普通高等教育“十一五”国家级规划教材
普通高等学校计算机教育“十二五”规划教材

C语言程序设计教程

（第4版）

THE C PROGRAMMING LANGUAGE
(4th edition)

李丽娟 ◆ 主编

人民邮电出版社
POSTS & TELECOM PRESS

普通高等教育“十一五”国家级规划教材
普通高等学校计算机教育“十二五”规划教材

C 语言程序设计教程

（第 4 版）

THE C PROGRAMMING LANGUAGE
(4^{th} edition)

李丽娟 ◆ 主编

人 民 邮 电 出 版 社
北 京

图书在版编目（CIP）数据

C语言程序设计教程 / 李丽娟主编. -- 4版. -- 北京 : 人民邮电出版社, 2013.1(2013.3重印)
普通高等学校计算机教育“十二五”规划教材
ISBN 978-7-115-29599-6

Ⅰ. ①C… Ⅱ. ①李… Ⅲ. ①C语言－程序设计－高等学校－教材 Ⅳ. ①TP312

中国版本图书馆CIP数据核字(2012)第262555号

内 容 提 要

本书以C语言的基本语法、语句为基础，深入浅出地讲述了C语言程序设计的基本概念、思想与方法。全书以程序案例为导向，采用计算思维的方法设计程序，通过程序案例，拓宽学生的思维，引导学生自主思考，逐步掌握程序设计的一般规律和方法。从对基本概念的讲解到编写程序解决实际问题，本书注重解决问题的方法引导。全书理论联系实际，突出模块化程序设计方法。

全书内容可分为三部分，共11章。第一部分为第1章、第2章，是初学者的入门知识，简单介绍C语言的基础知识，主要内容有C语言程序的基本结构、数据类型和数据的存储方式、基本的程序表达式。第二部分为第3章～第5章，是程序设计的基础部分，主要介绍程序算法的方法、程序语句的基本控制结构。掌握了第一、第二部分的内容，读者可以完成简单的程序设计。第三部分为第6章～第11章，介绍模块化程序设计的概念和实现的方法，主要内容有函数、数组、指针、结构体、文件、位运算等。通过对这三部分知识单元的学习，读者可以逐步认识模块化程序设计的思想，掌握模块化程序设计的方法。

全书语言简洁，通俗易懂，内容叙述由浅入深。本书适合作为大学本科和专科院校的教材，也可供一般工程技术人员参考。

普通高等学校计算机教育“十二五”规划教材

C语言程序设计教程（第4版）

◆ 主　　编　李丽娟
责任编辑　邹文波

◆ 人民邮电出版社出版发行　　北京市崇文区夕照寺街14号
邮编　100061　　电子邮件　315@ptpress.com.cn
网址　http://www.ptpress.com.cn
中国铁道出版社印刷厂印刷

◆ 开本：787×1092　1/16
印张：19.25　　2013年1月第4版
字数：534千字　　2013年3月北京第3次印刷

ISBN 978-7-115-29599-6

定价：38.00元

读者服务热线：(010)67170985　印装质量热线：(010)67129223
反盗版热线：(010)67171154
广告经营许可证：京崇工商广字第0021号

目 录

第1章 引言

1.1 C语言的发展过程

C 语言的前身是 ALGOL 60。之后剑桥大学将 ALGOL 60 语言发展成为 CPL（Combined Programming Language）语言，1967 年由剑桥大学的 Matin Richards 对 CPL 语言进行了简化，产生了 BCPL 语言。1970 年由美国贝尔实验室的 Ken Thompson 对 BCPL 进行了修改，命名为“B 语言”。并用 B 语言写了第一个 UNIX 操作系统。

1973 年，美国贝尔实验室的 D.M.RITCHIE 在 B 语言的基础上设计出了一种新的语言——C 语言。1977 年由 Dennis M.Ritchie 发表了不依赖于具体机器系统的 C 语言编译文本《可移植的 C 语言编译程序》。1978 年由 Brian W.Kernighian 和 Dennis M.Ritchie 合作出版了著名的《The C Programming Language》一书，1983 年由美国国家标准协会（American National Standards Institute）在此基础上制定了一个 C 语言标准，通常称为 ANSI C。从而使 C 语言成为目前世界上流行最广泛的高级程序设计语言。

