

■ 杨柯 田丹 / 主编

C语言 程序设计

 北京理工大学出版社
BEIJING INSTITUTE OF TECHNOLOGY PRESS

“十三五”普通高等教育应用型本科院校重点建设系列教材

C 语言程序设计

主 编 杨 柯 田 丹

副主编 靳 新 梁 爽

王 岩 吴晓艳

内 容 简 介

本书介绍了 C 语言的基本概念、基本数据类型、运算符和表达式、数据的输入与输出、流程控制语句、数组、函数、指针、编译预处理、结构体和共用体，以及文件。通过对本书的学习，读者可以真正学会用 C 语言去编写程序，并从中体会学习 C 语言的无穷乐趣。

版权专有 侵权必究

图书在版编目 (CIP) 数据

C 语言程序设计/杨柯, 田丹主编. —北京: 北京理工大学出版社, 2019. 4 (2019. 5 重印)

ISBN 978 - 7 - 5682 - 6942 - 1

I. ①C… II. ①杨… ②田… III. ①C 语言 - 程序设计 - 教材 IV. ①TP312. 8

中国版本图书馆 CIP 数据核字 (2019) 第 071776 号

出版发行 / 北京理工大学出版社有限责任公司

社 址 / 北京市海淀区中关村南大街 5 号

邮 编 / 100081

电 话 / (010) 68914775 (总编室)

(010) 82562903 (教材售后服务热线)

(010) 68948351 (其他图书服务热线)

网 址 / <http://www.bitpress.com.cn>

经 销 / 全国各地新华书店

印 刷 /

开 本 / 787 毫米 × 1092 毫米 1/16

印 张 / 20. 25

字 数 / 477 千字

版 次 / 2019 年 4 月第 1 版 2019 年 5 月第 2 次印刷

定 价 / 49. 80 元

责任编辑 / 梁铜华

文案编辑 / 曾 仙

责任校对 / 周瑞红

责任印制 / 李志强

图书出现印装质量问题, 请拨打售后服务热线, 本社负责调换

前 言

C 语言是国内外广泛使用的计算机语言之一，是计算机从业人员应该掌握的一种重要的程序设计工具，发展至今仍保持着强大的生命力。C 语言具有语法丰富、结构清晰、使用灵活、执行高效、可移植性强等特点，被大多数高等学校的计算机专业作为第一程序设计基础课程来安排教学，对学生掌握程序设计的基础方法、进行计算思维方法训练、掌握问题的抽象表示和解决方法都能起到很好的作用。此外，C 语言的专业地位很重要，对 C 语言掌握的程度将直接影响多门后续课程及相关技术的学习和掌握。

本书结合编者多年的教学实践经验编写而成，结构合理、逻辑性强、文字流畅、通俗易懂、便于自学。针对应用型人才的培养要求，经过精心策划、准确定位，全书以清晰的概念、大量的图例，分 12 章详细介绍了 C 语言编程的基本理论及实用知识，并由点到面、由浅入深、循序渐进地介绍了数据类型、运算符和表达式、数据的输入与输出、选择结构、循环结构、数组、函数、指针、结构体和文件的操作，还阐述了程序设计的思想与技术。

本书的所有案例均已在 Visual C++ 6.0 集成环境中调试通过，并在每章都附有精心设计的习题，希望读者学好语法知识，积极上机实践、勤思考、多模仿，通过上机演示例题和习题来逐步掌握 C 语言程序设计的基本思想和方法。另外，本书还配有相应的实验内容，以增强实践性环节的学习。通过对本书的学习，读者可以真正学会用 C 语言去编写程序，并从中体会学习 C 语言的无穷乐趣。

本书由杨柯、田丹担任主编，靳新、梁爽、王岩、吴晓艳担任副主编。具体分工：第 1、2、11 章由田丹和吴晓艳编写，第 3、9、10 章由靳新和梁爽编写，第 4~8 章由杨柯编写，第 12 章由王岩编写，全书由杨柯统稿。

