新起点电脑教程·程序设计基础

C语言程序设计基础与上机指导

李岩 主编

选材适当、通俗易懂
内容全面、示例丰富
结构清晰、目标明确
针对性强、定位准确

清华大学出版社

新起点电脑教程　程序设计基础

C 语言程序设计基础与上机指导

李　岩　主编

清华大学出版社
北　京

内 容 简 介

C 语言是一种高级编程语言，同时也具有很多低级语言编程的灵活性。它支持面向过程的编程方式，是学习计算机语言编程最基础的学习内容。在 C 语言的基础之上诞生了很多优秀的程序通过学习 C 语言将使读者对计算机编程有更深刻的认识。

本书作者在多年教学经验基础上，根据学生的认知规律精心组织了本教材内容，并通过大量有现实意义的例题，循序渐进地介绍了 C 语言程序设计的有关概念和编程技巧。书中例题都经过了仔细的调试，配有大量的上机实训题和课后习题，并为教师配有上机实训参考答案和课后习题参考答案

本书概念清晰、例题丰富、深入浅出、知识结构及深度合理，可作为高职高专院校的教材，也可作为计算机培训班的教材以及自学者的参考书。

图书在版编目(CIP)数据

C 语言程序设计基础与上机指导/李岩主编. —北京：清华大学出版社，2006.3

(新起点电脑教程　程序设计基础)

ISBN 7-302-12289-X

Ⅰ.C…　Ⅱ.李…　Ⅲ.C 语言—程序设计—教材　Ⅳ.TP312

中国版本图书馆 CIP 数据核字(2005)第 158822 号

出 版 者：清华大学出版社　　**地　　址**：北京清华大学学研大厦
http://www.tup.com.cn　　**邮　　编**：100084
社 总 机：010-62770175　　**客户服务**：010-62776969

组稿编辑：凌宇欣
文稿编辑：桑任松
排版人员：李月菊
印 装 者：北京市清华园胶印厂
发 行 者：新华书店总店北京发行所
开　　本：185×260　**印张**：21　**字数**：498 千字
版　　次：2006 年 3 月第 1 版　2006 年 3 月第1 次印刷
书　　号：ISBN 7-302-12289-X/TP·7898
印　　数：1～5000
定　　价：29.00 元

前　言

计算机技术的飞速发展，促进了计算机基础教育的发展。根据我国当前教学改革和建设的需要，教育部提出了“计算机文化基础”、“计算机技术基础”和“计算机应用基础”三个层次的教学体系。计算机程序设计语言是高等院校各专业学生的一门基础课程，属于计算机技术基础教育，是当代大学生必须掌握的一种应用技能。

早期的C语言主要是用于UNIX系统。由于C语言的强大功能和各方面的优点逐渐被人们所认识，到了20世纪80年代，C语言开始进入其他操作系统，并很快在各类大、中、小和微型计算机上得到了广泛的使用。成为当代最优秀的程序设计语言之一。

C语言是一种结构化语言。它层次清晰，便于按模块化方式组织程序，易于调试和维护。C语言的表现能力和处理能力极强。它不仅具有丰富的运算符和数据类型，便于实现各类复杂的数据结构；它还可以直接访问内存的物理地址，进行位(bit)一级的操作。由于C语言实现了对硬件的编程操作，因此C语言集高级语言和低级语言的功能于一体，既可用于系统软件的开发，也适合于应用软件的开发。此外，C语言还具有效率高，可移植性强等特点，因此广泛地移植到了各种类型计算机上，从而形成了多种版本的C语言。

本教材是在作者多年从事C语言教学的基础上编写而成的，作者根据多年的教学经验和学生的认知规律精心组织教材内容，做到内容丰富、深入浅出、循序渐进，力求达到可读性、实用性、先进性。通过大量例题使读者能迅速掌握有关概念及一些编程技巧，书中的例题都经过了仔细的调试；每章都配有大量的上机实训题，通过上机实践，使读者掌握程序设计与调试的方法，提高动手能力，掌握所学内容；每章还配有大量的课后习题，供读者课外巩固所学的内容。

