21世纪高等学校计算机规划教材

高等学校应用型本科系列教材

C语言程序设计

C Programming Language

主　编：杨曙贤

副主编：高宇鹏　李鑫　魏丽娟

主　审：刘发久

- 内容由浅入深、通俗易懂
- 教学深入浅出、循序渐进
- 案例详细丰富、实践性强

人民邮电出版社
POSTS & TELECOM PRESS

21世纪高等学校计算机规划教材
高等学校应用型本科系列教材

C语言程序设计

C Programming Language

主　编：杨曙贤
副主编：高宇鹏　李鑫　魏丽娟
主　审：刘发久

编委会成员：（排名不分先后）

孟建晖	张　文	杨华琼	韩　芳	康晶晶	高　艳
张海绒	杨倩倩	王　龙	李青云	吕艳阳	陈少华
张丽霞	赵　昕	席二辉	章五一	张　举	张　帆
丁　戎	范铁林	张晓磊	徐志强	路　璐	王庆军

高校系列

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 / 杨曙贤主编. -- 北京 : 人民邮电出版社, 2014.9
21世纪高等学校计算机规划教材
ISBN 978-7-115-36082-3

Ⅰ. ①C… Ⅱ. ①杨… 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(2014)第154140号

内 容 提 要

本书是针对三本院校非计算机专业学生学习C语言程序设计而编写的基础教材。全书通过问题和案例的引入，逐步展开对各种概念的介绍，结合C语言的相关知识，由浅入深地讲解各种语句，重点讲解了各种语句的作用和使用方法。全书涵盖C语言的基本内容，包括数据类型与运算规则、顺序程序设计、分支结构程序设计、循环结构程序设计、函数、数组、指针、结构体与共用体以及文件操作等。同时，还强调了程序编写的风格，重点在于指导学生掌握C语言的基本概念和编程方法，养成良好的程序设计习惯。本书对程序算法不做过多的介绍，使得学生在有限的学时内，能尽快掌握C语言的语句和基本的程序编写方法。

本书中的所有程序，都在VC++ 6.0环境下调试运行过，读者可借鉴例子中的程序在自己的计算机上进行实验，通过实验来提高编程能力。

本书还配套编写了《C语言程序设计实验指导》一书，在教学过程中，学生可以通过该书进行上机实验。该指导书的每个题目都有具体的实践目的、内容和操作步骤，通过这些实践，可以进一步理解和掌握C语言编程的各种方法，更灵活自如地编写C语言程序。

◆ 主　　编　杨曙贤
副 主 编　高宇鹏　李　鑫　魏丽娟
主　　审　刘发久
责任编辑　邹文波
执行编辑　吴　婷
责任印制　彭志环　杨林杰

◆ 人民邮电出版社出版发行　　北京市丰台区成寿寺路 11 号
邮编　100164　　电子邮件　315@ptpress.com.cn
网址　http://www.ptpress.com.cn
北京圣夫亚美印刷有限公司印刷

◆ 开本：787×1092　1/16
印张：12.25　　2014 年 9 月第 1 版
字数：318 千字　　2014 年 9 月北京第 1 次印刷

定价：33.00 元

读者服务热线：(010)81055256　印装质量热线：(010)81055316
反盗版热线：(010)81055315

前　言

21世纪是信息技术高度发展并且得到广泛应用的时代，信息技术从多方面改变着人类的生活、工作甚至思维方式。程序设计是信息技术中的一门基本技能，是高等学校重要的计算机基础课。

其中，C语言是当今世界上最广泛、最具影响力的程序设计语言之一。它的语法简洁，数据类型丰富，表达力强，运算符丰富且用法灵活，控制流程和数据结构清晰，程序结构特性突出，可读性强。如果使用 Visual C++ 6.0 集成开发环境，则该工具提供了自动编排C语言程序的缩进格式功能，更有利于培养学生建立良好的编程风格。C语言本身既具有高级语言程序设计的特点，又具有汇编语言的功能；既能有效地进行算法描述，又能对硬件直接进行操作；既适合于开发系统软件，也适合于编写应用程序。C语言本身还具备整体语言紧凑整齐、设计精巧、编辑方便、编译后目标代码运行效率高、操作简便、使用灵活等许多鲜明的特点。