由于编者的水平和经验有限，书中难免存在错误或疏漏之处，敬请同行专家和读者批评指正。

第1章 C语言概述	(1)
1.1 程序和程序设计	(1)
1.1.1 程序	(1)
1.1.2 程序设计	(1)
1.1.3 结构化程序设计方法	(2)
1.2 算法	(3)
1.2.1 算法的概念	(3)
1.2.2 算法的特性	(3)
1.2.3 算法的表示方法	(3)
1.3 C语言简介	(6)
1.3.1 C语言的发展及其特点	(7)
1.3.2 C程序的结构组成与书写风格	(7)
1.4 C程序的上机操作	(11)
1.4.1 C程序的执行过程	(11)
1.4.2 在 Visual C++ 环境下建立和运行 C 程序的步骤	(12)
小结	(16)
习题	(16)
上机实验	(17)
第2章 C程序设计基础知识	(18)
2.1 常量和变量	(18)
2.1.1 常量	(18)
2.1.2 变量	(22)
2.1.3 标识符	(23)
2.2 C语言的基本数据类型	(24)
2.2.1 整型数据	(24)
2.2.2 实型数据	(26)
2.2.3 字符型数据	(27)
2.3 运算符与表达式	(28)
2.3.1 算术运算符与表达式	(28)



2.3.2 赋值运算符与表达式	(32)
2.3.3 逗号运算符与表达式	(34)
小结	(35)
习题	(36)
上机实验	(38)
第3章 顺序结构程序设计	(41)
3.1 C语句概述	(41)
3.2 赋值语句	(43)
3.3 字符数据的输入/输出	(44)
3.3.1 字符输出函数	(44)
3.3.2 字符输入函数	(45)
3.4 格式的输入/输出	(45)
3.4.1 格式输出函数	(46)
3.4.2 格式输入函数	(50)
3.5 顺序结构程序设计举例	(52)
小结	(54)
习题	(55)
上机实验	(59)
第4章 选择结构程序设计	(61)
4.1 关系运算符与关系表达式	(61)
4.1.1 关系运算符	(61)
4.1.2 关系表达式	(62)
4.2 逻辑运算符与逻辑表达式	(63)
4.2.1 逻辑运算符	(63)
4.2.2 逻辑表达式	(64)
4.3 if语句	(65)
4.3.1 if语句(单分支)	(65)
4.3.2 if...else...语句(双分支)	(66)
4.3.3 if...else if语句(多分支)	(68)
4.3.4 if语句的嵌套	(70)
4.4 条件运算符与条件表达式	(70)
4.5 switch语句	(71)
4.6 选择结构程序设计举例	(73)
小结	(79)
习题	(80)
上机实验	(84)
第5章 循环结构程序设计	(87)
5.1 while语句	(87)
5.2 do...while语句	(89)



5.3	for 语句	(91)
5.4	循环的嵌套	(94)
5.5	break 语句和 continue 语句	(97)
5.5.1	break 语句	(97)
5.5.2	continue 语句	(98)
5.6	几种循环语句的比较	(99)
5.7	循环结构程序设计举例	(100)
	小结	(103)
	习题	(104)
	上机实验	(110)
第 6 章	数组	(112)
6.1	一维数组的定义与引用	(112)
6.1.1	一维数组的定义	(112)
6.1.2	一维数组的引用	(113)
6.1.3	一维数组的初始化	(114)
6.1.4	一维数组应用实例	(115)
6.2	二维数组的定义与引用	(120)
6.2.1	二维数组的定义	(120)
6.2.2	二维数组的引用	(121)
6.2.3	二维数组的初始化	(121)
6.2.4	二维数组应用实例	(122)
6.3	字符数组的定义与引用	(124)
6.3.1	字符数组的定义	(124)
6.3.2	字符数组的初始化	(124)
6.3.3	字符数组的引用	(125)
6.3.4	字符数组的输入与输出	(126)
6.3.5	字符串处理函数	(128)
6.3.6	字符数组应用实例	(131)
	小结	(134)
	习题	(134)
	上机实验	(142)
第 7 章	函数	(145)
7.1	函数概述	(145)
7.2	函数的定义与调用	(146)
7.2.1	无参数函数	(147)
7.2.2	有参数函数	(150)
7.3	函数的声明	(154)
7.4	函数的嵌套调用	(155)
7.5	函数的递归调用	(157)



