技术的发展提供了强大的动力。但是，当今时代，影响最大的还是各类高级语言，其中 C 语言便是典型代表。本章首先向读者介绍有关 C 语言的一般知识。

本章学习目标

- 了解 C 语言的历史和发展过程。
- 了解 C 语言的特点及 C 程序的组成部分。
- 了解 C 程序的编译、链接过程。

1.1 C 语言简史

C 语言于上个世纪 70 年代诞生于美国的贝尔实验室。在此之前，人们编写系统软件主要是使用汇编语言。由于汇编语言编写的程序依赖于计算机硬件，其可读性和可移植性都比较差。而高级语言的可读性和可移植性虽然较汇编语言好，但一般高级语言又不具备低级语言能够直观地对硬件实现控制和操作，程序执行速度快的特点。在这种情况下，人们迫切需要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。于是 C 语言就应运而生了。

由于 C 语言既具有高级语言的特点又具有低级语言的特点，因此迅速普及，成为当今最有发展前途的计算机高级语言之一。C 语言既可以用来编写系统软件，也可以用来编写应用软件。例如美国的导弹防御系统就是用 C 语言编写而成的。现在，C 语言广泛地应用在机械、建筑和电子等行业，编写各类应用软件。

C 语言的发展历程如下。

- C 语言是美国贝尔实验室的 D.M.Ritchie 在 B 语言的基础上设计出来的，此时的 C 语言只是为了描述和实现 UNIX 操作系统的一种工作语言。在一段时间里，C 语言还只在贝尔实验室内部使用。
- 1975 年，UNIX 第 6 版公布后，C 语言突出的优点引起人们的普遍注意。
- 1977 年出现了可移植的 C 语言。
- 1978 年 UNIX 第 7 版的 C 语言成为后来被广泛使用的 C 语言版本的基础，被称为标准 C。
- 1983 年，美国国家标准化协会（ANSI）根据 C 语言问世以来的各种版本，对 C 进行发展和扩充，并制定了新的标准，称为 ANSI C。
- 1990 年，国际标准化组织 ISO 制定了 ISO C 标准，目前流行的 C 语言编译系统都是以它为标准的。

1972 年以来，C 语言几经修改和发展，出现多个版本。在微型机上使用的有 Microsoft C、Turbo C、Quick C 以及 Borland C 等，它们的不同版本又略有差异。目前最流行的版本是 Microsoft C 和 Turbo C。本书的叙述基本上以 ANSI C 为基础，上机练习则以 Turbo C 环境为基础。

1.2 C 语言的特点

C 语言兼有汇编语言和高级语言的优点，既适合于开发系统软件，也适合于编写应用程



序。被广泛应用于事务处理、科学计算、工业控制、数据库技术等领域。

C 语言之所以能存在和发展，并具有强大的生命力，这都要归功于其鲜明的特点。这些特点是多方面的，归纳如下。

(1) C 语言是结构化的语言

C 语言程序有 3 种基本结构：

- 顺序结构。
- 选择结构。
- 循环结构。

而由这 3 种基本结构组成的程序可以解决许多复杂的问题。C 语言还具有结构化的控制语句，如 `if...else` 语句、`while` 语句、`switch` 语句以及 `for` 语句等，使用这些语句可以方便地控制程序的流程。因此，C 语言是理想的结构化语言，符合现代编程风格的要求。

(2) C 语言是模块化的语言

C 语言主要用于编写系统软件和应用软件。一个系统软件的开发需要很多人经过几年的时间才能完成。一般来说，一个较大的系统程序往往被分为若干个模块，每一个模块用来实现特定的功能。

在 C 语言中，用函数作为程序的模块单位，便于实现程序的模块化。在程序设计时，将一些常用的功能模块编写成函数，放在函数库中供其它函数调用。模块化的特点可以大大减少重复编程。程序设计时，只要善于利用函数，就可减少劳动量、提高编程效率。

(3) 语言简洁、紧凑，使用方便、灵活

C 语言一共只有 32 个关键字（如表 1-1 所示）和 9 种控制语句，程序书写形式自由，主要用小写字母表示。在一般语言中的许多结构单元在 C 语言中都通过库函数调用来完成，库函数可根据需要方便地扩充，压缩了一切不必要的程序组成部分。

表 1-1 ANSIC 的关键字

auto	break	case	char	const
continue	default	do	double	else
enum	extern	float	for	goto
if	int	long	register	return
short	signed	sizeof	static	struct
switch	typedef	union	unsigned	void
volatile	while			

(4) 程序可移植性好

C 语言程序便于移植，目前 C 语言在许多计算机上的实现大都是由 C 语言编译移植得到的，不同机器上的编译程序大约有 80% 的代码是公共的。程序不做任何修改就可用于各种型号的计算机和各种操作系统。