1.2 C语言的特点

C 语言是一种结构化的程序设计语言，它简明易懂，功能强大，适合于各种硬件平台，与常见的高级语言不一样的是：C 语言兼有高级语言和低级语言的功能。既可用于系统软件的开发，也适合于应用软件的开发。

C 语言的特点还表现在以下几个方面：

1. 程序设计结构化

结构化就是将程序的功能进行模块化，每一个模块具有不同的功能，程序将一些不同功能的模块有机的组合在一起，通过模块之间的相互协同工作，共同完成程序所要完成的任务。这种模块化的程序设计方式使得 C 语言程序易于调试和维护。

了解和掌握结构化程序设计的思想，有助于提高我们分析问题和解决问题的能力。

2. 运算符丰富

C 语言共有 34 种运算符。它把括号、赋值、逗号等都作为运算符处理，从而使 C 语言的运算类型极为丰富，可以实现其他高级语言难以实现的一些运算。

3. 数据类型丰富

C 语言除了具有系统本身规定的一些数据类型外，还允许用户定义自己的数据类型，以满足

程序设计的需要。

4. 书写灵活

只要符合 C 语言的语法规则，程序书写的格式并不受到严格的限制。

实际编写程序时并不提倡这样做，而是要求根据语法规则按缩进格式书写程序。

5. 适应性广

C 语言程序生成的目标代码质量高，程序执行效率高，与汇编语言相比，用 C 语言写的程序可移植性好。

6. 关键字简洁

在 C 语言中，关键字有其特殊的意义和作用，不允许用户将其用作其他用途，所有关键字都必须是小写英文字母。ANSI C 规定 C 语言共有 32 个关键字，这 32 个关键字可以分为以下 4 大类：

（1）数据类型关键字 12 个；

（2）控制语句关键字 12 个；

（3）存储类型关键字 4 个；

（4）其他关键字 4 个。

C 语言关键字及其作用如表 1-1 所示。

表 1-1　　C 语言的关键字及作用

序号	类型	关键字					作　用
1	数据类型 9 个	char	int	float	double	void	用于数据类型的声明
		short		long			用于声明整型数据的大小
		signed		unsigned			用于声明整型数据在正负坐标上的区间
	自定义的数据类型 3 个	struct					声明结构数据类型
		union					声明联合数据类型
		enum					声明枚举数据类型
2	控制类型 12 个	if	else	switch	case	default	用于分支结构
		for	while	do～while			用于循环结构
		continue					结束本次循环，进入下一轮循环
		break					直接跳出循环结构或分支结构
		goto					直接转移到指定的语句处
		return					返回到函数调用处
3	变量存储类型 4 个	auto					声明自动变量（可缺省）
		extern					声明外部变量
		register					声明寄存器变量
		static					声明静态变量
4	其他类型（4 个）	const					声明只读变量
		sizeof					计算数据类型长度
		typedef					给自定义数据类型取别名等
		volatile					变量在程序执行中可被隐含地改变

需要说明的是，除了上述的关键字以外，不同的实现环境对 C 语言的关键字有所扩充，并且扩充的关键字会因实现环境的不同而不同，读者只需要从使用的实现环境中去了解即可，在此不多加叙述，扩充的关键字只适合于特定的实现环境。

7. 控制结构灵活

C 语言的程序结构简洁高效，使用方便、灵活，程序书写自由。C 语言一共有 9 种控制结构，可以完成复杂的计算，9 种控制结构及作用如表 1-2 所示。

表 1-2　　C 语言的 9 种控制结构及作用

关 键 字	作　用	关 键 字	作　用	关 键 字	作　用
goto	直接转移	for	循环语句	break	直接跳出循环结构或分支结构
if	条件分支	do～while	循环语句	continue	结束本次循环，开始下一轮循环
switch	多路分支	while	循环语句	return	返回到函数调用处

表 1-2 所示的关键字与表 1-1 所示的关键字的意义和作用是相同的，表 1-2 中的 continue 只能用于循环，而 break 可以用于循环或 switch 多路分支。