本教材以应用为中心，以初学者为对象，以提高程序设计能力为宗旨，为读者学习编写C语言程序提供了捷径。

本教材为教师配有上机实训参考答案和课后习题参考答案。

本教材适合于大专院校学生、成人继续教育及自学人员使用。

建议本教材授课时数为40～60学时，另外还要安排大量的上机练习，以巩固所学知识。

本教材由李岩担任主编，张翠霞、李松梅、邵淑霞、何红玉、代宏亮参加编写。本书的第1、4、7、8章由李岩编写；第2章由张翠霞编写；第3章由李松梅、代宏亮编写；第5、9、11章由邵淑霞编写；第6、10章由何红玉编写。

由于计算机技术发展迅速，加上作者水平有限，书中难免存在缺点和错误，请读者不吝指正。

编　者

目　录

第 1 章　C 语言概述

教学提示：本章主要介绍 C 语言和 C 程序设计的初步知识，这些知识在后续章节中还会出现，学习完这些内容后，读者对 C 程序将会有一个初步的了解。

教学目标：熟悉 C 程序的基本结构和书写风格；掌握 C 语言关键字和标识符的命名方法，了解 C 编译系统提供的标题文件的功能，学会用赋值语句和格式输入/输出函数编制简单的 C 程序，熟悉 Turbo C 集成环境的使用。

1.1　C 语言的产生和发展

C 语言是一种面向过程的通用程序设计语言，它以表达简明、使用灵活、结构化的流程控制、丰富的数据结构和操作符集合、良好的程序可移植性和高效率的目标代码为特征。C 语言不仅是一种高级语言，还兼有低级语言的功能，因此既可用于编写系统程序，也可用于编写不同领域的应用程序。

C 语言的祖先是 ALGOL60(ALGOrithm Language)。ALGOL60 具有可读性好和可移植性强的特点，但它不能直接对硬件进行操作，不宜用来编写系统程序。人们开始考虑一种集高级语言和低级语言功能于一身的语言，以便可以用它编写可读性和可移植性都比较好的系统程序。

1963 年，英国的剑桥大学和伦敦大学首先将 ALGOL60 发展成 CPL，1967 年，剑桥大学的 Martin Richards 将 CPL 改制成 BCPL，1970 年，美国贝尔实验室的 Ken Thompson 将 BCPL 修改成 B 语言，并用 B 语言开发了第一个高级语言 UNIX 操作系统，在 DEC 公司的 PDP-7 小型机上运行。

1972 年，Ken Thompson 在 UNIX 系统上的亲密合作者 Dennis M. Ritchie 将 B 语言修改设计成 C 语言。C 语言既保持了 BCPL 和 B 语言的精炼和接近于硬件的特点，也克服了它们过于简单、数据无类型等缺点。1973 年，Ken Thompson 和 Dennis M. Ritchie 又合作将 1969 年用汇编语言编写的 UNIX 操作系统改用 C 语言编写，其中 C 语言代码占 90%以上，只保留了少量汇编语言代码。这样就使得 UNIX 操作系统向其他类型的机器上移植变得相当简单。到 20 世纪 70 年代中期，UNIX 操作系统和 C 语言作为软件设计师的良好工具开始风靡世界。

1978 年，以 UNIX 第 7 版中的 C 编译程序为基础，Brain W. Kernighan 和 Dennis M. Ritchie 合著了影响深远的名著《The C Programming Language》。这本书中介绍的 C 语言成为后来广泛使用的 C 语言版本的基础，称为 K&R C 语言。其后的十几年中，适用于不同机种和不同操作系统的 C 编译系统相继问世，从而把 C 语言的应用推向了更加广泛普及的阶段。1983 年美国国家标准局 ANSI 制定了 C 语言标准。这个标准不断完善，并从 1987

年开始实施，称为 ANSI C。1988 年，K&R 修改了他们的经典著作《The C Programming Language》，按 ANSI C 标准重新编写了该书。现在一般称 ANSI C 为新标准或现代 C，K&R C 为旧标准或传统 C。此后陆续出现的各种 C 语言版本，如 Microsoft C 5.0、9.0，Turbo C 2.0、3.0，Quick C 等都是与 ANSI C 兼容的版本。它们的语法和语句功能是一致的，差异表现在各自的标准函数库中收纳的函数种类、格式和功能上，尤其是图形函数库的差异更大一些。