7.6	数组作为函数参数	(161)
7.7	局部变量和全局变量	(166)
7.8	变量的存储类别	(170)
7.8.1	动态存储方式与静态存储方式	(170)
7.8.2	局部变量存储类别	(171)
7.8.3	全局变量存储类别	(173)
7.8.4	存储类别小结	(176)
7.9	内部函数和外部函数	(177)
小结		(179)
习题		(180)
上机实验		(189)
第8章	指针	(194)
8.1	指针概述	(194)
8.2	指针和指针变量	(195)
8.2.1	定义一个指针变量	(195)
8.2.2	指针变量的引用	(196)
8.2.3	指针变量作为函数参数	(199)
8.3	指针和数组	(202)
8.3.1	指向数组元素的指针	(202)
8.3.2	通过指针引用数组元素	(203)
8.3.3	数组名作函数参数	(206)
8.3.4	指向多维数组的指针和指针变量	(212)
8.4	指针和字符串	(216)
8.4.1	字符串的表示形式	(216)
8.4.2	使用字符串指针变量与字符数组的区别	(219)
8.5	指针和函数	(220)
8.5.1	函数指针变量	(220)
8.5.2	指针型函数	(221)
8.6	指针数组和指向指针的指针变量	(223)
8.6.1	指针数组	(223)
8.6.2	指向指针的指针变量	(226)
小结		(228)
习题		(230)
上机实验		(238)
第9章	预处理命令	(241)
9.1	概述	(241)
9.2	宏定义	(241)
9.2.1	无参宏定义	(241)
9.2.2	带参宏定义	(245)



9.3 文件包含	(249)
9.4 条件编译	(251)
小结	(253)
习题	(254)
第10章 结构体与共用体	(257)
10.1 结构体数据	(257)
10.1.1 结构体数据类型的定义	(257)
10.1.2 结构体变量的定义	(258)
10.1.3 结构体变量的引用	(259)
10.1.4 结构体变量的初始化	(260)
10.1.5 结构体数组	(260)
10.1.6 结构体指针	(262)
10.2 结构体指针的应用——链表	(265)
10.3 共用体数据类型	(268)
10.3.1 共用体的概念	(268)
10.3.2 共用体的引用	(269)
10.4 枚举类型	(269)
10.5 类型重定义 typedef	(271)
小结	(272)
习题	(272)
上机实验	(277)
第11章 位运算	(278)
11.1 位运算符	(278)
11.2 位段	(282)
小结	(284)
习题	(284)
第12章 文件	(287)
12.1 文件概述	(287)
12.1.1 C语言操作文件	(287)
12.1.2 文件类型指针	(288)
12.2 文件的打开与关闭	(288)
12.2.1 文件的打开 (fopen 函数)	(289)
12.2.2 文件的关闭 (fclose 函数)	(290)
12.3 文件的读写操作	(291)
12.3.1 字符读写函数 fgetc 和 fputc	(291)
12.3.2 字符串读写函数 fgets 和 fputs	(294)
12.3.3 数据块读写函数 fread 和 fwrite	(296)
12.3.4 格式化读写函数 fscanf 和 fprintf	(297)
12.4 文件定位	(298)



12.5 文件检测函数	(300)
12.5.1 文件结束检测函数 feof	(300)
12.5.2 读写文件出错检测函数 ferror	(300)
小结	(300)
习题	(301)
上机实验	(303)
附录	(304)
附录1 常用字符与 ASCII 码对照表	(304)
附录2 C 库函数	(305)
附录3 C 语言的运算符及其结合性	(310)
参考文献	(312)

第1章

C 语言概述

1.1 程序和程序设计

1.1.1 程序

计算机的高速发展和广泛应用，深刻地影响了人类社会的发展，计算机已成为人们生产劳动和日常生活中必备的重要工具。人们可以利用它来完成各种各样的事情，如计算、娱乐等。计算机能完成如此繁杂的工作，就是由于有不同程序的支持，而程序则是人们利用程序语言经过程序设计过程得到的智力结晶。所谓程序，就是一组计算机能识别和执行的指令序列，即用计算机语言对所要解决的问题中的数据以及处理问题的方法和步骤所做的完整而准确的描述。Pascal 之父，瑞士著名计算机科学家沃思（Niklaus Wirth）教授提出了一个公式：

$$\text{程序} = \text{算法} + \text{数据结构}$$

从中可以看出，数据结构和算法是程序的两个重要组成部分。数据结构是对数据的描述，指数据的逻辑结构和存储方法，在程序中应指定数据的类型和数据的组织形式；算法是对数据操作的描述，即要求计算机进行操作的步骤。