(5) 数据结构丰富，具有现代化语言的各种数据结构

C 语言的基本数据类型有整型（如 `int`、`unsigned int` 等）、实型（如 `float`、`double` 等）以及字符型（`char`）等。在此基础上还可创建各种构造数据类型，如数组、指针、结构体和共用体等。使用 C 语言还能用来实现复杂的数据结构，如链表、树等。这样丰富的数据结构无疑极大地增强了 C 语言的功能。



(6) C 语言运算符丰富、代码效率高

C 语言共有 34 种运算符，使用各种运算符可以实现在其他高级语言中难以实现的运算。在代码质量上，C 语言可与汇编语言媲美，其代码效率仅比用汇编语言编写的程序的代码低 10%~20%。

总之，C 语言以其突出的优点，吸引着越来越多的使用者的关注。目前，国内外使用和研究 C 语言的人正在迅猛增加，优秀的 C 语言版本和配套的工具软件不断出现。现在，C 语言不仅为计算机专业工作者所使用，而且为广大的计算机应用人员所喜爱和使用。



虽然 C 语言的优点很多，但并不是完美无缺。仍然存在一些不足，例如运算符的优先级太多，不便于记忆；语法限制不太严格，程序设计自由度大；类型检验太弱，不同类型数据转换比较随便，因此不太安全等。

1.3 C 程序的组成

程序是在特定语言环境下为解决某一具体问题而编写的代码集合。下面先介绍几个简单的 C 程序，从中分析 C 程序的组成及其特性。

【例1-1】 在屏幕上显示 I love you, China!。

```
#include<stdio.h>

void main()
{
    printf("I love you,China!");
}
```

本程序的作用是在屏幕上输出以下信息：

```
I love you,China!
```

本程序的说明如下。

- “main()”表示主函数，大括号“{}”括起来的部分称为函数体。每个程序中都必须有且仅有一个 main 函数。
- 语句是组成程序的最小结构单元。主函数的函数体中只包括一个输出语句：printf()是 C 语言的输出函数；双引号内的内容按原样输出；“\n”是换行符，即在输出“I love you,China!”后光标换到下一行。
- 每个语句最后必须有一个分号。

【例1-2】 比较两个变量值的大小。

```
#include<stdio.h>

void main()
{
    int a,b,c;           /*定义变量 a,b,c 为整型变量*/
    a=20;b=15;         /*给变量 a,b 赋初值*/
    c=a>b?a:b;         /*比较变量 a,b 的大小，将较大的数赋给变量 c*/
    printf("%d\n",c);  /*输出比较结果*/
}
```

程序运行结果为：



本程序的作用是比较整数 a 和 b 的大小，说明如下。

- 程序中 “/*……*/” 包围的部分为注释（如果在 $c++$ 的环境下运行 c 程序，其注释部分可用 “//” 代替 “/*……*/”，把注释语句放在 “//” 之后即可）。注释部分便于阅读程序的人理解程序员的设计意图，并不影响程序的编译和运行。注释可以放在程序中的任何位置。
- C 语言中使用变量作为数据组织的基本形式，变量的基本操作包括变量的定义、变量的赋值以及变量的运算等。
- 程序第 4 行定义了 a 、 b 和 c 这 3 个变量，并指定它们为整型（ int ）变量；程序第 5 行是两个赋值语句，为变量 a 和 b 设定初始值分别为 20 和 15；程序第 6 行将变量 a 与 b 中较大的数赋给变量 c 。

【例1-3】 使用函数对两个数据进行加减运算。

```
#include<stdio.h>
void main()
{
    int add(int x,int y);
    int mul(int x,int y); /*对被调用函数的声明*/
    int a,b,sum,count; /*声明部分，定义变量*/
    a=20;b=35; /*为 a,b 赋初值*/
    sum=add(a,b); /*调用函数 add，将得到的值赋给 sum*/
    count=mul(a,b); /*调用函数 mul，将得到的值赋给 count*/
    printf("sum=%d,count=%d\n",\
sum,count); /*输出 sum,count 的值*/
}
int add(int x,int y) /*定义 add 函数，函数值为整型，形式参数 x 和 y 为整型*/
{
    int z; /*add 函数中的声明部分，定义本函数用到的变量 z 为整型*/
    z=x+y;
    return(z); /*将 z 的值返回，通过 add 带回调用处*/
}
int mul(int x,int y) /*定义 mul 函数*/
{
    int w; /*定义 mul 函数中用到的变量 w 为整型*/
    w=x*y;
    return(w); /*返回 w 的值*/
}
```