C 语言是一种面向过程的语言，意思是用 C 语言编程时，必须按照算法的实现过程逐条语句编写，通知计算机一步一步怎么做。进入 20 世纪 80 年代后，面向对象的程序设计概念日益普及。所谓面向对象，是通过类和对象把程序所涉及的数据结构和对它施行的操作有机地组织成模块，将数据本身和对数据的处理细节进行最大限度的封装，从而使开发出来的软件易重用、易修改、易测试、易维护、易扩充。正如其他传统的程序设计语言都在发展自己面向对象的新版本一样，C 语言也在发展的同时，朝着支持面向对象程序设计(Object Oriented Programming，OOP)的方向迈出了步伐。1986 年，美国 AT&T 的贝尔研究所的 Bjarne Stroustrup 推出了 C 语言的超集 C++语言，也叫“带类的 C”。在软件开发领域，C 和 C++一直是最有生命力的程序设计语言，大多数优秀的程序员都是 C/C++/Visual C++的高手，大多数成功的软件也是用 C 或 C++语言编制的。

随着信息时代的到来，由于设计 Internet 上的 Web 浏览器的需要，1994 年出现了 Java 语言，它不仅支持 OOP，而且具有软硬件平台无关性的特点，适合于程序员进行网络开发。Java 脱胎于 C++，被称为 C++的衍生语言。2000 年，Microsoft 推出了其下一代计算计划——Microsoft.NET，它是一个具有公共语言子集的开发平台，实现了多种语言及其类库的无缝集成，使应用程序的开发更容易、更简单，并可能成为下一代网络通信标准。C#是专为这一平台推出的全新语言，它也派生于 C 和 C++，并具有语法简洁、面向对象、与 Web 紧密结合、卓越的安全性能、灵活性和兼容性俱佳等特点，成为.NET 平台一流的网络编程工具。

C、C++、Visual C++、C#的发展轨迹反映了人类信息处理能力的不断深化。

1.2 C 程序的构成简述

用 C 语言编写的程序称为 C 程序。一个程序有一个或若干个函数构成。程序中至少应包括一个 main 函数。函数是 C 程序的基本单位。

在有多个函数组成的 C 程序中，函数定义的顺序与其被引用的先后次序无关，即函数的定义次序不影响其引用次序。由此可以看出，一个 C 程序实质上是一系列相互独立的函数的定义，函数之间只存在引用和被引用的关系。

一个 C 程序总是从 main 函数开始执行，而不论 main 函数在程序中的位置如何。其他函数由 main 函数直接或间接调用执行。

一个 C 程序可以由一个文件组成，也可以由多个文件组成。每一个文件中包含一个或多个函数。当一个 C 程序由一个或多个文件组成时，每个文件分别进行编译(生成.obj 文件)，再通过连接合并为一个可执行文件(生成.exe 文件)，然后运行。

1.2.1　基本单词

一个 C 语句由若干个基本单词组成。C 语言共有五种基本单词，即关键字(也称保留字)、标识符、常数、操作符和分隔符。例如，语句

```
float a,b,S;
```

中，float 是关键字(代表数据类型)，a、b 和 S 是标识符(表示变量)；又如，语句

```
S=a*b;
```

中，“=”和“*”是操作符。

1. 关键字

关键字是 C 语言中有特定意义和用途、不得作为它用的字符序列，其中 ANSI C 标准规定的关键字有 32 个，如下所示。

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

注意： 所有的 C 关键字都必须小写。

2. 标识符

标识符用来表示变量名、数组名、函数名、指针名、结构名、联合名、枚举常数名、用户定义的数据类型名及语句标号等用途的字符序列，可由 1～32 个字符组成，第一个字符必须是字母或下划线，后面的字符可以是字母、数字或下划线。例如

AB，Ab，aB，ab，A_b，_ab，ab_，s2d，W_length

等都是正确的标识符，而

A+B，A'B，A.B，2abc，α，β，d%

等是错误的标识符。

注意： 标识符不能与 C 关键字相同，而且区分大小写。例如 AB 和 Ab 是两个不同的标识符，ELSE 可以作为标识符，它不会与 C 关键字 else 混淆。

顺便指出，在 C 编译系统的库函数中，经常使用以下划线“_”打头的函数名或变量名，所以在程序中也应尽量避免使用以“_”打头的标识符，免得与库函数冲突。

3. 常数

C 语言中的常数包括数值常数(如 123、-23.5、1.2E4 等)、字符常数(如'a'、'B'、'c'等)、字符串常数(如"xyz"、"good morning"等)、符号常数以及枚举常数(枚举将在第 7 章将介绍)。