了解 C 语言上述的特点，对学习和掌握好 C 语言程序设计很有帮助。虽然 C 语言程序对书写的要求没有太多的限制，只要符合语法规则就行，但在这里我们强调程序书写必须规范，特别对初学者，这一点很重要。一个书写规范、整齐的 C 语言程序能够帮助程序员快速读懂程序所表达的思想，同时也能更清晰地将程序设计的意图正确地表达出来。

1.3　简单的 C 语言程序

为了说明 C 语言源程序结构的特点，先看以下几个程序。这几个程序由易到难，表现了 C 语言源程序在组成结构上的特点。虽然有关内容还未介绍，但可从这些例子了解到组成一个 C 语言源程序的基本部分和书写格式。

【例 1-1】 编写程序，在屏幕上输出“Hello,World!”的字符串。

程序如下：

```
/* example1_1.c  在屏幕上输出字符串*/
#include <stdio.h>
main()
{
    printf("Hello,World!\n");
}
```

程序运行结果：

```
Hello,World!
```

程序说明：

1. #include 称为文件包含命令。其作用是把系统目录下的头文件 stdio.h 包含到本程序中，成为本程序的一部分。这里被包含的文件是由系统提供的，所以用尖括号< >来标定，其扩展名为.h，也称为头文件或首部文件。

C 语言系统提供的头文件中包括了各种标准库函数的函数原型，因此，凡是在程序中调用某个库函数时，都必须将该函数原型所在的头文件包含进来。在这里包含的文件是 stdio.h，该文件里的函数主要是处理数据流的标准输入/输出，在此表示在程序中要用到这个文件中的函数。“#”只是一个标志。

2. main 是主函数的函数名，表示这是一个主函数。1 个可执行的 C 语言源程序只允许有 1 个 main 函数。

3. printf()函数是一个库函数，其函数原型在头文件 stdio.h 中，该函数的功能是将圆括号内的内容输出到显示器。

4. main()函数中的内容必须放在一对花括号“{}”中。

【例 1-2】请从键盘输入一个角度的弧度值 x，计算该角度的余弦值，将计算结果输出到屏幕。

程序如下：

```
/* example1_2.c  计算角度的余弦*/
#include<stdio.h>
#include<math.h>
main()
{
     double x,s;
     printf("Please input value of x: ");
     scanf("%lf",&x);
     s=cos(x);
     printf("cos(%lf)=%lf\n",x,s);
}
```

程序运行结果：

```
Please input value of x: 0↵
cos(0.000000)=1.000000
```

注意 在这里要求输入的角度为弧度值，如果要计算 cos(180°)，则要输入 π 的值。再次运行上面这个程序。

程序运行结果：

```
Please input value of x: 3.1415926↵
cos(3.141593)=-1.000000
```

程序说明：

1. 程序包含了两个头文件：stdio.h、math.h。

2. 在 main 函数中定义了两个双精度实数型变量 x、s。

3. printf("Please input value of x:"); 用于显示提示信息。

4. scanf(" %lf ",&x); 用于从键盘获得一个实数 x。

5. s=cos(x); 用于计算 x 的余弦，并把计算结果赋给变量 s。

6. printf ("cos(%lf)=%lf\n",x,s); 将计算结果输出到屏幕。双引号中有两个格式字符“%lf”，分别对应着 x 和 s 两个输出变量。

7. 程序运行后会在屏幕的左上方显示提示信息，要求用户从键盘输入一个用弧度值表示的角度值 x，接下来程序会计算该角度的余弦，并将计算结果输出到屏幕。

在本例中，使用了 3 个库函数：输入函数 scanf()，余弦函数 cos()，输出函数 printf()。余弦函数 cos()是数学函数，其函数原型在头文件 math.h 中，scanf()和 printf()是标准输入/输出函数，其函数原型在头文件 stdio.h 中。

需要说明的是，在有些编译环境中，可以省去 scanf()和 printf()这两个函数的包含命令。所以在例 1-1 和例 1-2 中可以省略文件包含命令#include<stdio.h>。但我们不建议这么做，很显然，任何时候明确的表达对程序的理解是有益的，一般情况下，只要程序中用到了头文件中的函数原型，都应该将相应的头文件包含进来，这是一个良好的习惯，否则，有可能由于编译系统的不同而发生语法错误。希望读者在开始学习的时候就养成良好的书写习惯。

