



普通高等教育“十二五”重点规划教材

中国科学院教材建设专家委员会“十二五”规划教材

# C 语言程序设计 教程

岳莉◎主编



科学出版社

普通高等教育“十二五”重点规划教材  
中国科学院教材建设专家委员会“十二五”规划教材

# C 语言程序设计教程

岳莉 主编

边晶 李柯景 副主编

科学出版社

北京

## 内 容 简 介

本书语言简洁,通俗易懂,内容由浅入深,以C语言程序案例为导向,介绍了C语言程序设计的基本方法。

本书共11章,从内容上可分为三部分。第一部分为入门基础知识,主要内容包括C语言概述、数据类型与运算符。第二部分为程序设计的基础部分,主要内容包括程序语句的基本控制结构。掌握了第一部分、第二部分的内容,读者可以完成简单的程序设计。第三部分为模块化程序设计的概念和实现的方法,主要内容包括函数、数组、指针、结构体、共用体、枚举、文件操作和预处理命令等。

本书可作为高等院校各专业计算机程序设计课程的教材。

### 图书在版编目(CIP)数据

C语言程序设计教程/岳莉主编. —北京:科学出版社,2015  
(普通高等教育“十二五”重点规划教材·中国科学院教材建设专家委员会“十二五”规划教材)

ISBN 978-7-03-043180-6

I. ①C… II. ①岳… 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(2015)第018907号

责任编辑:戴薇 余梦洁 / 责任校对:刘玉靖

责任印制:吕春珉 / 封面设计:多边数字

科学出版社出版

北京东黄城根北街16号

邮政编码:100717

<http://www.sciencep.com>

北京路局票据印刷厂印刷

科学出版社发行 各地新华书店经销

\*

2015年2月第一版 开本:787×1092 1/16

2015年2月第一次印刷 印张:18 1/2

字数:439 000

定价:39.00元

(如有印装质量问题,我社负责调换<路局票据>)

销售部电话 010-62140850 编辑部电话 010-62135763-2038 (VT03)

版权所有,侵权必究

# 前 言



C 语言是一种简洁高效的计算机高级程序设计语言，是根据结构化程序设计原则设计并实现的。C 语言具有丰富的数据类型，为结构化程序设计提供了多种数据结构和控制结构，能够实现汇编语言中的大部分功能，同时，用 C 语言编写的程序具有良好的可移植性。C 语言虽然是为编写 UNIX 操作系统而设计的，但并不依赖于 UNIX 操作系统，目前 C 语言能在多种操作系统环境下运行，并且已经在很多领域里得到了应用，是目前国际上应用广泛的高级程序设计语言之一。

本书的主要特点如下：

- 1) 按照循序渐进的原则，逐步引出 C 语言中的基本概念，如 C 语言中的运算符比较丰富，其优先级也比较复杂，在本书中，根据运算符的种类，把运算符融入实例中进行讲解，这样有助于读者的理解和掌握。
- 2) 在介绍 C 语言中的基本概念时，除了阐述理论之外，还通过典型的例题着重强调了基本概念在程序设计中的应用，有助于读者的理解和掌握。
- 3) 本书的重点放在 C 语言的使用上，对例题中出现的每一个算法都做出了比较详细的解释，因此适合初学者和自学者使用。
- 4) 每章末尾均配有习题和实验，实验包括程序填空题、程序改错题和程序设计题等，有助于读者巩固已学习的内容。
- 5) 本书在文字叙述上力求条理清晰、简洁，便于读者阅读。
- 6) 本书中的所有例题都已上机调试并通过，读者可以边学习边上机操作。

本书由多年从事 C 语言教学的具有丰富教学经验的一线教师编写而成。本书由岳莉担任主编，边晶、李柯景担任副主编，其中第 1 章、第 2 章、第 9 章和第 10 章由岳莉编写，第 3 章由李克玲编写，第 4 章由郭南楠编写，第 5 章由李柯景编写，第 6 章和第 11 章由边晶编写，第 7 章由徐志伟编写，第 8 章由庄天舒编写。全书由陈玉明教授负责统稿。陈玉明教授对本书提出许多宝贵意见，在此表示衷心的感谢。

