

- 解决初学者的疑惑和难点
- 训练编程者的思想和方法
- 导学基本知识和编程技巧
- 导引独立编程和上机调试



C语言 程序设计导学

杜友福 主编



科学出版社
www.sciencep.com

清华大学出版社
清华大学出版社
清华大学出版社
清华大学出版社



程序设计导学

C 语 言 程 序 设 计 导 学

杜友福 主 编

科 学 出 版 社

北 京

内 容 简 介

本书为C语言程序设计课程导学和辅助教材,既可与《C语言程序设计》配套使用,亦可独立使用。全书共十三章,每章都由五部分组成:学习要求、内容要点、典型例题分析、习题、部分答案。

本书特别适合于本专科非计算机专业的初学者,也可供计算机等级考试者和其他各类学习者使用和参考。

图书在版编目(CIP)数据

C语言程序设计导学/杜友福主编. —北京:科学出版社,2003.12

ISBN 7-03-012637-8

I. C… II. 杜… III. C语言-程序设计 IV. TP312

中国版本图书馆CIP数据核字(2003)第115844号

责任编辑:冯贵层 王雨舸/责任校对:王望荣

责任印制:高 嵘/封面设计:晓 阳

科 学 出 版 社 出 版

北京东黄城根北街15号

邮政编码:100717

<http://www.sciencep.com>

湖北省京山金美印刷有限责任公司印刷

科学出版社出版 各地新华书店经销

*

2004年1月第 一 版 开本:787×1092 1/16

2004年1月第一次印刷 印张:15

印数:1 10 000 字数:344 000

定价:20.60元

(如有印装质量问题,我社负责调换)

前 言

C 语言以其独特的功能,受到了广大学生和使用者的喜爱。但由于C 语言的语法复杂性和语言灵活性,许多初学者学习起来感到有一定的难度,甚至在学完C 语言后,还不知其所以然。初学者对需要掌握哪些知识点,哪些是重点,哪些是难点,为什么会出现错误,如何解决这些错误,可能都很茫然。基于此,作者结合多年的教学经验编写了本书,旨在帮助学习者能够学好和用好C 语言。

本书共十三章,每章都由五部分组成。第一部分是学习要求,给出了需要掌握、了解的内容及基本要求,使学习者有一个明确的方向。第二部分是内容要点,对每一章的知识点进行了归纳和总结,明确了重点,指出了难点,对典型错误的表现、出错的原因进行了分析与解释,使学习者能够做到心中有数,以便更有效地进行学习。第三部分是典型例题分析,每一道题目指出了要求掌握和了解的知识点:选择题不仅给出了答案,还分析了各种选项对在什么地方、错在何处;填空题在进行比较详尽分析的基础上给出了答案;编程题既给出了程序代码,也对题目所涉及的算法或原理进行了分析,能够使学习者从中体会C 语言的基本概念、基本知识、基本方法及程序分析与设计的技巧。第四部分是习题,包含选择题、填空题和编程题等题型,希望以大量的题目来加强对C 语言基本知识和编程方法的训练。第五部分是部分习题的答案,给出了选择题和填空题的全部答案,对于编程题,只给出了难度相对比较大的题目的答案。希望学习者在做每一道题目时,能够像第三部分那样,去分析一下为什么。编程题的答案仅供参考,学习者可以采取更好的方法去实现,这在学习C 语言时是做得到的。书中除基本概念题和基础知识题外,大多数题目可以由学习者自己在机器上去亲自调试一下。这样,一来可以验证答案,二来可以通过调试和运行加强理解和掌握。只有通过大量而有效的训练,才能达到学好和用好C 语言的目的。

本书由杜友福教授任主编,并负责全书的统稿。第一、二、三章由胡必鑫副教授编写,第四、五章由李新玉副教授编写,第六、七、八、九章由杜友福教授编写,第十章由李克清副教授编写,第十一、十二、十三章由周云才副教授编写。