运行结果为：

```
sum=55,count=700
```

这个程序的目的是显示两个整数 a 、 b 的和与积，说明如下。

- 函数是构成 C 程序的基本单位。本程序包含 3 个函数：主函数 $main()$ 、函数 $add()$ 和 $mul()$ 。函数 $add()$ 的作用是求 a 、 b 之和，函数 $mul()$ 的作用是求两个



数的乘积。

- 在 C 程序中，一个函数可以使用函数调用的方法访问另一个函数。程序中的第 6 行和第 7 行为 `main()` 函数对 `add()` 函数和 `mul()` 函数的调用。在调用函数时将实际参数 `a` 和 `b` 的值分别传送给函数 `add()` 和 `mul()` 中的形式参数 “`x`” 和 “`y`”，然后执行函数相应的运算，并将结果通过 `return` 语句返回给主函数。

通过以上几个程序，可以看出 C 语言的以下结构特点。

(1) 语句是 C 程序的最小构成单位

C 程序中每个语句以一个 “`;`” (分号) 结束。分号是 C 程序语句必不可少的组成部分。如【例 1-3】中第 7 行 “`a=20; b=35;`”，如写成 “`a=20 b=35`”，编译时在 “`a=20`” 后面未发现分号，就把下一行 “`b=35`” 也作为上一行的语句的一部分，系统将显示程序有语法错误。

(2) `main()` 函数是程序执行的起点

一个 C 程序总是从 `main()` 函数开始执行的，而不论 `main()` 函数在整个程序中的位置如何。`main()` 函数可以放在程序的最前面，也可以放在程序最后，或放在一些函数之间。

(3) C 程序书写格式自由

C 程序的一行内可以书写几个语句。如【例 1-3】中第 7 行 “`a=20;b=35;`”，它包含 “`a=20;`” 和 “`b=35;`” 两个语句。一个语句也可以写在多行上，用 “`\`” 作续行符。如【例 1-3】中第 10 行。C 程序没有行号，不像其他语言（如 COBOL、Fortran 等）那样严格规定书写格式。

(4) 可在 C 程序的任何部分添加注释语句，以提高程序的可读性

C 语言注释由 “`/*`” 开始，至 “`*/`” 结束。注释可分为若干行，但不允许嵌套。注释使程序变得清晰，增加了程序的可读性，便于阅读和理解。给程序加上注释是一个良好的编程习惯。注释部分不会对程序本身的运行产生影响。

(5) 函数是 C 程序的基本组成单位

函数是构成 C 语言程序的基本单元。一个 C 源程序中有且仅有一个 `main()` 函数，还可以包含任意多个其他函数。程序运行时，通过函数调用的形式来实现函数之间的信息传递。被调用的函数可以是系统提供的库函数（如 `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 `add()` 和 `mul()` 函数）。各函数的书写位置无关紧要。

(6) 函数是 C 程序模块化结构的重要体现

根据【例 1-3】可以归纳出 C 语言程序的一般形式：

```
main()
{
    变量定义          (声明部分)
    调用函数 fun1
    调用函数 fun2
    .....
}
fun 1()
{
    变量定义
    调用函数 fun3
```



```

... .
    程序段
}
.....

funn( )
{
    变量定义
    调用函数 fun7
    程序段
}

```

其中 fun1、funn 为用户自定义函数。



C 语言把许多常用的函数功能模块通过库函数的形式提供给用户。C 语言的函数库十分丰富。函数库是由 C 语言开发者编制，用户需要时直接调用即可。灵活使用库函数，可以提高编程效率，使 C 程序本身规模减少，编译程序简单，很容易在各种机器上实现，同时还增强程序的可移植性。

1.4 C 程序的编译、链接和运行

在了解了 C 语言的基本知识后，下面接着介绍运行 C 语言程序的相关问题。本节介绍目前使用最广泛的 C 语言程序开发软件——Turbo C 的基本知识。Turbo C 是 Borland 公司开发的一个适用于微型计算机，同时集程序编辑、编译、链接和调试为一体的 C 编译系统。它具有良好的用户界面和丰富的库函数，同时还有运行速度快、效率高和功能强等优点，使用非常方便。

一个 C 语言程序要能够最终实现既定的功能，需要依次经历以下基本环节。

- 编辑：使用 C 语言编写程序代码，创建源文件。
- 编译：在 C 程序的编译过程中，可以查出程序中的语法错误。编译器将程序转换为机器代码后即可生成目标程序（.obj）。
- 链接：C 程序是模块化设计程序，一个 C 程序可能由多个程序设计者分工合作编写。最后需要将库函数以及其他目标程序链接为一个整体，生成可执行文件（.exe）。
- 运行：运行源文件经过编译链接后生成的可执行文件（.exe）即可获得正确的结果。