根据培养应用型人才的需要，为了使非计算机专业的高校学生能掌握基本的编程知识和技能，我们编写了这本C语言程序设计教材作为高等学校非计算机专业的公共基础课教材。针对非计算机专业学生的特点，对于C语言中的某些概念和内容相对复杂、比较难懂的部分，编者采取案例驱动的方式进行讲解。从生活中的简单例子，引出相关的概念，再结合C语言的规定，给出一个或几个程序例子，通过对例子的分析和解答，使C语言的基本概念、指令得到详细的阐述，使读者从例子中，逐步理解和掌握所学的知识。再配合适当的上机实验，巩固所学内容。通过32学时的理论课学习和16课时的上机实验，可以使第三类本科高校的非计算机专业学生，基本掌握C语言基础知识和基本的编程技能，为今后在工作中使用C语言或者学习其他高级程序设计语言打下比较牢固的基础。

本教材共分10章。第1章介绍了C语言的概况，讲述了C语言的产生及相关历史背景，阐述了C语言的基本内容和简单的例子，同时，对编写C语言的集成环境 VC++ 6.0 做了基本的介绍，该部分由杨曙贤编写。第2章介绍了C语言的基本数据类型、变量和常量及其表达式的含义。全面介绍了各种运算符的操作功能，包括变量自增、自减运算，关系运算符和关系表达式，混合运算及数据类型转换等内容，该部分由李鑫编写。第3章介绍了顺序程序设计的基本概念，数据的基本输入和输出函数及使用，介绍了返回语句的作用，该部分由魏丽娟编写。第4章介绍了分支结构的程序设计概念，主要介绍了关系运算符和关系表达式、逻辑运算符和逻辑表达式、条件运算符和条件表达式、if 语句及其嵌套使用、多分支的 switch 语句等，该部分由杨曙贤编写。第5章介绍了循环结构程序设计的概念，主要内容是 while、do-while 语句、for 循环语句等，介绍了循环语句的嵌套使用，还介绍了利用 break 语句提前结束循环、利用 continue 语句提前结束本

次循环等语句的功能，这部分内容由杨曙贤编写。第 6 章介绍了函数的概念，主要内容是介绍函数的分类、定义和调用，参数传递和返回值等，还介绍了变量的作用域和存储类型等概念，本章由李鑫编写。第 7 章介绍了数组的概念，主要阐述了一维数组、二维数组、字符类型数组的定义和引用，最后介绍了数组作为函数参数的使用，本章由高宇鹏编写。第 8 章介绍了指针的概念，重点介绍了数组与指针的关系、函数与指针的使用，本章由高宇鹏编写。第 9 章介绍了结构体和共同体的概念，枚举类型和自定义类型，本章由高宇鹏编写。第 10 章介绍了用 C 语言进行文件处理的概念和具体使用方法，本章由魏丽娟编写。

本教材的主编在软件行业工作了几十年，具有丰富的编程经验和娴熟的编程技术，掌握很多种编程语言，同时有着对各种类型学员进行培训的经验，希望这些经验能在本书的编写中有所体现。其他编者也是多年从事大学本科教育的优秀教师，有着丰富的教学经验，与学生的沟通和互动非常融洽，能深刻了解学生们的学习兴趣点和学习习惯。

由于时间仓促，我们在较短的时间内编写了这本教材，书中难免有错误和不足。许多同事和朋友为我们的写作提供了很多方便和支持，在此谨表衷心的谢意。同时，我们想通过今后的教学实践，逐步补充和完善教材的内容，也希望读者能在使用本书的过程中，给我们提出各种改进意见，在此先表谢意。

最后，向对于该书的编撰给予大力支持的姚来昌、林凤彩和唐志宏三位教授致以衷心的感谢。

编　者

2014 年 5 月

目　录

第 1 章
C 语言概述与简单的 C 程序

在现代工作和生活中，我们几乎天天都要跟计算机打交道。很多时候，我们在使用计算机时，其实就是在使用计算机上已经编好的应用软件，即计算机程序，如：我们常用的文字处理程序、图像处理程序、音频视频播放器等，都是在计算机上运行的程序。作为一个一般的软件使用者，我们不需要编写程序，只要使用别人编好的程序就可以了。社会上有许多专业的软件公司，根据市场的需要，在开发着各种各样的应用软件。