由于作者水平有限,书中难免会有疏漏和不足,敬请各位读者和专家提出宝贵意见,以便再版时予以修订。

编者

2003 年 10 月

目 录

第一章 程序设计基本概念	(1)
1.1 学习要求	(1)
1.2 内容要点	(1)
1.3 典型例题分析	(3)
1.4 习题	(8)
1.5 部分答案	(9)
第二章 数据类型、运算符和表达式	(11)
2.1 学习要求	(11)
2.2 内容要点	(11)
2.3 典型例题分析	(13)
2.4 习题	(18)
2.5 部分答案	(22)
第三章 顺序结构程序设计	(24)
3.1 学习要求	(24)
3.2 内容要点	(24)
3.3 典型例题分析	(25)
3.4 习题	(27)
3.5 部分答案	(29)
第四章 选择结构程序设计	(30)
4.1 学习要求	(30)
4.2 内容要点	(30)
4.3 典型例题分析	(33)
4.4 习题	(40)
4.5 部分答案	(43)
第五章 循环结构程序设计	(44)
5.1 学习要求	(44)
5.2 内容要点	(44)
5.3 典型例题分析	(47)
5.4 习题	(54)
5.5 部分答案	(59)
第六章 数组	(61)
6.1 学习要求	(61)
6.2 内容要点	(61)
6.3 典型例题分析	(64)
6.4 习题	(70)
6.5 部分答案	(81)
第七章 函数	(83)
7.1 学习要求	(83)
7.2 内容要点	(83)

7.3	典型例题分析	(86)
7.4	习题	(92)
7.5	部分答案	(105)
第八章	用户标识符的作用域和存储类	(108)
8.1	学习要求	(108)
8.2	内容要点	(108)
8.3	典型例题分析	(110)
8.4	习题	(114)
8.5	部分答案	(122)
第九章	编译预处理	(123)
9.1	学习要求	(123)
9.2	内容要点	(123)
9.3	典型例题分析	(125)
9.4	习题	(127)
9.5	部分答案	(132)
第十章	指针	(133)
10.1	学习要求	(133)
10.2	内容要点	(133)
10.3	典型例题分析	(138)
10.4	习题	(156)
10.5	部分答案	(166)
第十一章	结构体、共用体和用户定义类型	(174)
11.1	学习要求	(174)
11.2	内容要点	(174)
11.3	典型例题分析	(182)
11.4	习题	(193)
11.5	部分答案	(200)
第十二章	位运算	(203)
12.1	学习要求	(203)
12.2	内容要点	(203)
12.3	典型例题分析	(204)
12.4	习题	(206)
12.5	部分答案	(207)
第十三章	文件	(210)
13.1	学习要求	(210)
13.2	内容要点	(210)
13.3	典型例题分析	(212)
13.4	习题	(216)
13.5	部分答案	(221)
附录 C	语言上机指南	(226)

第一章 程序设计基本概念

1.1 学习要求

掌握程序及程序设计的基本概念,程序的运行过程,算法及其特点,算法的描述,C 语言程序的基本结构。

熟悉程序设计应遵循的基本原则。

了解利用计算机解决实际问题的过程、结构化程序设计方法,并在以后的程序设计中逐步应用,熟练掌握。

基本要求:能将日常生活中一些简单问题的解决步骤用算法描述。熟悉程序开发环境,学习上机基本操作。

1.2 内容要点

【知识点】

1. 程序和程序设计

程序是控制计算机运行的指令序列。程序设计就是设计、书写及检查程序的过程。程序设计语言是软件系统的重要组成部分,各种程序设计语言有自己的语法、词汇,一般可分为机器语言、汇编语言和高级语言三类。

高级语言程序通常要经过编辑、编译、装配链接后方能在计算机上运行。少数语言以解释方式执行。

2. 用计算机解决实际问题的基本过程

一般由分析问题、建立数学模型、选择合适算法、编写程序、调试、运行程序直至得到正确的结果等几个阶段所组成。其设计步骤可以分为:

(1) 确定要解决的问题。主要目的是搞清楚“做什么”。

(2) 算法设计与数据结构设计。主要任务是解决“如何做”,涉及两个方面的内容:一是数据结构设计(解决数据的表示与存储),另一个是算法设计(描述操作步骤)。

(3) 编写程序和调试程序。

(4) 整理资料,交付使用。

3. 算法

计算机算法是为计算机解题设计的有明确意义的运算步骤的有限集合。算法具有有穷性、确定性、有效性,有输入输出等特点。

表示一个算法,可以用不同的方法。常用的有自然语言、传统流程图、结构化流程图、伪代码、PAD图等。

4. 结构化程序设计的基本原则

结构化程序设计方法要求在设计程序时只能使用顺序、选择、循环3种基本结构。这3

种基本结构有以下共同特点:①只有一个入口;②只有一个出口;③结构内的每一部分都有机会被执行到;④结构内不存在“死循环”(无终止的循环)。

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。结构化程序设计方法的基本思路是,把一个复杂问题的求解过程分阶段进行,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。具体说,采取以下方法可以保证得到结构化的程序:①自顶向下;②逐步细化;③模块化设计;④结构化编码。

5. C 语言程序的基本结构

函数是C程序的基本单位。被调用的函数可以是系统提供的库函数,也可以是用户根据需要自己编制设计的函数。

一个函数由函数的首部和函数体两个部分组成。函数的首部描述该函数的外观(即该函数的名字,该函数需要外部提供哪些信息,执行完毕后能为外部提供哪些信息等),函数体则定义该函数的内涵(即该函数将对什么数据执行何种操作)。

一个C源程序至少包含一个函数(main函数)。一个C程序总是从main函数开始执行并在该函数结束。在程序中main函数可以放在程序的任意位置。

C程序书写格式自由,一行内可以写多个语句,一个语句也可以写在多行上。分号不是语句间的分隔符,而是C语句的必要组成部分。

C程序的输入和输出操作由函数(通常是库函数)来完成。

为增加程序的可读性,源程序中应当加上必要的注释。C程序中用/* */对程序的任意部分作注释。注释对程序的执行没有任何影响。

【重点】

(1) 算法的特点及简单算法的设计。

(2) C程序的基本结构。

【难点】

算法设计及其表示是本章的难点。算法在程序设计中占有极其重要的地位,是程序设计的灵魂。算法设计是程序设计中必不可少的步骤。结构化程序设计方法在其中起着重要的作用,用结构化程序设计方法设计算法可使编写的程序结构清晰,易于调试和修改。

流程图是算法的图形表示,可以清晰地反映控制结构的运作过程,对于制定和说明算法是非常有用的。

伪代码是算法的另一种常用的表示形式。伪代码不能在计算机上实际执行,仅仅用于程序员使用C等程序设计语言编写程序之前帮助其构思,而且与自然语言相似,非常方便和友好。

【典型错误分析】

1. 注释使用错误

包括忘记了注释的结束符*/,注释用/*开头,或用/*结束,或者在注释符号中插入了空格,即写成/* */的形式。

C语言编译程序对于注释部分(起始符/*与结束符*/之间)是不进行任何处理的,

也就是说,编译时,一旦遇到/*符号就对其后的任何符号不予处理,直到遇到*/符号为止。而对于其他部分则认为是C语句进行处理。因此注释符号使用错误会导致编译程序对程序的错误翻译,从而出现语法错误。

2. 分号使用错误

分号是C语句的一部分,漏掉分号或多分号都会导致编译程序对C语句无法正确理解。

3. 算法设计中的逻辑错误

这类错误编译程序往往不能发现或报错,但程序的结果可能是不正确的。对初学者来说,最常见的这类错误包括:忘记对计数器、总数等变量进行初始化;选用的用于控制输入循环结束的标记值也是一个合法的输入数据值;一个作为除数的变量的值为0等。