由于编者水平有限，编写时间紧，书中难免有疏漏和不妥之处，敬请广大读者批评指正。

编 者

2014 年 12 月

# 目 录

第 1 章 C 语言概述	1
1.1 程序设计语言概述	1
1.2 C 语言的发展及特点	2
1.2.1 C 语言的发展	2
1.2.2 C 语言的特点	3
1.3 第一个 C 语言程序	4
1.3.1 C 语言程序举例	4
1.3.2 C 语言程序的结构特点及书写规则	7
1.4 运行 C 语言程序的步骤与方法	8
习题	13
实验	14
第 2 章 数据类型与运算符	15
2.1 字符集与词汇	15
2.2 数据类型	16
2.3 常量与变量	17
2.3.1 常量	18
2.3.2 变量	21
2.3.3 数据类型转换	26
2.4 运算符和表达式	28
2.4.1 算术运算符和表达式	29
2.4.2 赋值运算符和表达式	30
2.4.3 逗号运算符和表达式	31
2.4.4 关系运算符和表达式	32
2.4.5 逻辑运算符和表达式	33
2.4.6 位运算符	34
习题	37
实验	39
第 3 章 顺序结构程序设计	45
3.1 引例	45

3.2 C 语言的基本语句	46
3.2.1 表达式语句	46
3.2.2 函数调用语句	46
3.2.3 控制语句	46
3.2.4 复合语句	47
3.2.5 空语句	47
3.3 赋值语句	47
3.4 基本输入/输出函数	49
3.4.1 引例	49
3.4.2 格式输出函数 printf()	50
3.4.3 格式输入函数 scanf()	57
3.4.4 字符数据的输入/输出函数	63
3.5 顺序结构程序设计举例	64
习题	66
实验	70
<b>第 4 章 选择结构程序设计</b>	<b>73</b>
4.1 引例	73
4.2 if 语句	74
4.2.1 双分支 if 语句	74
4.2.2 单分支 if 语句	75
4.2.3 if 语句的嵌套结构	77
4.3 switch 语句	79
习题	81
实验	86
<b>第 5 章 循环结构程序设计</b>	<b>89</b>
5.1 引例	89
5.2 循环语句	90
5.2.1 while 语句	90
5.2.2 do...while 语句	93
5.2.3 for 语句	94
5.2.4 goto 语句和 if 语句构成的循环结构	99
5.3 循环的嵌套	100
5.4 break 语句和 continue 语句	103
5.4.1 break 语句	104
5.4.2 continue 语句	105

5.5 循环结构程序设计举例	106
习题	110
实验	114
<b>第6章 函数</b>	<b>119</b>
6.1 引例	119
6.2 函数的定义	121
6.3 函数的调用与参数传递	122
6.3.1 函数调用的一般形式	122
6.3.2 函数参数的传递	122
6.3.3 函数调用的方式	125
6.4 函数的返回与返回值	125
6.5 函数的声明	126
6.6 函数的嵌套调用和递归调用	127
6.7 变量的作用域	129
6.7.1 局部变量	130
6.7.2 全局变量	132
6.8 变量的存储方式与类别	133
6.8.1 变量的存储方式	133
6.8.2 自动变量	134
6.8.3 寄存器变量	135
6.8.4 静态变量	135
6.8.5 全局变量	138
6.9 内部函数和外部函数	139
6.9.1 内部函数	139
6.9.2 外部函数	140
习题	140
实验	143
<b>第7章 数组</b>	<b>149</b>
7.1 一维数组	149
7.1.1 引例	149
7.1.2 一维数组的定义和引用	151
7.1.3 一维数组的初始化	152
7.1.4 一维数组的输入/输出	153
7.1.5 一维数组程序设计举例	154
7.2 二维数组	161

7.2.1	引例	161
7.2.2	二维数组的定义和引用	162
7.2.3	二维数组的初始化	163
7.2.4	二维数组的输入/输出	164
7.2.5	二维数组程序设计举例	165
7.3	字符数组	166
7.3.1	引例	166
7.3.2	一维字符数组	167
7.3.3	字符串	168
7.3.4	字符串处理函数	170
7.3.5	字符串程序设计举例	173
7.4	数组与函数	174
	习题	176
	实验	181
<b>第8章</b>	<b>指针</b>	<b>188</b>
8.1	指针的概念	188
8.2	指针变量	188
8.2.1	引例	188
8.2.2	指针变量的定义	189
8.2.3	指针变量的引用	190
8.2.4	指针变量作为函数参数	191
8.3	通过指针引用数组	192
8.3.1	数组元素的指针	192
8.3.2	引用数组元素时指针的运算	193
8.3.3	通过指针引用数组元素	194
8.3.4	数组名作为函数参数	195
8.3.5	通过指针引用多维数组	197
8.4	通过指针引用字符串	199
8.4.1	字符串的引用方式	199
8.4.2	字符指针作为函数参数	201
8.4.3	字符指针变量和字符数组的比较	202
8.5	指向函数的指针	203
8.5.1	函数指针的定义	203
8.5.2	用函数指针变量调用函数	203
8.5.3	函数指针变量的定义和使用	204
8.5.4	用指向函数的指针作为函数参数	204