4. 操作符

操作符包括各种运算符(如+、-、*、/)，有特定意义的标点符号(如花括号、方括号、圆括号、逗号)等。

5. 分隔符

分隔符用来分隔相邻的标识符、关键字和常数。最常用的分隔符是空格，此外还可以用制表符、换行符、换页符等作为分隔符。

1.2.2 语句

函数是组成 C 程序的最小结构，而组成函数的基本单位是语句。C 语句是完成某种程序功能(如赋值、输入、输出等)的最小单位，所有的 C 语句都以分号结尾。C 语句可分为表达式语句、复合语句和空语句三类。

1. 表达式语句

任何一个 C 表达式的末尾加上分号后，就构成表达式语句。如：

```
i=0;
x=x+1;
```

等。

2. 复合语句

复合语句是将一组 C 语句用花括号括住，它在语法上被视为一条语句。如：

```
while (i<10)
  {  sum=sum+i;
     i++;
   }
```

复合语句通常用在条件分支或循环语句中。有时为了数据隐藏的目的，用复合语句形成一个代码块，块中定义的局部变量不会对程序的其他部分发生副作用。

3. 空语句

空语句指的是只有一个分号的语句。如：

```
for(i=0; i<1000; i++)
    ;
```

由一条 for 语句(循环语句)和一条空语句组成。空语句用作循环语句的循环体，表示什么也

不做。事实上，这个循环的功能是延迟一小段时间。有时，空语句被用作转向点。

关于语句的详细内容将在第 3 章中进一步介绍。

1.2.3　函数

函数是 C 程序的基本结构，一个 C 程序由一个或多个 C 函数组成。C 函数是完成某个整体功能的最小单位，它是相对独立的模块。一个简单的 C 程序可能只有一个主函数，而复杂的 C 程序则可能包含一个主函数和任意多个其他函数。所有的 C 函数都具有相同的结构。

1. 函数名

函数名是标识和调用函数的依据。在 C 程序中，主函数有固定的名称 main，其他函数则可以根据标识符的命名方法任意取名。主函数通常包括了整个程序的轮廓，由它再调用其他函数。

2. 形式参数

形式参数是函数调用时进行数据传递的主要途径。形式参数简称形参、虚参或哑元，必须放在函数名后面的一对圆括号中。当形式参数超过一个时，相互之间用逗号隔开。有的函数也可以没有形式参数，但圆括号不能省略。

3. 函数体

函数体用来描述函数的功能。函数体要用花括号括住，主要有两大部分，第一部分是本函数内部用到的局部变量类型定义(C 语言中，所有的变量都要先定义后使用)，第二部分是语句序列，完成本函数的功能。

下面的结构描述了 C 程序的一般格式，除了 main( )函数外，函数 fun1( )到 funn( )均为用户自行命名的函数。程序执行时，无论各个函数的书写位置如何，总是先执行 main( )函数，由 main( )函数调用其他函数，最后终止于 main( )函数。

```
全局变量说明
main()
  { 局部变量说明
    语句序列
  }
fun1(形式参数表)
  { 局部变量说明
    语句序列
  }
fun2(形式参数表)
  { 局部变量说明
    语句序列
  }
  ⋮
funn(形式参数表)
```

```
{ 局部变量说明
  语句序列
}
```

关于函数的详细内容，将在第7章中详细介绍。

1.2.4 一个简单的例子

下面是一个简单的C程序，它只有一个main( )函数。

例1.1 编制程序根据给定的矩形的长a和宽b，计算圆的面积S。

由几何学知识可以知道 $S = a \times b$。

先定义三个变量分别表示长、宽和面积，由计算公式计算面积变量S，最后输出长a、宽b和面积S的值。程序如下：

```
main()                          /* 函数名 */
{                               /* 函数体开始 */
 float a,b,S;                   /* 长:a、宽:b、面积:S */
 a=10.2;                        /* 给定长 a 的值 */
 b=5.3;                         /* 给定宽 b 的值 */
 S=a*b;                         /* 计算面积 S */
 printf("a=%f,b=%f,S=%f \n",a,b,S);   /* 输出 a,b,S*/
}                               /* 函数体结束 */
```