下面介绍使用 Turbo C 开发 C 语言程序的一般方法和步骤。

(1) 启动 Turbo C

启动 Turbo C 后，将打开 Turbo C 集成开发环境，计算机屏幕上出现 Turbo C 主窗口，这也是 Turbo C 的图形用户界面。该界面由主菜单、编辑窗口、信息窗口和功能键提示行 4 部分组成，如图 1-1 所示。

(2) 主菜单

主菜单包含下面 8 个下拉菜单。

- **【File】**：文件操作，载入或保存文件，管理目录，调入 DOS 和退出 Turbo C 系统。



- **【Edit】**: 编辑操作, 调用 Turbo C 的编辑程序。
- **【Run】**: 运行操作, 运行当前程序。
- **【Compile】**: 编译操作, 编译、链接当前程序。
- **【Project】**: 项目文件, 管理多文件项目。
- **【Options】**: 选项, 设置编译程序和链接程序的选择项。
- **【Debug】**: 调试, 设置各种调试选择项。
- **【Break/Watch】**: 断点/监视, 主要用于程序的调试。

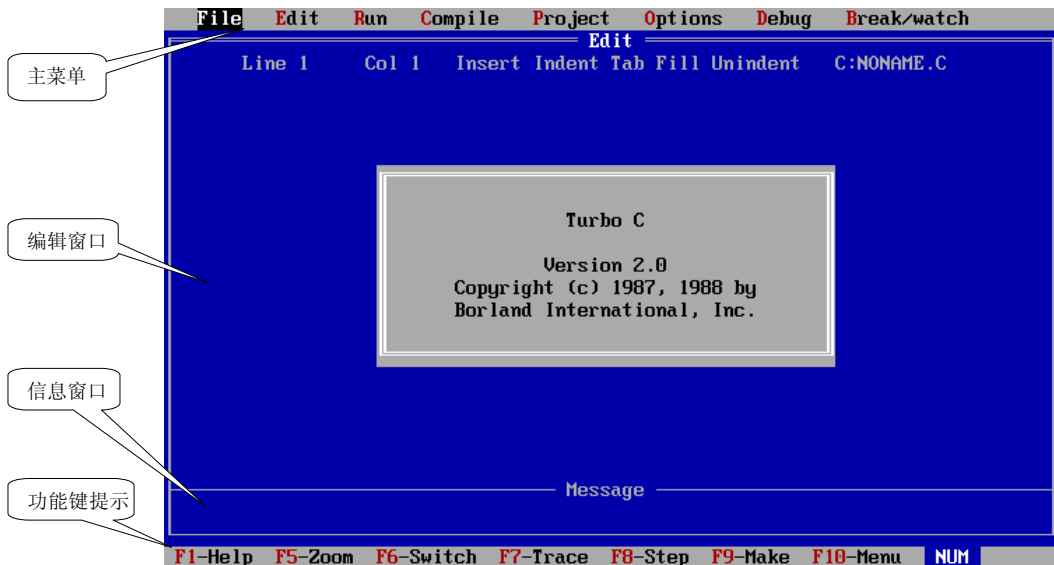


图1-1 Turbo C 主窗口

(3) 退出 Turbo C 系统

退出 Turbo C 系统有 3 种方法:

- 按 **Alt+x** 组合键。
- 选择 **【File】** 菜单下的 **【Quit】** 命令。
- 按 **Alt+F** 组合键后键入 “Q”。

(4) 编辑源文件

在编辑 (Edit) 状态下可以根据需要输入或修改源程序。编好一个程序后, 需要把它录入到 Turbo C 中进行调试、编译, 如编译不能通过, 则要对程序进行修改。

可以按 **Alt+E** 组合键选择 **【Edit】** 菜单, 按回车后进入编辑状态, 此时可对源文件进行编辑, 其编辑方法与一般的文字处理程序类似。编辑命令有光标移动、插入/删除、块操作及其他命令等。

如源程序 “abc.c” 已经存在, 则应通过 **【File】** 菜单下的 **【Load】** 命令将 “abc.c” 调入到 Turbo C 环境。然后进入编辑程序, 修改过程与前面介绍的方法相同。

(5) 编译源程序

编辑好一个源程序后, 必须经过编译、链接生成可执行文件后才能运行。按 **F10** 键, 将光标移动到编译 **【Compile】** 菜单, 按回车选择 **【Compile to OBJ】** 命令。则对源程序进行编译, 得到一个后缀为 “.obj” 的目标程序。如编译的源文件名为 “abc.c”, 则目标文件名为 “abc.obj”。



在 Turbo C 中进行编译时，会弹出一个【编译信息框】，显示编译状态和编译结果。如果发现错误，【信息窗口】中会显示所有错误信息。此时，按下任意键，【编译信息窗口】会消失，屏幕上会恢复显示源程序，光标会停留在出错之处。屏幕下半部分的【信息窗口】会显示出有错误的行和错误的原因。根据此信息修改源程序。修改确认后，再按 **F10** 进行编译，直到没有错误信息为止。一般情况下，一个程序需要经过多次地编辑和修改才能通过编译。