8.6	返回指针值的函数	206
8.7	指针数组和多重指针	207
8.7.1	指针数组的定义	207
8.7.2	多重指针	208
8.7.3	指针数组作为主函数形参	209
8.8	内存动态分配和空指针类型	210
8.8.1	内存动态分配的定义	210
8.8.2	建立内存动态分配的方法	210
8.8.3	空指针类型	211
	习题	212
	实验	213
<b>第 9 章</b>	<b>结构体、共用体和枚举</b>	<b>218</b>
9.1	结构体	218
9.1.1	结构体的定义	218
9.1.2	结构体变量的定义	219
9.1.3	结构体变量的赋值与引用	220
9.1.4	结构体数组	222
9.1.5	指向结构体变量的指针	225
9.1.6	指向结构体数组的指针	227
9.1.7	结构体指针变量作为函数参数	228
9.2	共用体	230
9.2.1	共用体的声明	230
9.2.2	共用体变量的定义	230
9.2.3	结构体和共用体的区别	231
9.3	枚举	232
9.3.1	枚举的定义与枚举变量的声明	232
9.3.2	枚举变量的赋值和使用	232
9.4	类型定义符	234
	习题	235
	实验	237
<b>第 10 章</b>	<b>文件操作</b>	<b>240</b>
10.1	文件概述	240
10.1.1	文件的分类	240
10.1.2	文件指针	240
10.2	文件的操作	241

10.2.1	文件的打开与关闭	241
10.2.2	文件的读/写	243
10.2.3	文件的随机读/写	251
10.2.4	文件检测函数	253
10.3	库文件	254
	习题	255
	实验	257
<b>第 11 章</b>	<b>预处理命令</b>	<b>260</b>
11.1	宏定义	260
11.1.1	无参宏定义	260
11.1.2	有参宏定义	263
11.2	文件包含	269
11.3	条件编译	270
	习题	272
	实验	273
附录 A	C 语言运算符的优先级与结合性	275
附录 B	C 语言的关键字及解释	276
附录 C	ASCII 码表	277
附录 D	C 语言标准库函数	278
	参考文献	284

# 第 1 章 C 语言概述

## 1.1 程序设计语言概述

程序设计语言是用于编写计算机程序的语言。语言的基础是一组记号和一组规则，记号根据规则组成的记号串就是语言。在程序设计语言中，这些记号串就是程序。

自 20 世纪 60 年代以来，世界上公布的程序设计语言达上千种之多，但是只有很小的一部分得到了广泛的应用。从发展历程来看，程序设计语言可以分为 4 代。

### 1. 第 1 代——机器语言

机器语言由二进制代码指令组成，不同的中央处理器 (Central Processing Unit, CPU) 具有不同的指令系统。机器语言程序编写难、修改难、维护难，需要用户直接对存储空间进行分配，编程效率极低。因此，机器语言已经逐渐被淘汰。

### 2. 第 2 代——汇编语言

汇编语言指令是机器指令的符号化，与机器指令有直接的对应关系，所以汇编语言同样存在难学、难用、容易出错、维护困难等缺点。但是汇编语言也有自己的优点，即可直接访问系统接口，汇编程序翻译成机器语言程序的效率高。从软件工程角度来看，只有在高级语言不能满足设计要求，或者不具备支持某种特定功能的技术性能（如特殊的输入/输出）时，才使用汇编语言。

### 3. 第 3 代——高级语言

高级语言是面向用户的、基本独立于计算机种类和结构的语言。高级语言最大的优点是形式上接近算术语言和自然语言，接近人们通常使用的概念。高级语言的一个命令可以代替几条、几十条甚至几百条汇编语言的指令。因此，高级语言易学、易用，通用性强，应用广泛。高级语言种类繁多，可以从其应用角度和对客观系统的描述两个方面对其进行进一步分类。