提示： “/*”和“*/”之间的部分，是程序的注释部分，这部分不参与程序的编译，只是起到帮助理解程序功能的作用，关于注释的具体内容请参看1.3节相应部分。

第1行是函数名main( )，C语言规定：一个程序由若干个函数组成，其中至少有一个用main( )命名的主函数；第2行至第8行为函数体，是程序的变量定义和执行部分，其中左花括号表示函数体的开始，右花括号表示函数体的结束。每行右边“/*”和“*/”之间的文字称为注释，它对程序的执行没有影响，只对程序进行某些必要的说明，以帮助阅读程序。程序及其运行结果如图1.1和图1.2所示。

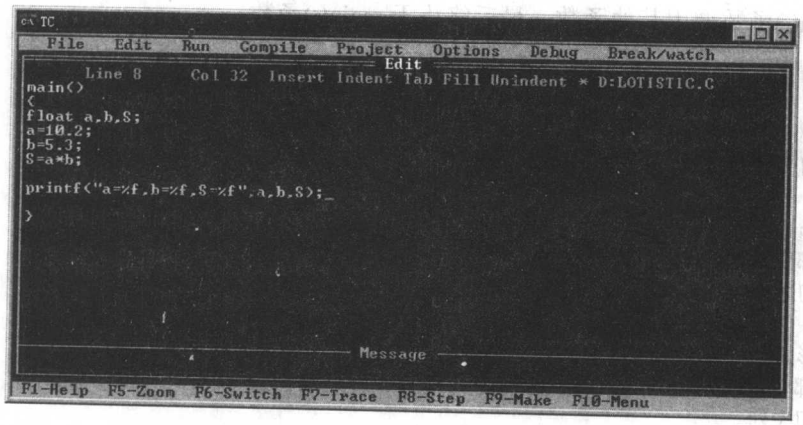

图1.1 例1.1的程序

按下 Ctrl+F9 组合键运行程序，再按下 Alt+F5 组合键查看输出结果，这时屏幕如图 1.2 所示。

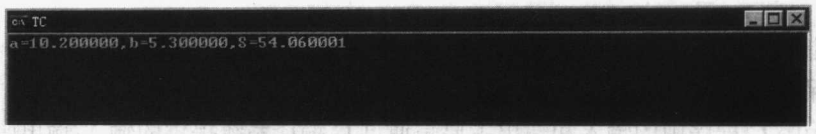

图 1.2　例 1.1 程序的运行结果

按下任意键将返回程序编辑界面(如图 1.1 所示)。

1.3　C 程序的书写格式

C 语言采用自由的程序书写风格。

(1) 每个函数在整个程序文件中的位置是任意的，主函数(main( )函数)不一定出现在程序的开始处，但不管主函数位于程序中的何处，程序运行时总是从主函数开始；

(2) 每个程序行中的语句数量任意，既允许一行内写几条语句，也允许一条语句分几行书写，但每条语句都必须以分号(；)结束。有时还可以在程序的适当地方(如两个函数之间)加进一个或多个空行，使程序结构更加清晰；

(3) 注释的位置任意，注释可以出现在程序的任何地方，既可以独占一行或几行，也可以出现在某语句的开头或结尾处。如果注释占有几行，则每一行都要以“/*”开始，以“*/”结束，“*”和“/”之间不能有空格。注释不是 C 语句，它对程序的编译和运行没有影响，使用注释的惟一目的是增加程序的可读性；

(4) 尽管 C 程序的书写几乎没有限制，但为使程序清晰易读，通常每行写一条语句；不同结构层次的语句从不同的位置开始，即按缩进格式书写成阶梯形状，可以用 Tab 键或空格键调整各行的起始位置。

1.4　C 程序设计简述

一个程序通常具备三个功能，即输入数据、数据运算和输出结果。在 C 语言中，数据运算主要是由赋值语句完成的，数据的输入/输出则需要调用 C 编译系统提供的输入/输出函数。本节简要介绍赋值语句和格式输入/输出函数的使用，以便能进行最简单的程序设计。