【例 1-3】设计一个加法器，能实现两数的相加。通过调用该加法器，计算两数的和。

```
/* example1_3 两数相加的加法器*/
#include<stdio.h>
```

```
int add(int x, int y); /*加法器的函数原型声明*/
main()
{
     int a, b, c;
     printf ("please input value of a and b:\n");
     scanf("%d %d", &a, &b);
     c=add(a,b);
     printf ("max=%d\n",c);
}
int add(int x, int y) /*加法器的函数定义*/
{
     return(x+y);
}
```

程序运行结果：

```
please input value of a and b:
5 9↵
max=14
```

程序说明：

1. 在例 1-3 中的主函数体分为两部分，一部分为说明部分，另一部分为执行部分。说明是指变量的类型说明（例 1-1 中未使用任何变量，因此无说明部分）。C 语言规定，源程序中所有用到的变量都必须先说明，后使用，否则会出现语法错误。这是编译型高级程序语言的一个特点，说明部分是 C 语言源程序结构中很重要的组成部分。

2. 程序语句 c=add(a,b);是通过调用加法器 add()来完成（a+b）的计算，并将计算结果赋给变量 c。

3. 运行程序时，首先在显示屏幕上显示字符串 please input value of a and b:，提示用户从键盘输入 a 和 b 的值，这是由执行部分的第一行完成的。用户在提示下从键盘上键入两个数，如 5↵9↵，（或 5 9↵），接着在屏幕上会显示出计算结果：max = 14。

1.4　C 语言程序的结构

上面所述的 3 个 C 语言程序范例虽然功能简单，但却包含了 C 语言程序的基本组成部分，概括来说，一个 C 语言程序可由下面几个不同的部分组合而成：

1. 文件包含部分；
2. 预处理部分；
3. 变量说明部分；
4. 函数原型声明部分；
5. 主函数部分；
6. 函数定义部分。

对于上面所述 C 语言程序的 6 个部分，有几点需要说明：

1. 并不是每一个程序都必须包含上面的 6 个部分，一个最简单的 C 语言程序可以只包含文件的包含部分和主函数部分。

2. 每一个 C 语言程序都必须有且仅有一个主函数，主函数的组成形式如下所示。

```
main()
{
   变量说明部分
   程序语句部分
}
```

3. 每一个 C 语言程序可以有 0 个或多个自定义的函数，自定义函数的形式同主函数形式一样。

```
<函数的返回值> <函数名>(<参数列表>)
```

```
{
    变量说明部分
    程序语句部分
}
```

4. 每一条语句由分号结束。

对于 C 语言程序中各部分的作用、使用方法和采用的什么语句来完成程序设计，我们将通过后续章节的学习，了解基本表达式、控制结构语句的作用，并通过了解模块化程序设计的思想和方法，掌握基本的 C 语言程序设计方法。

1.5 C 语言程序的执行

用程序语言编写的程序称为源程序（Source Program），实际上计算机本身并不能直接理解这样的语言，必须将程序语言翻译成机器语言，计算机才能理解程序。将源程序翻译成机器语言的过程称为编译，编译的结果是得到源程序的目标代码（Object Program）；最后还要将目标代码与系统提供的函数和自定义的过程（或函数）链接起来，就可得到机器可执行的程序。机器可执行的程序称为可执行程序或执行文件。

1.5.1 源程序翻译

C 语言源程序的后缀名为.c。它是不能直接在计算机上运行的，必须通过机器翻译成目标代码，再将目标代码链接成可加载模块（可执行文件），才能在计算机上运行。这种把源程序翻译成目标代码的程序被称为编译器或翻译器。适合 C 语言的编译器不止一种，不同的机器、不同的操作系统可能会有一种或多种不同的编译器。C 语言源程序的翻译过程如图 1-1 所示，它由词法分析器、语法分析器和代码生成器 3 部分组成。