从应用角度来看，高级语言可以分为基础语言、结构化语言和专用语言。

#### (1) 基础语言

基础语言也称为通用语言。基础语言历史悠久，应用广泛，拥有大量的已开发的软件库和众多的用户，被人们所熟悉和接受。FORTRAN、COBOL、BASIC、ALGOL 等都属于基础语言。

### (2) 结构化语言

20 世纪 70 年代以来,结构化程序设计和软件工程的思想日益为人们所接受和认可。在这种背景下,一些有影响的结构化语言先后出现。这些结构化语言直接支持结构化的控制结构,具有很强的数据处理能力。Pascal、C、Ada 是结构化语言的突出代表。

### (3) 专用语言

专用语言是为某种特殊应用而专门设计的语言,通常具有特殊的语法形式。一般来说,专用语言的应用范围窄,可移植性和可维护性不如结构化语言好。随着时代的发展,专业语言已有数百种,应用比较广泛的有 APL、Forth、LISP 等。

从对客观系统的描述来看,高级语言可以分为面向过程语言和面向对象语言。

#### (1) 面向过程语言

以“数据结构+算法”程序设计范式构成的程序设计语言,称为面向过程语言。前面介绍的程序设计语言大多为面向过程语言。

#### (2) 面向对象语言

以“对象+消息”程序设计范式构成的程序设计语言,称为面向对象语言。比较流行的面向对象语言有 Delphi、Visual Basic、Java、C++ 等。

## 4. 第 4 代——非过程化语言

非过程化语言在编程时只需说明“做什么”,不需要描述算法细节。

数据库查询和应用程序生成器是非过程化语言的两个典型应用。用户可以用结构化查询语言(Structured Query Language, SQL)对数据库中的信息进行复杂的操作。用户只需将要查找的内容在什么地方、根据什么条件进行查找等信息告诉 SQL,SQL 将自动完成查找过程。应用程序生成器是根据用户的需求自动生成满足需求的高级语言程序。但是,真正的第 4 代程序设计语言还没有出现,所谓的第 4 代语言大多是指基于某种语言环境的具有非过程化语言特征的软件工具产品,如 System Z、PowerBuilder、FOCUS 等。第 4 代程序设计语言是面向应用,为最终用户设计的一类程序设计语言,具有缩短应用开发过程、降低维护代价、最大限度地减少调试过程中出现的问题及对用户友好等优点。

## 1.2 C 语言的发展及特点

### 1.2.1 C 语言的发展

C 语言的原型是 ALGOL 60 语言,也称为 A 语言。1963 年,剑桥大学在 ALGOL 60 语言的基础上推出了 CPL(Combined Programming Language)。1967 年,剑桥大学的 Martin Richards 对 CPL 语言进行了简化,推出了 BCPL(Basic Combined Programming Language)。

1970年,美国贝尔实验室的Ken Thompson对BCPL进行了修改,设计出了B语言,并用B语言编写了第一个UNIX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C语言。为了使UNIX操作系统得到推广,1977年,D. 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,B. W. Kernighian和D. M. Ritchie出版了名著*The C Programming Language*,从而使C语言成为目前世界上应用最广泛的高级程序设计语言。

1988年,随着微型计算机的日益普及,出现了许多C语言版本。由于没有统一的标准,不同版本的C语言代码无法在多种编译器上实现。为了改变这种情况,美国国家标准化协会(American National Standard Institute, ANSI)为C语言制定了一套ANSI标准,此标准已成为现行的C语言标准。

### 1.2.2 C语言的特点

C语言既有高级语言的特点,又有汇编语言的特点。C语言既可以作为工作系统设计语言,编写系统应用程序,也可以作为应用程序设计语言,编写不依赖计算机硬件的应用程序。具体来讲,C语言的特点如下。

1) 简洁紧凑、灵活方便。C语言一共只有32个关键字和9种控制语句,程序书写形式自由,区分大小写。C语言把高级语言的基本结构和语句与低级语言的实用性结合起来,因此,C语言可以像汇编语言一样对位、字节和地址进行操作,而这三者是计算机的基本工作单元。

2) 运算符丰富。C语言的运算符很多,达34种。C语言把括号、赋值、强制类型转换等作为运算符处理。因此,C语言的运算类型极其丰富,表达式类型多样化。灵活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。

3) 数据类型丰富。C语言的数据类型有整型、实型、字符型、数组型、指针型等,可以实现各种复杂的运算。C语言引入了指针概念,使程序效率更高。另外,C语言具有强大的图形绘制功能,并而且计算功能、逻辑判断功能强大。