1.1.2 程序设计

程序设计是给出解决特定问题程序的过程，往往以某种程序设计语言为工具，编写在这种语言下能解决某个问题的程序。程序设计一般包含以下 4 个步骤：

第 1 步，分析问题，建立数学模型。

对问题进行分析，找出已知的数据和条件，确定输入、处理及输出对象。将解题过程归纳为一系列数学表达式，建立各种量之间的关系，即建立解决问题的数学模型。

第 2 步，确定数据结构和算法。

针对建立的数学模型确定数据结构，选择合适的算法加以实现。

第 3 步，编制程序。



根据确定的数据和算法，用程序语言描述出来，即编写程序代码。

第4步，调试程序。

在计算机上运行已编好的程序，分析得到的结果，进行测试和调整，直至得到预期结果。

因此，编写一个完整的程序，涉及数据结构、算法、编程语言和程序设计方法4个方面的问题。

1.1.3 结构化程序设计方法

程序设计方法按照程序的结构性质，有结构化程序设计与非结构化程序设计之分。前者是指具有结构性的程序设计方法与过程，它具有由基本结构构成复杂结构的层次性，后者反之。其中，结构化程序设计方法是公认的面向过程编程应遵循的基本方法和原则。结构化程序设计方法的基本思想是：把一个复杂问题的求解过程分阶段进行，将每个阶段处理的问题都控制在人们容易理解和处理的范围内，就像玩积木游戏那样，用简单的模块搭建起复杂的程序。

结构化程序设计强调程序设计风格和程序结构的规范化，提倡清晰的结构。怎样才能得到一个结构化的程序呢？具体应采取方法为：自顶向下、逐步细化，模块化设计，结构化编码。首先把一个复杂的大问题分解为若干相对独立的小问题。若小问题仍复杂，则继续细化，直至将小问题分解为语句。然后，对应每一个小问题编写一个功能上相对独立的程序块。这种具有一定功能的程序块被称为模块，确切地说，模块是程序对象的集合，模块化就是把程序划分成若干个模块，每个模块完成一个确定的功能，把这些模块集中起来组成一个整体，就可以完成对问题的求解。结构化编码，指程序设计基于三种基本结构——顺序结构、选择结构和循环结构。由这三种基本结构组成的程序，就是结构化程序。

1. 顺序结构

顺序结构是最简单的一种基本结构，表示程序中的各操作是按照它们出现的先后顺序执行的。

2. 选择结构

选择结构也称分支结构，表示程序的处理步骤出现了分支，它需要根据某一特定的条件选择其中的一个分支执行。选择结构有单分支、双分支和多分支三种形式。

3. 循环结构

循环结构表示程序反复执行某个（或某些）操作，直到某条件为假（或为真）时才终止循环。循环结构的基本形式有两种：当型循环和直到型循环。

1) 当型循环

在当型循环中，先判断条件，当满足给定的条件时，就执行循环体，并在循环终端处流程自动返回循环入口处；如果条件不满足，则退出循环体直接到达流程出口处。因为是“当条件满足时执行循环”，即先判断后执行，所以称为当型循环。

2) 直到型循环

在直到型循环中，从结构入口处直接执行循环体，在循环终端处判断条件，如果条件不满足，就返回入口处继续执行循环体，直到条件为真时再退出循环到达流程出口处。因为是“直到条件为真时为止”，即先执行后判断，所以称为直到型循环。

1.2 算 法

1.2.1 算法的概念

算法就是为解决一个特定的问题而采取的方法和步骤。好比解数学题就有解题的步骤和过程。计算机要完成一定的任务，也是按照指定的步骤执行一系列的操作，计算机中的算法是指用计算机语言解决问题的方法和步骤。因此，算法是程序的组成部分之一。解决一个问题的步骤往往多种多样，也就是说，算法通常不唯一。