有的时候，我们需要让计算机做的事情有着自己的特点和特殊功能。当市面上的软件无法满足特殊的需要时，如果自己能编写一个计算机程序来解决这种特殊的需求，将会对我们的工作和生活带来很大的便利，大大提高了工作效率。在我们的工作中，有时需要编制特定的软件来安装在我们的产品中，这时候，编制软件就成了我们工作的一部分特定的内容了。

用来编制计算机程序的编程语言，就是一整套有着严格规定的编写计算机指令的方法和规则以及计算机能识别的指令的集合。计算机编程语言有很多种，其中 C 语言是流行最广、使用最多，也是功能最强大的编程语言。它从诞生到现在，经历了几十年的时间，与它同时代的很多编程语言由于各种原因，很多都逐渐消失了。只有 C 语言以其强大的功能和优异的性能，一直活跃在各种软件开发的环境中。因此，现在几乎所有的高等院校，都把 C 语言作为各专业学习计算机编程的公共基础课。

本章主要介绍 C 语言的基本概念及它的起源和发展过程。还介绍了 C 语言的基本内容，通过一个简单的 C 程序，初步展示 C 语言的编写方式和特点。最后介绍了一种编写 C 语言程序的集成开发环境 VC++ 6.0 的基本使用。

【学习目标】

（1）了解 C 语言的基本概念。

（2）了解 C 语言的符号集以及编程风格。

（3）熟悉 C 语言的上机环境与调试过程。

1.1 C 语言简介

早期的系统软件主要是用汇编语言编写的，如操作系统。由于汇编语言依赖于计算机硬件，

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，最好改用高级语言，但一般高级语言难以实现汇编语言的某些功能，而汇编语言可以直接对硬件进行操作，例如，对内存地址的操作、位（bit）操作等。人们设想能否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，集它们的优点于一身。于是，C 语言就在这种情况下应运而生了，之后成为国际上广泛流行的计算机高级语言。它适合于作为系统描述语言，即用来写系统软件，也可用来写应用软件。

C 语言是在 B 语言的基础上发展起来的，它的根源可以追溯到 ALGOL 60。1960 年出现的 ALGOL 60 是一种面向问题的高级语言，它离硬件比较远，不宜用来编写系统程序，1963 年英国的剑桥大学推出了 CPL（Combined Programming Language）语言。CPL 语言在 ALGOL 60 的基础上接近硬件一些，但规模比较大，难以实现。1967 年英国剑桥大学的 Matin Richards 对 CPL 语言做了简化，推出了 BCPL（Basic Combined Programming Language）语言。1970 年美国贝尔实验室的 Ken Thompson 以 BCPL 语言为基础，又做了进一步简化，它使得 BCPL 能挤压在 8KB 内存中运行，这个很简单的而且很接近硬件的语言就是 B 语言（取 BCPL 的第一个字母），并用它写了第一个 UNIX 操作系统，在 DEC PDP-7 上实现。1971 年在 PDP-11/20 上实现了 B 语言，并写了 UNIX 操作系统。但 B 语言过于简单，功能有限，并且和 BCPL 都是“无类型”的语言。1972 年至 1973 年间，贝尔实验室的 D.M.Ritchie 在 B 语言的基础上设计出了 C 语言（取 BCPL 的第二个字母）。C 语言既保持了 BCPL 和 B 语言的优点（精练，接近硬件），又克服了它们的缺点（过于简单，数据无类型等）。最初的 C 语言只是为描述和实现 UNIX 操作系统提供一种工具语言而设计的。1973 年，K. Thompson 和 D.M. Ritchie 两人合作把 UNIX 的 90%以上用 C 改写，即 UNIX 第 5 版。原来的 UNIX 操作系统是 1969 年由美国的贝尔实验室的 K. Thompson 和 D. M. Ritchie 开发成功的，是用汇编语言写的，这样，UNIX 使分散的计算系统之间的大规模联网以及互联网成为可能。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 版公布后，C 语言的突出优点才引起人们普遍注意。1977 年出现了不依赖于具体机器的 C 语言编译文本《可移植 C 语言编译程序》，使 C 移植到其他机器时所需做的工作大大简化了，这也推动了 UNIX 操作系统迅速地在各种机器上实现。例如，VAX、AT&T 等计算机系统都相继开发了 UNIX。随着 UNIX 的日益广泛使用，C 语言也迅速得到推广。C 语言和 UNIX 可以说是一对孪生兄弟，在发展过程中相辅相成。1978 年以后，C 语言已先后移植到大、中、小、微型机上，如 IBM System/370、Honeywell 6000 和 Interdata 8/32，已独立于 UNIX 和 PDP 了。现在 C 语言已风靡全世界，成为世界上应用最广泛的几种计算机语言之一。以 1978 年由美国电话电报公司（AT&T）贝尔实验室正式发表的 UNIX 第 7 版中的 C 编译程序为基础，Brian W. Kernighan（柯尼汉）和 Dennis M. Ritchie（里奇）合著了影响深远的名著《The C Programming Language》，常常称它为“K&R”，也有人称为“K&R 标准”或“白皮书”（white book），它成为后来广泛使用的 C 语言版本的基础，但在“K&R”中并没有定义一个完整的标准 C 语言。为此，1983 年，美国国家标准化协会（ANSI）X3J11 委员会根据 C 语言问世以来各种版本对 C 的发展和扩充，制定了新的标准，称为 ANSI C，ANSI C 比原来的标准 C 有了很大的发展：K&R 在 1988 年修改了他们的经典著作《The C Programming Language》，按照 ANSI C 标准重新写了该书。1987 年，