(6) 链接源程序

生成目标文件后，再选择编译菜单下的【Link EXE file】项，进行链接操作，生成一个后缀为“.exe”的可执行文件。例如目标文件为“abc.obj”，则生成的可执行文件为“abc.exe”。同样，如果发现错误，还要返回编辑修改。也可以将编译和链接合成一个步骤进行。选择编译菜单下的【Make EXE file】项或按 **F9** 键，即可一次完成编译和链接。

(7) 运行程序

经过编辑、编译、链接后，产生了一个可执行的文件（后缀为“.exe”）。将文件调入到 Turbo C 中，按 **F10** 键，选择主菜单中的【Run】，按回车键，在其下拉菜单中选择【Run】项。系统就会执行已编译和链接好的目标文件，得到程序的运行结果。按 **Alt+F5** 键可以看到运行结果。如果程序需要输入数据，则应从键盘输入所需数据，然后程序会接着运行，输出结果。可按任意键返回【TC 集成环境】窗口。

有时通过编译和链接的文件，运行后发现结果不正确，可能是逻辑错误造成的，要重新修改程序。修改过程与前面编辑时相同。

关于 Turbo C 环境和使用方法的详细介绍，请参阅与本书配套的《C 语言上机指导与练习》。

1.5 小结

本章首先介绍了 C 语言的发展简史。C 语言诞生于美国贝尔实验室，经过不断地修改、完善，C 语言目前已成为人们广泛使用的语言之一。现在通用的 C 语言为 1983 年美国国家标准化协会（ANSI）制定的标准 C 语言，称为 ANSI C。

C 语言的特点使得它在各种语言中独树一帜，受到人们的青睐，广泛应用于社会各个领域，其特点归纳如下。

- 结构化的语言。
- 模块化的语言。
- 简洁、紧凑，使用方便，灵活。
- 可移植性好。
- 数据结构丰富。
- 运算符丰富，生成代码质量高。

C 语言程序由函数组成。一个 C 源程序有且仅有一个 main() 函数，但是可以包含任意多个其他函数，因此，函数是 C 程序的基本单元。C 语言开发者编制了内容丰富的库函数，编程者可以通过函数调用的形式使用这些库函数，方便、快捷、编程效率高。

C 程序中，每一个基本语句和数据定义都是以分号“;”结束，书写格式自由，一行可写多个语句，一个语句也可以分别写在几行上。可以使用 /*……*/ 对程序添加注释以方便读者阅读、理解程序的编写意图。

一个 C 程序的开发过程如图 1-2 所示。

C 语言是编译型语言，源程序必须经过编译和链接生成可执行文件后才能运行。编译过程中可以检查出程序的错误，经过编辑修改后，再重新编译。一般情况下，一个程序要经过多次编辑和修改才能通过编译。

链接目标文件的目的是要生成最终的可执行文件。C 程序经过编译、链接，产生了一个可执行文件 (*.exe)。运行该文件，即可得到程序的运行结果。

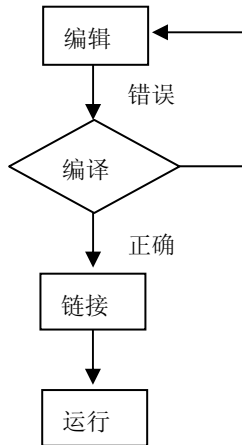


图1-2 C 程序的开发过程

1.6 习题

1. C 语言有何特点？
2. C 语言主要用途是什么？与其他语言有什么区别？
3. 一个 C 程序由哪几部分组成？
4. C 语言以函数作为基本单元，有什么好处？
5. 一个 C 程序的开发应经过哪几个步骤，简述各步骤的作用。
6. 请参照本章例题，编写一个 C 程序，输出以下信息：

I love this game!

7. 上机运行本章 3 个例题。熟悉所用系统的上机方法与步骤。
8. 上机运行习题第 6 题。

第2章 数据类型、运算符与表达式

计算机处理的对象是数据，而数据是以某种特定的形式存在的，例如，姓名可以用一个字符数据表示，年龄可以用一个整数表示，各种商品的价格可以用实数表示等。为了能准确、方便地用数据描述生活中的各种信息，C 语言将数据划分为不同的类型。当然，我们的最终目的是对各种数据进行处理，C 语言特定的运算符和表达式能完成各种数据的处理。本章学习 C 语言的基础知识：数据类型、运算符与表达式。