1.2.2 算法的特性

一个有效的算法应具有以下特性：

1. 有穷性

一个算法应包含有限的操作步骤而不能是无限的。

2. 确定性

算法中每一个步骤必须是计算机可以明确操作的，而不能是含糊的、模棱两可的。

3. 有零个或多个输入

输入是指在执行算法时需要的数据，既可以来自程序中的对象，也可以来自程序外部。

4. 有一个或多个输出

算法的一系列步骤操作完后，得到的结果就是输出。

5. 有效性

算法中的每一个步骤必须是计算机能有效执行的，并能得到确定的结果。

1.2.3 算法的表示方法

对算法的描述有多种方法，一般可以归纳为两类，即文字描述方法和图形描述方法。在文字描述方法中，常用的表示法为自然语言和伪代码；在图形描述方法中，常用的表示法是传统流程图和 N-S 流程图。



1. 自然语言

自然语言就是人们日常使用的语言。自然语言表示法虽然通俗易懂，但文字冗长，使用不方便，容易出现“歧义性”。所以，只有在描述简单问题时，才会使用自然语言表示法。

【例 1-1】 输入三个数 A、B、C，然后输出最大数 MAX。用自然语言写出算法。

算法表示：

- (1) 输入 A，B，C。
- (2) 若 A 大于 B，则将 A 赋值给 MAX；若 A 小于 B，则将 B 赋值给 MAX。
- (3) 若 C 大于 MAX，则将 C 赋值给 MAX。
- (4) 输出 MAX，MAX 即最大数。

2. 传统流程图

流程图也称框图，是用一些几何框图、流程线和文字说明表示各种类型的操作。形象直观，易于理解。美国国家标准协会（American National Standards Institute, ANSI）规定了一些常用的流程图符号，用于表示一定的含义，如图 1.1 所示。

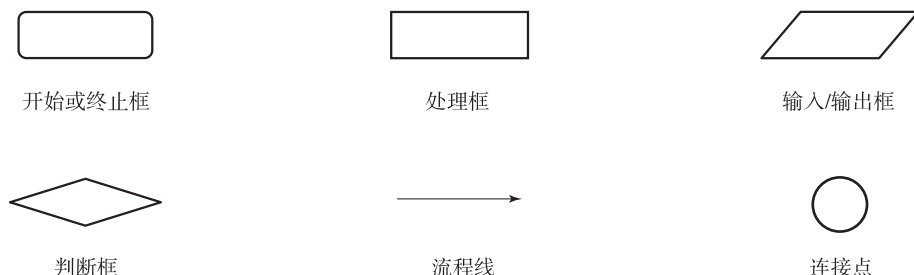


图 1.1 流程图中的符号

传统的流程图用流程线指出各框图的执行顺序，形象直观，简单方便。下面用传统的流程图来表示三种基本结构。

1) 顺序结构

图 1.2 所示为顺序结构。其中，A、B 两个框顺序执行，即在执行完 A 后，接着执行 B。

2) 选择结构

选择结构又称分支结构。如图 1.3 所示，判断框里放的是条件 P，判断条件 P 成立与否。若成立，则执行 A 框；不成立，则执行 B 框。图中的“成立”与“不成立”可以用“真”与“假”表示，还可以用字母“T”与“F”或“Y”与“N”表示。

3) 循环结构

循环结构又称重复结构，即反复执行某一部分的操作。循环结构可分为当型循环、直到型循环。

(1) 当型循环。

如图 1.4 所示，先执行判断框，当条件 P1 成立时，就执行 A 框操作，执行完 A 后，再判断条件 P1，若成立，再执行 A，如此反复，直至条件 P1 不满足，就结束循环。