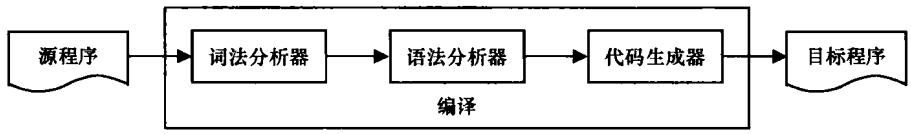

图 1-1 程序语言的翻译过程

1. 词法分析器（Lexical Analyzer）

词法分析器主要是对源程序进行词法分析，它是按单个字符的方式阅读源程序，并且识别出哪些符号的组合可以代表单一的单元，并根据它们是否是数字值、单词（标识符）、运算符等，将这些单词分类。词法分析器将词法分析结果保存在一个结构单元里，这个结构单元称为记号（Token），并将这个记号交给语法分析器，词法分析会忽略源程序中的所有注释。

2. 语法分析器（Parser）

语法分析器直接对记号进行分析，并识别每个成分所扮演的角色。这些语法规则也就是程序设计语言的语法规则。

3. 代码生成器（Code Generator）

代码生成器将经过语法分析后没有语法错误的程序指令转换成机器语言指令。

例如，假定编写了一个名为 mytest 的程序，源程序的全名为 mytest.c，用 Microsoft C 编译器，在命令方式下，可以采用下面这样的方式对 mytest.c 进行编译：

```
cl -c mytest.c↵
```

如果源程序没有错误，就会生成一个名为 mytest.obj 目标代码程序。其他程序语言也会有类似的命令将源程序翻译成目标代码，具体的命令与每种程序语言的编译器有关。

1.5.2 链接目标程序

通过翻译产生的目标代码程序尽管是机器语言的形式，但却不是机器可以执行的方式，这是因为为了支持软件的模块化，允许程序语言在不同的时期开发出具有独立功能的软件模块作为一个单元，一个可执行的程序中有可能包含一个或多个这样的程序单元，这样可以降低程序开发的低水平重复所带来的低效率。因此，目标程序只是一些松散的机器语言，要获得可执行的程序，还需将它们链接起来。

程序的链接工作由链接器（Linker）来完成。链接器的任务就是将目标程序链接成可执行的程序（或称载入模块），这种可执行的程序是一种可存储在磁盘存储器上的文件。

例如，假设已对源程序 mytest.c 进行编译后生成了目标代码程序 mytest.obj，我们可以利用链接器生成可执行代码，在命令方式下，将用下面这样的方式来链接程序：

```
link /out:mytest.exe mytest.obj↵
```

如果不发生错误，就会生成一个名为 mytest.exe 的加载模块，也就是可执行的代码程序。最后，可以通过操作系统将这个加载模块加载入内存，执行程序的进程。

上面对程序进行编译、链接都只针对了一个源程序文件，实际上，可以将多个源程序文件通过编译，链接成一个可执行文件。

例如，假定有 3 个源程序：file1.c，file2.c 和 file3.c，每一个源程序都包含有不同的函数或过程，在命令方式下可先用编译器对 3 个源程序进行编译：

```
cl  -c file1.c↵
cl  -c file2.c↵
cl  -c file3.c↵
```

分别得到 3 个目标程序：file1.obj，file2.obj 和 file3.obj。接下来可用链接器将 3 个目标程序进行链接。

```
Link /out:mytest.exe file1.obj file2.obj file3.obj↵
```

可得到一个可执行的程序：mytest.exe。

对于程序的编译、链接，有必要强调以下几点。

1. 并不是每一个目标程序都可以链接成可执行程序。

2. 被链接成可执行程序的目标程序中，只允许在一个程序中有且仅有一个可被加载的入口点，即只允许在一个源程序中包含一个 main()函数。在上面的范例中这个可被加载的入口点在源程序 file1.c 中。

3. 对于具体的程序语言，编译、链接程序的方法会有所不同，针对某一种程序语言的编译器，不可以用于对另一种源程序语言的编译。

4. 上面对 C 语言进行编译、链接的方式并不是唯一的，它允许有一些其他的变化，具体可参考各编译器的使用说明。

总之，完成一个 C 语言程序的完整过程主要包括 4 个部分：编辑、编译、链接和加载，如图 1-2 所示。

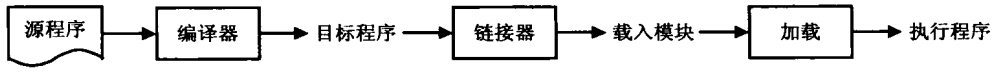

图 1-2 完整的程序生成过程

一旦生成了可执行程序，就可以反复地被加载执行，而不需要重新编译、链接，如果修改了源程序，也不会影响到已生成的可执行程序，除非对修改后的源程序重新编译和链接，生成一个新的可执行程序。