本章学习目标

- 了解常量与变量的含义。
- 掌握数据类型及类型间的混合运算。
- 掌握运算符与表达式及其优先级和结合性。

2.1 常量与变量

常量和变量是 C 语言中的两种重要的数据组织形式。在日常生活中，常常使用各种容器来盛放物品，例如，使用酒瓶和酒杯来盛放酒。但是，使用酒瓶和酒杯来盛放酒时有所区别，一般来说，一个酒瓶在未开启前里面盛放的是一种固定不变的酒，而一个酒杯随时盛放不同种类的酒。由此可以引出 C 语言里面常量和变量的概念。

在 C 语言中，各类数据是计算机处理的主要对象，这些数据在计算机处理过程中，也需要使用一个类似“容器”的东西加以表达和储存，这通常是由常量和变量来完成的。

- 常量：用来表达在程序运行过程中其值保持恒定的数据。
- 变量：用来表达在程序运行过程中其值可以根据需要经常更新的数据。

想一想：常量中的数据是如何存入的？当变量中的数据更新后，它原来的值到哪儿去了？

2.1.1 常量

在程序的运行过程中，其值不发生改变的量称为常量。C 语言中的常量包括普通常量和由 `#define` 所定义的符号常量。

普通常量：普通常量又可分为整型常量、字符常量、字符串常量和实型常量。普通常量从其字面形式即可判断。

符号常量：也可以用一个标识符代表一个常量，这种用标识符代表的常量称为符号常量，此时，符号就像是常量的别名。

(1) 定义格式

符号常量由宏定义命令 `#define` 语句进行定义，它的一般形式为：

```
#define 标识符 字符串
```

(2) 应用实例

下面通过一个实例来说明符号常量的用法，程序中每个语句后面都有注释，可以帮助读者阅读程序。

【例2-1】 编写一个求圆面积的程序，已知圆半径 $r=3m$ ，圆周率取 3.14。

```
#include<stdio.h>
#define PI 3.14
```



```

void main()
{
    double radius,area;           /*定义存储半径和运算结果的变量*/
    radius=8;                     /*为半径变量设置初始值*/
    area=PI*radius*radius;       /*计算圆的面积*/
    printf("该圆的面积为:%f\n",area); /*输出运算结果*/
}

```

运行程序后，输出结果为：

该圆的面积为：200.960000

程序中的第 2 行用宏定义“#define”定义了一个标识符 PI，这是一个符号常量，其值为 3.14，在程序运行过程中，其值不变。程序运行时，所有的 PI 都用“3.14”代替。

(3) 符号常量用途

在程序中使用符号常量，具有以下优点。

- 含义清楚。在定义符号常量时，可以按“见名知义”的原则为符号常量命名。如用 PRICE 代表价格、用 PI 代表圆周率等。
- 修改方便，在需要改变一个常量的值时，只要在符号常量的定义位置修改即可，其值在整个程序中都将发生改变。如当某种商品的价格改变时，只需改变宏定义处的字符串，就能使整个程序中用到此价格的地方全部改变。



常量的值保持不变，并不是说其值不可以被改变，只是说在程序运行过程中不会被修改，但是还是可以通过修改定义的方法来修改，例如符号常量的修改。另外，为了区别普通变量，通常将符号常量用大写字母来表示。

2.1.2 变量

在程序运行过程中，其值可以改变的量称为变量。变量可分为整型变量、字符变量、实型变量等。

(1) 变量的基本要素

每一个变量都具有以下 4 个基本要素。

- 变量名：用于区分不同变量的标识符号。
- 变量值：变量内存放的数据，变量使用其值参与各种运算，变量值可以在程序运行过程中不断被修改。
- 变量对应的内存空间：变量将其最新值存放在系统为其分配的内存空间中。当系统需要读取一个变量的值时，访问该变量的存储空间并取出变量值；当系统需要改写一个变量的值时，访问该变量的存储空间并将新值写入该存储空间。
- 变量的数据类型：变量在定义时，根据数据类型的不同占据不同长度的存储空间。

(2) 变量名

为了区分不同的变量，每个变量都有一个名称。在对变量命名时，应遵循 C 语言的规定。C 语言规定变量名只能由字母、数字和下划线 3 种字符组成，且第 1 个字符必须为字符或下划线。例如：

hi、_day、sum、month、s23、567、average、football、Student、zhang-yang、QB、



PRICE, 都是合法的变量名。Sun.day、\$500、@yahoo、a<=b 均为不合法的变量名。



在 C 语言中, 用来标识变量名、符号常量名、函数名、数组名以及文件名等的有效字符序列称为标识符。简单地说, 标识符就是一个名称。这些标识符的命名规则基本相似, 以后不再赘述。

(3) 变量的定义

任何变量在使用前必须定义。在程序中使用未经定义的变量是不合法的。

- 变量定义的实质。

定义一种数据类型变量的实质, 就是在内存中为该变量分配一定大小的存储空间。此时变量名作为访问该存储空间的标识符。在程序中对该变量进行读写操作时, 都要访问该变量对应的存储空间进行数据读写。