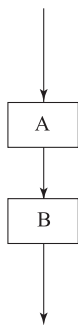


图 1.2 顺序结构

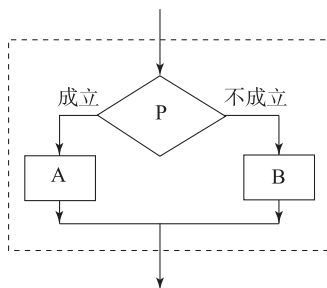


图 1.3 选择结构

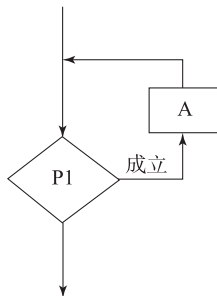


图 1.4 当型循环

(2) 直到型循环。

如图 1.5 所示，先执行 A 框，然后判断条件 P2，若条件不成立，则再执行 A，然后再判断条件 P2，若条件不成立，再执行 A，如此反复，直至条件 P2 成立，就结束循环。

分析以上三种基本结构，有以下两个共同特点：

- (1) 只有一个入口、一个出口。
- (2) 结构内的每一部分都有机会被执行。

【例 1-2】将例 1-1 用传统流程图来描述其算法。

流程图如图 1.6 所示。

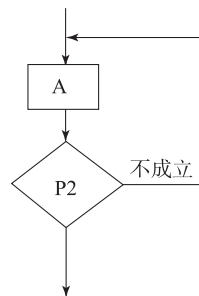


图 1.5 直到型循环

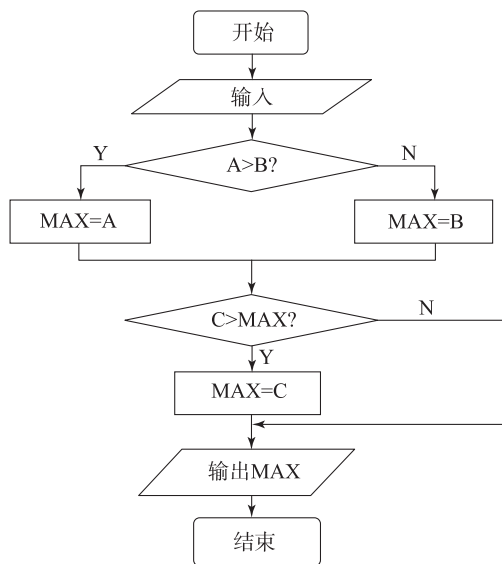


图 1.6 例 1-1 的流程图

3. N-S 流程图

传统的流程图，由于对流程线的走向没有任何限制，可以任意转向，在描述复杂的算法时所占篇幅较多，费时费力且不易阅读。一种新的流程图形式于 1973 年被美国学者 I. Nassi 和 B. Shneiderman 提出，这种流程图完全去掉了流程线，算法的每一步都用一个矩形框来描



述，把一个个矩形框按执行的次序连接起来，就是一个完整的算法描述。这种流程图称为N-S流程图，简称N-S图。下面用N-S图来描述三种基本结构，如图1.7~图1.10所示。图中的A或B是操作块，可以是输入、输出和赋值等基本功能的操作，也可以是三种基本结构。

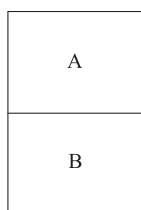


图 1.7 顺序结构

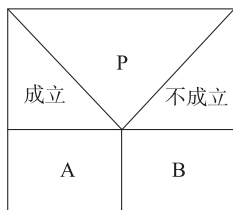


图 1.8 选择结构

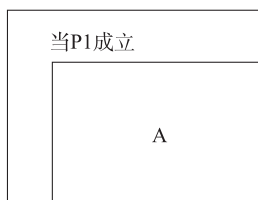


图 1.9 当型循环结构

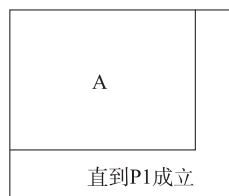


图 1.10 直到型循环结构

【例1-3】将例1-1用N-S图描述其算法。

N-S图如图1.11所示。

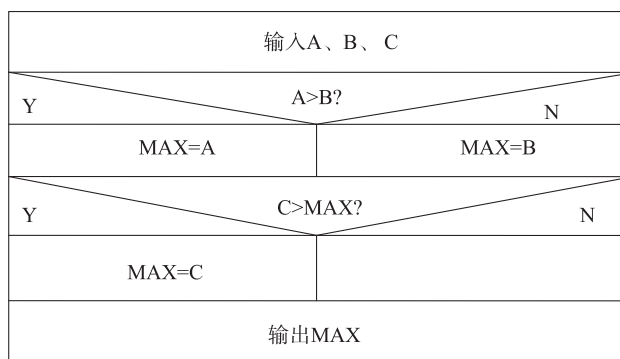


图 1.11 例 1-1 的 N-S 图

4. 伪代码

伪代码是用介于自然语言和计算机语言之间的文字和符号来描述算法，自上而下按照顺序编写，每一行表示一项操作，不用图形符号。因此，伪代码书写方便、结构紧凑，比较好懂，同时也方便向计算机程序过渡。