1.5.3 集成开发工具

显然，用命令方式来编译、链接生成可执行的程序，并不是很方便，尤其是源程序的编辑。一般地，纯文本编辑器都可以输入源程序，如果编译时有错误，就必须回到编辑器修改程序。如此反复，源程序的编辑不是很方便，而且程序开发效率不高，目前已很少采用这种方法来编辑 C 语言源程序，大多采用集成开发工具来开发 C 语言程序。

程序的集成开发工具是一个经过整合的软件系统，它将编辑器、编译器、链接器和其他软件单元集合在一起，在这个工具里，程序员可以很方便地对程序进行编辑、编译、链接以及跟踪程序的执行过程，以便寻找程序中的问题。

适合 C 语言的集成开发工具有许多，如 Turbo C、Microsoft C、Visual C++、Dev C++、Borland C++、C++ Builder、GCC 等。这些集成开发工具各有特点，分别适合于 DOS 环境、Windows 环境和 Linux 环境，几种常用的 C 语言开发工具的基本特点和所适合的环境如表 1-3 所示。

表 1-3　　几种常用的 C 语言开发工具

开发工具	运行环境	各工具的差异	基本特点
Turbo C	DOS	不能开发 C++语言程序	（1）符合标准 C （2）各系统具有一些扩充内容 （3）能开发 C 语言程序（集程序编辑、编译、链接、调试、运行于一体）
Borland C	DOS		
Micrsoft C	DOS		
Visual C++	Windows	能开发 C++语言程序（集程序编辑、编译、链接、调试、运行于一体）	
Dev C++	Windows		
Borland C++	DOS、Windows		
C++ Builder	Windows		
GCC	Linux		

从表 1-3 中可以看出，有些集成开发工具不仅仅适合于开发 C 语言程序，还适合开发 C++语言程序。这些既适合于 C 语言又适合于 C++语言的开发工具，一开始并不是为 C 语言而写的，而是为 C++语言设计的集成开发工具，但因为 C++语言是建立在 C 语言的基础之上，C 语言的基本表达式、基本结构和基本语法等方面同样适合 C++语言，因此，这些集成开发工具也能开发 C 语言程序。

事实上，表 1-3 所示的几种适合于 DOS 环境的集成开发工具，也有适合于 Windows 环境的版本，但在计算机的图形用户界面还没有像现在这样普遍和成熟的时候，人们大多都是采用 DOS 环境下集成开发工具，如 Turbo C 2.0，它既能作为初学者的一个实验工具，同时还是很多专业人员进行程序开发的首选工具。

另外，还有一些小型的集成开发工具比较适合于学习者使用，如 Turbo C/C++ for Windows，它支持 C、C++、Windows C 等程序的编辑、编译、调试、运行，为 C 语言的学习提供了方便。

1.6 本章小结

本章简要介绍了 C 语言的特点和发展过程，介绍了 C 语言程序的基本组成部分，使读者对 C 语言程序的组成部分有一个初步的了解；还介绍了 C 语言程序的开发过程：

1. 编辑源程序——主要是编写源程序；
2. 编译源程序——用编译器对源程序进行语法检查，生成目标代码程序；
3. 生成可执行的程序——对目标代码进行链接，生成可执行的文件；
4. 检验——运行上面生成的可执行程序，检验该程序是否符合设计要求。

最后，介绍了几种适合开发 C 语言程序的开发工具，使大家明白开发 C 语言程序可以有多种途径，学习者可以根据自己的喜爱和用途选择不同的工具。作为 C 语言的初学者，建议选择那些操作简单、易学的开发工具来作为 C 语言程序设计的实验平台，这样可以将注意力更好地关注在 C 语言程序设计的方面，而不是更多地去了解开发工具本身。当学习者对 C 语言程序设计有了一定的认识和了解之后，可以根据程序设计的目的和要求的不同，选择不同的开发工具来完成 C 语言程序的开发。

学习 C 语言程序设计的一个重要环节就是要又动手又动脑地做实验，衷心地希望每一位学习 C 语言程序设计的人，都能够从程序设计实验中有所收获，获得快乐。

习　　题

【题 1.1】请查阅文献，了解 C 语言的发展过程，ANSI C 是如何形成的?