- 变量定义的方法。

变量定义的一般形式为:

<类型名> <变量名>

例如:

```
int a,b,c;      /*定义变量 a,b,c 为整型*/
float f;       /*定义变量 f 为实型*/
char c;        /*定义变量 c 为字符型*/
```

(4) 变量的值

任何变量必须具有确定的数值才能参加运算和操作, 否则将会引发潜在的错误。变量定义后其赋值的过程称为变量的初始化, 变量初始化的方法有以下几种。

- 可以在变量定义的同时为其赋初值:


```
int age=10;
```
- 也可以在定义变量后再为其赋初值:


```
int age;
```
- age=10; 此处的 “=” 为赋值运算符。
- 还可以同时定义多个变量并为其赋初值:

```
int a=6,b=3,c=7;
```

一个变量在某一时刻只有一个确定的数值, 一个变量在获得新值时原值将不复存在。

(5) 变量的用法小结

在使用变量时, 应注意以下几方面的问题。

- 任何变量必须定义在先, 使用在后。C 语言中的每一变量具有惟一确定的数据类型, 变量定义后, 在程序编译时系统根据其类型为其在内存中分配相应的存储单元。未经定义的变量将不会被分配存储单元, 因而无法实现对该变量的数据读写。
- 在定义变量名和其他标识符时, 应做到见名知义。习惯上, 变量名用小写字母, 符号常量名用大写字母, 以示区别。
- 大写字母和小写字母被认为是不同的字符。如 SUM 和 sum、PRICE 和 price 等都是不同的变量。
- ANSI C 标准没有规定标识符的长度, 但是各编译系统有自己的规定。Turbo C 允许 32 个字符, 当多于 32 个字符时, 系统自动取前面 32 个字符作为标识符, 后面的不被识别。在对变量命名时应注意其有效标识符长度, 以免出现重复的变量名。



2.2 基本数据类型

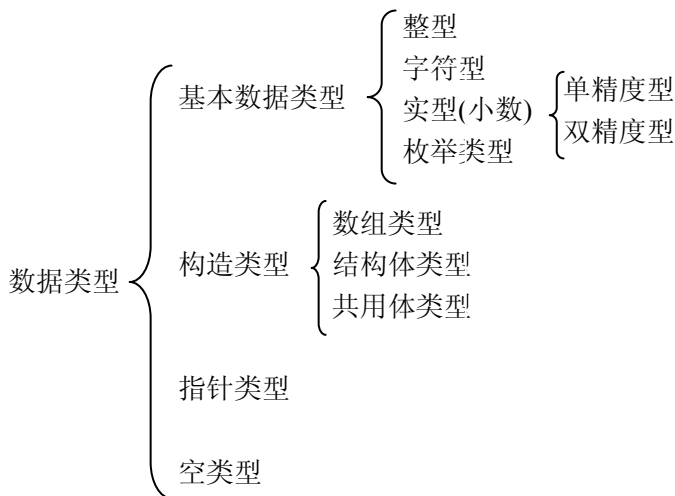
在计算机中进行处理的信息都可以表达为数据的形式。这里的数据不仅包括我们熟知的数学运算中的数据，还包括声音、图像等复杂信息。

数据在计算机中进行处理时，首先必须解决的问题是数据在存储器里的存储问题。

想一想：

- 由于计算机的内存大小有限，怎样存储更加节约存储空间？
- 怎样存储便于数据的查找。

为了便于 C 语言的数据处理，合理地使用存储空间，将数据划分为不同的类型，进行分类处理。C 语言的数据类型如下：



本节主要介绍基本数据类型及其使用方法，对于其他数据类型，将在后面的章节中讲述。

2.2.1 划分数据类型的必要性

在 C 程序中，每一个数据都被赋予一个确定的数据类型，不同数据类型的数据在存储时占用不同长度的内存空间。

(1) 引例

假设要记录一个人的详细资料，具体内容如下。

- 姓名：徐晴。
- 性别：女。
- 年龄：28。
- 工资：2000。
- 工作单位：成都市西城派出所侦察科。

个人资料中各基本信息的特点如下。

- 姓名：一串较短的字符。
- 性别：只有“男”和“女”两种选择。
- 年龄：一般为一个 100 以内的整数。
- 工资：一个实数。



- 工作单位：一串较长的字符。

由此可见：不同的数据信息在表达时具有不同的特点。表示“姓名”、“性别”和“工作单位”的数据，都是符号量，这些符号量一般不用作数学计算。但是，虽然它们都是符号量，这3个符号量的数据长度并不相同，“性别”用一个汉字即可表示；“姓名”一般需要用2~4个汉字表示；而“工作单位”需要用一串汉字来表示；对于“年龄”数据，则需要用一个100以内的整数来表示；对于“工资”数据，则需要使用一个实数来表示。