用伪代码写算法并无固定的、严格的语法规则，既可以只用英文，也可以中英文混用。只要把意思表达清楚，便于书写和阅读即可，书写的格式要写成清晰易读的形式。

1.3 C语言简介

程序设计语言是用于编写计算机程序的语言，按照语言级别可以分为低级语言和高级语言。低级语言有机器语言和汇编语言。机器语言是用二进制代码表示的计算机能直接识别和执行的一种机器指令的集合，它是计算机设计者通过计算机的硬件结构赋予计算机的操作功能。汇编语言是机器语言中用助记符代替机器指令的操作码，用地址符号或标号代替指令或操作数的地址。低级语言与特定的机器有关、功效高，但使用复杂、烦琐、费时、易出差

错。高级语言的表示方法是比低级语言更接近待解问题的表示方法，其特点是在一定程度上与具体计算机无关，易学、易用、易维护。C语言作为一种高级语言与其他高级语言相比功能更强，它既有高级语言的特点，又具有汇编语言的特点。它不但可以作为系统设计语言，编写工作系统应用程序，而且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。因此，C语言的应用范围广泛。

1.3.1 C语言的发展及其特点

20世纪60年代，英国剑桥大学的 Martin Richards 开发了C语言的祖先——BCPL语言（Basic Combined Programming Language）。1970年，Ken Thompson 在继承BCPL语言的许多优点的基础上发明了实用的B语言。1972年，贝尔实验室的 Dennis Ritchie 和 Brian kernighan 在B语言的基础上，作了进一步充实和完善，设计出了C语言。当时，设计C语言是为了编写UNIX操作系统。后来，C语言吸收了其他语言的优点，逐步成为实用性很强的语言。1983年，美国国家标准协会（ANSI）根据C语言问世以来各种版本对C的发展和扩充，制定了第一个C语言标准草案，称为ANSI C。1989年，ANSI公布了新的C语言标准，称为C89。1990年，国际标准化组织（ISO）接受C89为ISO C的标准，称为标准C。1995—2004年，ISO又对C语言标准进行了多次修订，将C89扩充为C99。但目前广泛流行的各种版本C语言还是以C89为基础，由不同软件公司提供的一些C语言编译系统并未完全实现C99建议的功能。

C语言发展迅速，而且成为最受欢迎的语言之一，主要是因为它具有强大的不同于其他语言的功能。C语言既具有高级语言的特点又具有低级语言的特点，使之得以存在和发展，并具有生命力。C语言有以下主要特点：

- （1）C语言简洁、紧凑，使用方便、灵活。ANSI C一共只有32个关键字，9种控制语句，程序书写形式自由。
- （2）运算符丰富。C语言共有34种运算符（见附录3）。C语言把括号、赋值、逗号等都作为运算符处理，因此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运算。
- （3）数据结构丰富。C语言可以实现各种复杂的数据结构的运算。
- （4）具有结构化的控制语句。由于用函数作为程序的模块单位，所以C语言是完全模块化和结构化的程序设计语言。
- （5）语法限制不太严格，程序设计自由度大。
- （6）C语言允许直接访问物理地址，能进行位（bit）操作，能实现汇编语言的大部分功能，可以直接对硬件进行操作。由于C语言可用于编写系统软件，因此有人把它称为中级语言。
- （7）C语言生成目标代码质量高，程序执行效率高。
- （8）与汇编语言相比，用C语言编写的程序具有更好的可移植性。

1.3.2 C程序的结构组成与书写风格

任何一种计算机程序语言，都具有特定的语法规则和一定的表现形式。按照规定的结构