【题 1.2】由 B.W.Kernighan 和 D.M.Ritchie 合著的《The C Programming Language》，是否定义了完整的标准 C 语言?

【题 1.3】请查阅文献，了解“K&R”标准指的是什么?

【题 1.4】请查阅文献，了解标准 C 语言和扩展 C 语言是什么关系?

【题 1.5】请查阅文献，了解标准 C 语言中的关键字共有多少个? 是否可以用大写字母表示?

【题 1.6】请查阅文献，了解怎样在程序中使用系统提供的函数?

【题 1.7】请查阅文献，了解由系统提供的函数是放在什么文件中?

【题 1.8】一个 C 语言程序可由哪些不同的部分组合而成?

【题 1.9】请选择 Turbo C 2.0 集成开发工具，自己动手将它安装到计算机中，然后分别输入本章的 3 个程序，再对程序进行编译、运行，了解程序的开发过程。

【题 1.10】请选择 Visual C++ 6.0 集成开发工具，自己动手将它安装到计算机中，然后分别输入本章的 3 个程序，再对程序进行编译、运行，了解程序的开发过程。

【题 1.11】请选择 Dev C++集成开发工具，自己动手将它安装到计算机中，然后分别输入本章的 3 个程序，再对程序进行编译、运行，了解程序的开发过程。

【题 1.12】请选择一种集成开发工具（除 Turbo C 2.0、Visual C++ 6.0、Dev C++以外），自己动手将它安装到计算机中，然后分别输入本章的 3 个程序，再对程序进行编译、运行，了解程序的开发过程。

【题 1.13】拓展实验训练 1。在你所选定的开发环境下，对本章的第 1 个程序范例进行改造，使其能完成你的预期设计目标，如输出其他的内容。记录所获得的收获和存在的问题。

【题 1.14】拓展实验训练 2。在你所选定的开发环境下，对本章的第 2 个程序范例进行改造，使其能完成你的预期设计目标，如计算角度的正弦值、正切值等。记录所获得的收获和存在的问题。

【题 1.15】拓展实验训练 3。在你所选定的开发环境下，对本章的第 3 个程序范例进行改造，使其能完成你的预期设计目标，如计算两数的差、两数的积、两数的商等。记录所获得的收获和存在的问题。

第2章 基本的程序语句

C 语言的变量有不同的数据类型，每一种数据类型的取值空间是不同的。因此，不同数据类型的变量，它们的取值空间也是不相同的。

ANSI C 中对数据类型的取值空间只规定了一个范围，对于具体哪种数据类型的变量，其取值范围是多少，则会因实现环境的不同而稍有不同。当然，这些不同所带来的差异相对来说比较小，对于程序的设计一般不会带来很大的影响，因为程序的基本语句和基本操作都还是相同的。如果有必要，只要在程序设计之前，了解一下所使用的 C 语言开发工具对数据类型的取值空间所做的规定。

本章将介绍 C 语言中的数据类型和它们的取值范围以及简单的表达式语句，通过不同的数据类型和简单的表达式语句，能够编写一些简单的、顺序结构的程序。

2.1 预备知识[①]

2.1.1 定点数和浮点数的概念

在 C 语言程序中，各种类型的变量大多是以十进制的形式来描述的，但实际上这些变量在计算机中是以二进制的形式存储的，C 语言允许对二进制数直接进行位操作来完成特殊的要求。因此，在学习 C 语言程序设计之前，有必要了解变量的十进制数和二进制数的关系及其它们之间的转换规则。