1.4.1　赋值语句的简单使用

赋值语句具有计算和存储两大功能，其一般格式为

```
v=e;
```

其中，v 是变量名，e 是表达式。例如，x=a+b，其功能是先计算表达式 a+b 的值，并把该值转换成和 x 相同的数据类型后保存在变量 x 中。

1.4.2 格式输入/输出函数的简单使用

输入指的是通过输入设备将原始数据送入计算机，以便计算机对它们进行处理；输出指的是将保存在内存中的计算结果送到输出设备上，以便人能阅读。由于 C 语言不提供输入/输出语句，C 程序中的输入和输出主要是通过 C 编译系统提供的输入/输出函数实现的。其中用得最多的是格式输出函数 printf( )和格式输入函数 scanf( )。本节对它们做一简单介绍。

1. 格式输出函数 printf( )

格式输出函数 printf( )用来按指定的格式输出数据，是内存与显示器之间进行数据交换的主要手段，也是用户获得程序运行结果的主要途径，其一般调用形式为

```
printf("格式控制字符串",输出项目清单);
```

其中，printf 是函数名，其后的括号中是该函数的参数：格式控制字符串外面加上双引号，用来规定输出格式，例如，%d 用来输出十进制整数，%f 用来输出实数；输出项目清单中包含零个或多个输出项，它们可以是常数、变量或表达式，当有多个输出项时，相互之间用逗号隔开。例如：

```
printf("%f,%f",x,y);
```

用来按小数形式输出变量 x 和 y 的值。

2. 格式输入函数 scanf( )

格式输入函数 scanf( )用来按指定的格式接收输入数据，是内存与键盘之间进行数据交换的主要手段，也是用户为程序运行提供原始数据的主要途径，其一般调用形式为

```
scanf("格式控制字符串",输入项目清单);
```

其中，scanf 是函数名，其后的括号中是该函数的参数：格式控制字符串外面加上双引号，用来规定输入格式，其用法和 printf( )相同；输入项目清单中至少包含一个输入项，且必须是变量的地址(变量地址的表示形式是在变量名前面加一个“&”)，当有多个输入项时，相互之间用逗号隔开。例如

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

用来接收从键盘输入的一个实数和一个整数，并分别存放在变量 a 和 b 中。

1.4.3 库函数和头文件

上面介绍的 printf( )和 scanf( )是 Turbo C 编译系统提供的库函数。库函数不是 C 语言的组成部分，而是由 C 编译系统提供的一些非常有用的功能函数，例如各种输入/输出函数、数学函数、字符串处理函数等，可供用户在自己的程序中直接调用。这些库函数的说明、类型和宏定义都分门别类地保存在相应的头文件(也称标题文件)中，而对应的子程序则存放在运行库(.lib)中，当需要使用系统提供的库函数时，只要在程序开始用

```
#include  <头文件>
```

或

```
#include "头文件"
```

就可以调用其中定义的库函数。printf( )和 scanf( )是在头文件 stdio.h 中定义的，因此，在调用它们之前，应先用

```
#include  <stdio.h>
```

或

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

将它们所在的头文件包含进来。这里，由#开头的 include 行以及前面用到的#define 都称为命令行，它们是 C 语言的编译预处理命令(详见第 9 章)，不是 C 语句，命令动词 include 和 define 也不是 C 关键字。

Turbo C 提供的常用头文件如下(头文件名不区分大小写)：

alloc.h	动态分配函数
conio.h	屏幕处理函数
ctype.h	字符处理函数
graphics.h	图形函数
math.h	数学函数
stdio.h	标准输入/输出及文件操作函数
stdlib.h	标准实用函数
string.h	字符串处理函数

从技术角度讲，任何函数都可以自行编制，完全可以不使用 C 编译系统提供的函数库。然而很少这样做，因为函数库是 C 编译系统提供的一种重要的软件资源，充分利用它们，避免重复自行编制，可达到事半功倍的效果；另一方面，C 语言为了减少对硬件的依赖，不提供任何执行 I/O 操作的方法，这时，自行编制 I/O 函数也存在很大困难。读者应在学习 C 语言本身的同时，逐步了解和掌握各种库函数的功能和用法，以简化编程。

需要说明的是，不同的 C 编译系统提供的库函数在数量、种类、名称及使用上都有一些差异，甚至连标题文件的名称也不一定相同。

1.4.4　简单程序设计举例

程序设计是学习和使用 C 语言的出发点和归宿。本节介绍两个最简单的 C 程序设计例子。

例 1.2　最简单的 C 程序。

```
main()
{
    printf("This is a C program.\n");
}
```

程序的运行结果为：