ANSI 又公布了新标准——87 ANSI C。目前流行的 C 编译系统都是以它为基础的。当时广泛流行的各种版本 C 语言编译系统虽然基本部分是相同的，但也有一些不同。在微型机上使用的有 Microsoft C（MS C）、Borland Turbo C、Quick C 和 AT&T C 等，它们的不同版本又略有差异。到后来的 Java、C++、C#都是以 C 语言为基础发展起来的。

1.2　C 语言符号集与词汇

C 语言符号集就是用 C 语言编写程序时要用到的所有符号的集合，包括英文字母、数字和一些有特定含义的标点符号。由 C 语言符号集里的这些符号构成 C 语言程序中的语句，再由这些语句组成程序。

C 语言符号集包括如下的内容：

（1）字母、数字和下划线

大写英文字母：A～Z

小写英文字母：a～z

数字符：0～9

下划线：_

（2）空白符

这里的空白符不是指键盘上的一个空白键所表示的空白字符，而是由若干个看不见的符号组成的，如空格符、制表符（table）、垂直制表符、回车符、换行符、换页符。它们在 C 语言源程序中看上去是“空白”。空白符在字符和字符串中起作用。在其他地方使用时，只起隔离作用，使不同的指令（运算符）或语句之间有一个间隔。编译程序在编译 C 语言源程序时，会理解它们的含义，并自动忽略它们。显然，对于 C 语言中的关键字（即 C 语言中由英文字符组成的具有完整意义的指令或运算符号）内，不能用空白符将这些指令中的字符隔离开，如关键字 switch，不能在这 6 个字母中插入任何空白符。

在程序中适当地使用空白符也可以使程序的段落结构更加清晰，增加可读性，易于日后对程序的维护。

（3）标点符号和特殊符号

C 语言使用的标点符号和特殊符号，如表 1-1 所列。

表 1-1　　C 语言使用的标点符号和特殊符号

符号	名称	符号	名称
,	逗号	>	右尖括号
.	圆点	!	感叹号
;	分号	|	竖线

续表

符号	名称	符号	名称
:	冒号	/	斜杠
?	问号	\	反斜杠
‘	单引号	~	波折号
“	双引号	#	井号
(	左圆括号	%	百分号
)	右圆括号	&	and（与）
[	左方括号	^	xor（异或）
]	右方括号	*	乘号
{	左花括号	-	减号
}	右花括号	=	等于号
<	左尖括号	+	加号

（4）转义字符

转义字符是 C 语言中的一种特殊形式。通常使用转义字符表示 ASCII 码字符集中不可打印的控制字符和特定功能的字符。转义字符用反斜杠\后面跟一个字符或一个八进制或十六进制数表示。表 1-2 列出了 C 语言中常用的转义字符。