4) 结构式语言。结构式语言的显著特点是代码及数据的分隔化,即程序的各个部分除了必要的信息交互外彼此独立。结构式语言可使程序层次清晰,便于使用、维护及调试。C语言是以函数形式提供给用户的,这些函数方便调用,并具有多种循环、条件语句控制程序流向,从而使程序完全结构化。

5) 语法灵活,程序设计较自由。虽然C语言也是强类型语言,但语法比较灵活,用户拥有较大的自由。

6) 允许直接访问物理地址,对硬件进行操作。由于C语言允许直接访问物理地址,可以直接对硬件进行操作,因此其既具有高级语言的功能,又具有低级语言的许多功能。

7) 生成目标代码质量高,程序执行效率高。C语言一般只比汇编程序生成的目标

代码效率低 10%~20%。

8) 适用范围大, 可移植性好。C 语言的突出优点是适用于多种操作系统, 如 DOS、UNIX、Windows 98、Windows NT 等, 也适用于多种机型。C 语言的可移植性好, 具备很强的数据处理能力, 因此适用于编写系统软件, 及三维、二维图形和动画程序。

## 1.3 第一个 C 语言程序

### 1.3.1 C 语言程序举例

下面举例说明 C 语言源程序结构的特点。以下几个程序由简到难, 表现了 C 语言源程序在组成结构上的特点。从这些例题中可以初步了解 C 语言源程序的基本结构和书写格式。

**【例 1-1】** 输出“你好!”。

```
#include<stdio.h>
void main()
{
    printf("你好 !");
}
```

程序运行结果:

你好!

说明:

- 1) include 称为文件包含命令。
- 2) main 是主函数的函数名, 表示这是一个主函数。
- 3) 每一个 C 语言源程序都必须有主函数, 并且只能有一个主函数, 即 main() 函数。
- 4) printf() 函数是一个由系统定义的标准函数, 可在程序中直接调用。其功能是把要输出的内容送到显示器显示。

**【例 1-2】** 从键盘上输入一个数 x, 求 x 的正弦值, 然后输出结果。

```
#include<math.h>
#include<stdio.h>
void main()
{
    double x,s;
    printf("input number:\n");
    scanf("%lf",&x); /*输入 x 的值*/
    s=sin(x);
    printf("sin of %lf is %lf\n",x,s);
}
```

程序运行结果:

```
input number:
5↵ (“↵”代表按Enter键,全书统一用“↵”)
sin of 5.000000 is -0.958924
```

说明:

- 1) include 称为文件包含命令。
- 2) 扩展名为.h的文件称为头文件。
- 3) 第3行为main()函数头。
- 4) 第4行main()函数开始。
- 5) 第5行定义两个实型变量,以便被后面程序使用。
- 6) 第6行输出提示信息。
- 7) 第7行从键盘获得一个实数x,“/\*”和“\*/”之间的内容为注释,不参与程序运行。
- 8) 第8行求x的正弦值,并把x赋值给变量s。
- 9) 第9行输出程序运算结果。
- 10) 第10行main()函数结束。

在main()之前的两行语句称为预处理命令。预处理命令还有其他几种,在第11章中将进行详细介绍,这里的include称为文件包含命令,其意义是把尖括号或引号内指定的文件包含到本程序中,成为本程序的一部分。被包含的文件通常是由系统提供的,其扩展名为.h,因此也称为头文件或首部文件。C语言的头文件中包括各个标准库函数的函数原型。因此,在程序中调用某个库函数时,都必须包含该函数原型所在的头文件。本程序使用了3个库函数,即输入函数scanf()、正弦函数sin()、输出函数printf()。sin()函数是数学函数,其头文件为math.h,因此在程序的主函数前用include命令包含了math.h。scanf()和printf()是标准输入/输出函数,其头文件为stdio.h,因此在主函数前用include命令包含了stdio.h。