(2) 划分数据类型的必要性

划分数据类型的意义如下。

- 划分数据类型有利于数据的管理。

不同的数据在程序中有不同的处理方法，例如整数和实数类型的数据可以求和、求差，可以比较大小；而符号类型的数据一般不对其进行求和、求差等运算，但是可以比较两个数据是否相同，甚至可以比较大小。因此，划分数据类型后，可以便于数据的管理。

- 划分数据类型有利于有效地利用计算机有限的内存空间。

计算机在处理数据时，首先必须在内存中为数据开辟一片存储区域。而对于计算机来说，内存总是有限的，从前面的例子可以看出，即使是同一种类型的数据，其数据量也有大有小。因此，对数据划分数据类型后，可以根据数据量的大小为不同数据设置不同的类型，合理而有效地利用有限的内存空间。



C 程序中的每一个数据都必须有一个确定的数据类型，没有无类型的数据，也不可能有一个数据同时具有多种数据类型。

2.2.2 整型数据

整型数据用来表示整数。

一、整型常量

整型常量即整常数。C 语言的整常数可以用以下 3 种形式表示。

- 十进制整数：例如 154、-340、0 等。
- 八进制整数：例如 0140、-0234 等。八进制整数用以 0 开头的数表示，0140（即 $(140)_8$ ）表示八进制整数 140，相当于十进制整数 96。
- 十六进制整数：例如 0x140、-0x15 等。十六进制整数用以 0x 开头的数表示，0x140（即 $(140)_{16}$ ）表示十六进制整数 140，等于十进制数 320。-0x15 等于十进制数 -21。

下面举例说明整型常量的 3 种表示方法及相互关系。

【例2-2】 将数据以不同进制输出，其中程序中有关变量定义的内容将在稍后讲述，请参照程序后面的注释阅读程序。

```
#include<stdio.h>
void main()
{
    int x;                /*定义变量类型*/
    int y;
    int z;
```



```

x=140; /*为变量赋初值*/
y=0140;
z=0x140;

printf("%d\t%d\t%d\n",x,y,z); /*按十进制输出变量的值*/
printf("%o\t%o\t%o\n",x,y,z); /*按八进制输出变量的值*/
printf("%x\t%x\t%x\n",x,y,z); /*按十六进制输出变量的值*/
}

```

程序运行结果为：

```

140    96    320
214    140   500
8c     60    140

```

在本例中“%d”、“%o”、“%x”为格式转换控制符，分别用于输出十进制、八进制、十六进制整数。输出时，将后面变量的值进行替换。“\t”为水平制表转义符，“\n”为换行转义符。其关系如图 2-1 所示，关于 printf() 函数的用法将在第 3 章中作详细的介绍。

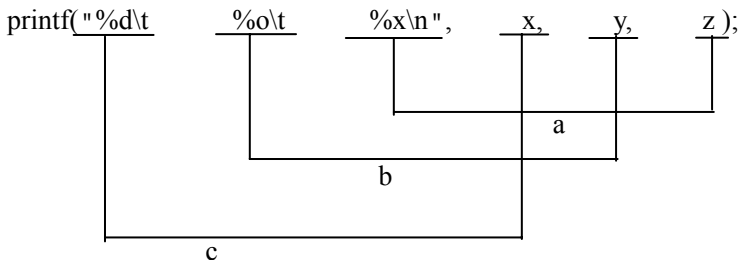


图2-1 整型数据的输出格式

其中：

- 变量 a: 按十六进制输出 z 的值并换行。
- 变量 b: 按八进制输出 y 的值并向后跳一个水平制表位。
- 变量 c: 按十进制输出 x 的值并向后跳一个水平制表位。

二、整型变量

整型变量用来存储整型数据。

(1) 整型变量的基本类型

数据在内存中是以二进制数形式存放的。为了充分利用内存空间，提高运行速度，根据数据的表数范围将整型变量分为以下 3 类。

- 基本整型，以 int 表示。
- 短整型，以 short int 表示，或以 short 表示。
- 长整型，以 long int 表示，或以 long 表示。

(2) 有符号和无符号整型变量

在实际应用中，有些变量的值始终为正数，如一个人的年龄、身高和体重等。为了充分利用变量的表数范围，节约内存空间，C 语言将这类变量定义为“无符号 (unsigned) 类型”。unsigned 可以加在以上 3 种类型的整型变量的前面，以指定变量为“无符号数”。

如果在变量前加上 signed (有符号)，则指定变量为“有符号数”。这时的整型变量还可以存放负数。