表 1-2　　C 语言中常用的转义字符

转义字符	意义	ASCII 码值（十进制）
\a	响铃（BEL）	007
\b	退格（BS）	008
\f	换页（FF）	012
\n	换行（LF）	010
\r	回车（CR）	013
\t	水平制表（HT）	009
\v	垂直制表（VT）	011
\\	反斜杠	092
\?	问号字符	063
\’	单引号字符	039
\”	双引号字符	034
\0	空字符（NULL）	000
\ddd	任意字符	3 位八进制
\xhh	任意字符	2 位十六进制

转义字符的作用：表示控制代码、表示字符和字符串常量、表示 ASCII 码字符集中任意字符。

在字符串中如果使用单引号、反斜杠、双引号、反斜杠时，都必须使用转义字符表示，即在这些字符前加上反斜杠。否则，编译程序会“误解”这些字符的本意，将它们“理解”成 C 语言中的关键字，并进行编译，将产生语法或逻辑错误。

转义字符中的字母只能用小写字母，每个转义字符本身虽然是由两个字母组成的，但是只能看作是一个字符。\v（垂直制表符）和\f（换页符）这两个转义字符对屏幕没有任何影响，但会影响打印机执行相应的操作。

（5）标识符

在 C 语言中，标识符是一种记号，是对变量、函数、标号和其他各种用户定义的对象的一种命名。这种命名的内容不是 C 语言规定的，但是命名的规则，却在 C 语言中有一定的限定。

① 标识符必须使用 C 符号集里的英文字母、数字或下划线。

② ANSI（美国国家标准）规定，标识符的长度为 1～32 个字母，但是在普通的 C 编译环境中，标识符的所有字符中，只有前面的 8 个字符被唯一地作为标识内容，超过 8 个字符后的字符都被忽略了。例如，下面的标识符都被当作同一个标识符来处理了，counters、counters1、counters2。

③ 命名标识符时，第一个字符必须是字母或下划线，随后的字符可以是字母、数字或下划线，不能有特殊字符。同时，标识符中的字母是分大小写的，标识符拼写的字母相同，但是字母的大小写不同，就不是同一个标识符。如 Good 和 good，在 C 语言的标识符中，是两个不同的标识符。

④ 标识符既不能与 C 语言的关键字（见下面的阐述）相同，也不能与用户已编写的函数或 C 语言库函数名字相同。

（6）关键字

关键字是由 C 语言规定的，在程序中具有特定意义的字符串，通常也称为保留字。用于定义的标识符不能与关键字相同。因为计算机（编译程序）一遇到这些关键字，就把它们赋予了规定的含义。

C 语言的关键字分为如下几类。

① 类型说明符，用于定义和说明变量、函数或其他数据结构类型。

② 语句定义符，用于表示一个语句的功能，如 if else 就是条件语句的语句定义符。

③ 预处理命令字，用于表示一个预处理命令，如 include。

C 语言有如下 32 个关键字。

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

有如下运算符号。

-（负号或减），+（加），*（乘），/（除），%（取余），--（自减 1），++（自增 1）。

有如下关系比较符号。

>（大于），<（小于），>=（大于等于），<=（小于等于），==（等于），!=（不等于）。

有如下逻辑操作（运算）符号。

&&（与），||（或），!（非或反）。

有如下元操作符号。

<<（向左移一位元），>>（向右移一位元），&（与），|（或），～（非），^（互斥）。

其他符号。

/* */注释符号，

= 指定符号，

： 叙述分隔，

* 指标符号，

（）算数表达或参数列表的开始与结束符号，

[]下标开始与结束符号，

，用来分隔一列中的各项，

\ 控制格式的起始符号或转义符号，

预处理符号，

'' 字符的开始和结束符号，

"" 字符串的开始与结束符号，

？：条件运算符，

& 位运算符，

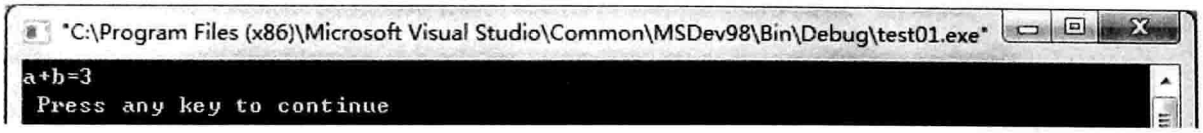
% 输入或输出控制格式的前导符号。

1.3 简单的 C 程序

一个简单的 C 程序例子如下。

```
#include <stdio.h>
main()
{
    int a,b,c;
    a = 1;
    b = 2;
    c = a+b;
    printf("a+b=%d\n ",c);
}
```