计算机中的数，通常是用定点数或浮点数表示。

定点数是小数点位置固定的数，整型数和纯小数通常是用定点数来表示的，分别称为定点整数和定点纯小数。

浮点数的小数点位置是不固定的，可以浮动。不论是定点数还是浮点数，小数点都不单独占 1 个二进制位。对于既有整数部分、又有小数部分的数，一般用浮点数表示。

如 234，6 278，0.194 2，0.005 6 等是定点数；而 629 314，38.96，2 836.635 等是浮点数。

对于定点整型数，小数点的位置默认在数值位最低位的右边。计算机能表示的定点整型数的大小范围并不是任意的，它与计算机本身的字长有关，还与程序语言的实现环境有关。

例如，24，678，9 325 等均为定点整型数。

对于定点纯小数，小数点的位置固定在符号位与最高数值位之间，显然，定点纯小数所能表示数的范围较小，并不能满足实际问题的需要。定点纯小数的精度与计算机本身的字长有关，还与程序语言的实现环境有关。

① 如果读者已经具有了这方面的知识，可以跳过这一节，直接进入下一节内容的学习。

例如，0.423 7，−0.619 83，0.052 等均为纯小数。

对于浮点数，因为既有整数部分又有小数部分，所以能表示的数值范围更大，在浮点数表示法中，小数点的位置是可以浮动的。

例如，61.329 就是一个浮点数，它还可以表示成如下几种形式：

$61\ 329\times10^{-3}$，$6\ 132.9\times10^{-2}$，613.29×10^{-1}，$6.132\ 9\times10$，$0.613\ 29\times10^{2}$。

在大多数计算机中，存储浮点数的时候，都会把浮点数转化成两个部分：整数部分和纯小数部分。因此，计算机存储浮点数 61.329 时，最终要将其转化成用二进制数来表示。

一般而言，计算机的存储数据是以 8 个位（bit）为一个单位的，每 8 个位构成一个字节。

2.1.2　整型数的二进制表示

对于整型数而言，又分为有符号的和无符号的两种，有符号的整型数既可以是正数，也可以是负数，正负号由字节的最高位来表示，0 表示正数，1 表示负数。

（1）有符号的二进制整型数

① 1 个字节表示的数。

如 10110100，其最高位的 1 为符号位，因此，（10110100）的十进制数为：$-(2^5+2^4+2^2)=-52$。

而 00110100 的十进制数为：$2^5+2^4+2^2=52$。

必须提醒的是，为了不浪费计算机的存储空间，对于“正零”和“负零”有不同的处理，对于“正零”（00000000），代表数字 0；而“负零”（10000000），代表−128。

② 2 个字节表示的数。

如 10110100 10101101，其最高位的 1 为符号位，因此，（10110100 10101101）的十进制数为：$-(2^{13}+2^{12}+2^{10}+2^7+2^5+2^3+2^2+2^0)=-13\ 485$。

而 00110100 10101101 的十进制数为：$2^{13}+2^{12}+2^{10}+2^7+2^5+2^3+2^2+2^0=13\ 485$。

对于“正零”（00000000 00000000），代表数字 0；而“负零”（10000000 00000000），代表−32 768。

③ 4 个字节表示的数。

如 10000000 00000100 00000001 10000001，其最高位的 1 为符号位，因此，（10000000 00000100 00000001 10000001）的十进制数为：$-(2^{18}+2^8+2^7+2^0)=-262\ 529$。

而 00000000 00000100 00000001 10000001 的十进制数为：$(2^{18}+2^8+2^7+2^0)=262\ 529$。

对于“正零”（00000000 00000000 00000000 00000000），代表数字 0；而“负零”（10000000 00000000 00000000 00000000），代表−2 147 483 648。

（2）无符号的二进制整型数

其最高位的 0 或 1 不再代表符号位，而是代表具体的数值。

① 1 个字节表示的数。

如 10110100，它的十进制数为：$(2^7+2^5+2^4+2^2)=180$。

② 2 个字节表示的数。

如 10110100 10101101，它的十进制数为：$(2^{17}+2^{13}+2^{12}+2^{10}+2^7+2^5+2^3+2^2+2^0)=46\ 253$。

③ 4 个字节表示的数。

如 10000000 00000100 00000001 10000001，它的十进制数为：$(2^{31}+2^{18}+2^8+2^7+2^0)=2\ 147\ 746\ 177$。

显然，对于有符号的整型数和无符号的整型数，它们的取值范围如表 2-1 所示。