本程序的主函数体分为两部分,一部分为声明部分,另一部分为执行部分。声明部分是指变量的类型声明。例1-1中未使用任何变量,因此无声明部分。

C语言规定,源程序中所有用到的变量都必须先声明,后使用,否则将会出错。这是编译型高级程序设计语言的一个特点,与解释型的BASIC语言是不同的。声明部分是C语言源程序结构中重要的组成部分。本程序使用了两个变量x、s,用来表示输入的变量和sin()函数值。由于sin()函数要求这两个变量必须是双精度浮点型,因此用类型说明符double来说明这两个变量。声明部分后的4行为执行部分或称为执行语句部分,用以完成程序的功能。执行部分的第1行是输出语句,调用printf()函数输出提示字符串,提示用户输入变量x的值。第2行为输入语句,调用scanf()函数,接受键盘上输入的数据并存入变量x中。第3行调用sin()函数并把函数值存入变量s中。第4行调用

printf()函数输出变量s的值,即x的正弦值。程序结束。

运行本程序时,首先输出提示信息“input number:”,这是由执行部分的第一行完成的。用户根据提示从键盘上输入任意数字,如5,按Enter键,输出计算结果。

在例1-1和例1-2中用到了输入/输出函数scanf()和printf(),这里先简单介绍一下其格式,以便后面使用(具体内容见第3章)。

scanf()函数和printf()函数分别称为格式输入函数和格式输出函数,其作用是按指定的格式输入/输出值。这两个函数的参数由以下两部分组成。

(1) 格式控制字符串

格式控制字符串是一个字符串,必须用双引号括起来,表示输入/输出变量的数据类型。在printf()函数中,允许在格式控制字符串内出现非格式控制字符,这时会原样输出非格式控制字符。

(2) 参数表

参数表中给出了输入/输出变量。当有多个变量时,用英文逗号分开。例如:

```
printf("sin of %lf is %lf\n",x,s);
```

其中,%lf为格式控制字符,表示按双精度浮点型处理。%lf出现了两次,分别对应x变量和s变量。其余字符为非格式控制字符,因此原样输出。

**【例1-3】**输入两个整数,并输出其中较大的数。

```
#include<stdio.h>
int max(int a,int b);          /*自定义函数说明*/
void main()
{
    int x,y,z;                /*变量声明*/
    int max(int a,int b);     /*函数声明*/
    printf("input two numbers:\n");
    scanf("%d%d",&x,&y);     /*输入x,y值*/
    z=max(x,y);              /*调用max()函数*/
    printf("maxmum=%d",z);   /*输出*/
}
int max(int a,int b)
{
    if(a>b){
        return a;
    }else{
        return b;           /*返回主函数*/
    }
}
```

主函数

max()函数



程序运行结果:

```
input two numbers:
5 7
maxmun=7
```

本程序由两个函数组成,即主函数 `main()` 和 `max()` 函数。函数之间是并列关系,可从主函数中调用其他函数。`max()` 函数的功能是比较两个数,然后把较大的数返回给主函数。`max()` 函数是一个用户自定义函数,因此在主函数中调用应先予以声明。可见,在程序的声明部分中,不仅可以有变量声明,而且可以有函数声明。

程序的执行过程:首先输出提示信息,提示用户输入两个数,按 Enter 键后由 `scanf()` 函数语句接收输入的两个数并送入变量 `x`、`y` 中,然后调用 `max()` 函数,并把 `x`、`y` 的值传递给 `max()` 函数的参数 `a`、`b`。在 `max()` 函数中比较 `a`、`b` 的大小,把较大的数返回给主函数的变量 `z`,最后输出 `z` 的值。

### 1.3.2 C语言程序的结构特点及书写规则

#### 1. C语言程序的结构特点

C语言程序结构有以下特点。

1) C语言程序是由函数(如 `main()` 函数和 `max()` 函数)组成的,每一个函数完成相对独立的功能,函数是C语言程序的基本模块单元。`main` 是函数名,函数名后面的一对圆括号内为函数的参数。参数可以省略,但圆括号不能省略。

2) 一个C语言程序总是从 `main()` 函数开始执行。主函数执行完毕,程序执行结束。

3) C语言编译系统区分字母大小写。C语言把大小写字母视为不同的字符,并规定每条语句均以分号结束。分号是语句不可缺少的组成部分。

4) 主函数 `main()` 既可以放在其他函数之前,也可以放在其他函数之后。习惯上,将主函数 `main()` 放在最前面。

5) C语言程序中调用的函数既可以是由系统提供的库函数,也可以是由用户根据需要而自定义的函数。

#### 2. C语言程序的书写规则

C语言程序的书写规则如下。

1) C语言程序是由一个主函数和若干个其他函数组成的。

2) 函数名后必须有圆括号,函数体放在花括号内。

3) C语言程序必须用小写字母书写。

4) 每条语句的末尾加分号。

5) 可以一行多句。