程序的运行结果如下。



这是一个非常简单的 C 语言源程序，但是已经包含了 C 语言程序基本的内容，如主函数名 main（），这是每个 C 语言程序必须有的一个函数，而且只能有一个主函数，每个 C 语言程序都是从 main 函数开始运行的。

函数名后面的内容，是需要执行的程序段，用一对{}号括起来了。int 是一个声明程序中要使用的存储单元的指令，存储单元是计算机内部存储器中的一个单元，为了使用该单元，C 语言对这些用到的单元，赋予一个名字，这些单元在程序中，就用这个名字来标记和使用。不同的数据需要用不同的单元格式来存储，因此就产生了对存储单元声明不同类型的要求。

程序中的 a、b、c 都是存储单元，不同的是 a 和 b 被直接赋给了一个整数值，而 c 是被赋给了 a 和 b 相加后的结果。

Pirntf()是一个系统里已经存在的库函数，在程序里写出来，就是被调用了。这个函数可以给它加上一些参数，这些参数要放在()里，根据参数的不同，可以显示不同的内容。为了使用库函数或者在其他地方已经写好的函数，必须在调用它之前，告诉编译程序，从哪里才能找到库函数或外部函数，这就是#include 指令的含义，它后面跟着的内容，就是要把这个库的内容要包含到此程序中。

1.4　C 语言的编程风格

C 语言是一种无格式语言，也就是说，只要按规定，顺序地编写 C 语言的指令即可，没有书写格式的要求。

但是，在实际开发工作中，为了使程序的结构比较清晰，可读性强，便于检查和维护，一般有如下的编写风格。

（1）对齐和缩进。

{和}独占一行，且位于同一列，与引用他们的语句左对齐，便于检查配对情况；

位于同一层{}之内的代码在{右边若干个位置处左对齐；可以设置 2 个空格键或 Tab 键缩进。

例如：

不良的风格

```
#include <stdio.h>
#include <math.h>
main()
{int i;
for (i=2;i<100;i++)
{if(isprime(i))
```

```
    printf( "%d\t",i)}
    int isprime(int n)
    {int k,i;
    if(n==1) return 0;
    k=sqrt((double)n);
    for(i=2;i<=k;i++)
    {if(n%i==0) return 0;}
    return 1;
    }
```

良好的风格

```
#include<stdio.h>
#include<math.h>
main()
{
    int i;
    for (i=2;i<100;i++)
    {
          if (isprime(i))
               printf("%d\t",i);
    }
}
int isprime(int n)
{
    int k,i;
    if (n == 1) return 0;
    k = (int) sqrt((double) n);
    for (i=2; i<=k; i++)
    {
        if (n%i ==0)
             return 0;
 }
    return 1;
    }
```

表 1-3 列出了两种风格的编程格式，我们希望采用建议的风格。

表 1-3　　两种程序编写风格的对比

建议的风格	不建议的风格

void Function(int x)
{
…// program code
}
	void Function(int x) {
	… // program code
	}

续表

建议的风格	不建议的风格

if (condition)
{
…// program code
}
else
{
…// program code
}
	if (condition){
	…// program code
	}
	else {
	…// program code
	}

for (initialization; condition; update)
{
…//program code
}
	for (initialization; condition; update) {
	…//program code
	}

while (condition)
{
…// program code
}
	while (condition){
	… // program code
	}

如果出现嵌套的{ }，则使用缩进对齐，如：
{
…
{
…
}
…
}	

（2）每个函数都必须有注释，即使函数可能只有几行。

如：

```
/*这个程序是为了计算圆的面积 */
void function1()
{
…
}
```

（3）每个函数定义结束之后以及每个文件结束之后都要加上一个或若干个空行。

如：

```
/*这是函数 function1 */
void function1 ( )
{
…
}

/* 这是函数 function2*/
void function1 ( )
{
```