排除源程序中的错误(包括语法错误和逻辑错误)是进行程序设计所必须具备的能力,是对所学知识的综合检验过程,需要有清醒的头脑,严谨细致的工作作风,扎实的基础知识以及逻辑分析能力。对于初学者,调试程序时应注意以下两点:

(1) C编译程序尽管能够指出大多数程序的语法错误,但不可能完全准确地指出错误发生的源程序行,有时应当综合观察上下文、参考出错信息进行综合分析判断,不要死抠所报的出错行号。

(2) 程序中一处发生的错误会产生若干连带的错误信息,即若干条错误信息可能只有第一条是真正的错误。改掉这一错误,所有其他出错信息将随之消失。因此查错纠错应从最前面的出错点开始,纠正某个错误后,若不能发现其他错误,则应再次编译源程序,按照新的提示信息查找错误。

1.3 典型例题分析

【选择题】

例 1.1 一个C程序的执行是()。

- A. 从main()函数开始,直到main()函数结束
- B. 从第一个函数开始,直到最后一个函数结束
- C. 从第一个语句开始,直到最后一个语句结束
- D. 从main()函数开始,直到最后一个函数结束

解 本题的关键是掌握C程序的执行过程。

C程序中必须要有一个而且也只能有一个main()函数,不论main()函数处于程序中的什么位置,C程序执行时总是从main()函数开始,C程序的执行与函数位置无关,故选项B是不正确的。

C程序中除了顺序结构的语句执行与排列顺序一致外,选择结构和循环结构都会改变语句的执行顺序,因而选项C是不正确的。

如果程序中有多个函数,其他函数要通过调用才能执行,而且被调用函数在执行完毕后,必须返回到调用处,故选项D也不正确。

因此,正确答案应是选项A。

例 1.2 以下叙述中正确的是()。

- A. C 程序由主函数组成 B. C 程序由函数组成
C. C 程序由函数和过程组成 D. C 程序中的注释由 / * 开头，由 * / 结束
- 解 本题的关键是掌握 C 程序的构成。

在初学阶段，由于程序较小，可能只需 main() 函数即可，但解决规模较大的实际问题的最好办法是把程序分成较小的模块，每一模块完成一个独立的功能，这样便于管理和调试。C 语言的程序模块就是函数，因此读者应该明确且牢记：C 程序是由函数组成的。每个 C 语言的源程序可以包含多个函数，但只能有一个也必须有一个主函数。因此，选项 A 是片面的。在 C 程序中不存在“过程”这一程序结构，故选项 C 也是错误的。

程序设计中要求对源程序有详尽的注释，以便于阅读和软件的维护。C 程序中的注释行，由 / * 开头，由 * / 结束，“/”和“*”必须紧接着，在这两个字符之间不得插入空格。在选项 D 中“/”和“*”之间都插有空格，很多初学者常犯这样的错误。

因此，正确答案应是选项 B。

例 1.3 C 编译程序是()。

- A. C 程序的机器语言版本 B. 一组机器语言指令
C. 将 C 源程序翻译成目标程序的程序 D. 由制造厂家提供的一套应用软件

解 本题的关键是掌握编译程序的作用。

C 语言编写的源程序要执行，必须经过以下步骤：用编辑程序将源程序输入并存放在计算机的存储设备中，然后通过 C 编译程序将源程序翻译成计算机能识别的目标程序，最后通过连接装配程序将目标指令模块组合成完整的可执行程序才能在计算机上运行。C 编译程序是系统软件的一部分，并非应用软件。C 程序的机器语言版本称为目标程序。因此选项 A、D 均不正确。虽说能够运行的程序都是一组机器语言指令，但选项 B 是笼统的，不准确。

因此，正确答案应是选项 C。

例 1.4 C 函数由()构成。

- A. 函数首部 and 函数体 B. 执行语句和非执行语句
C. 自变量和因变量 D. 定义域和值域

解 本题的关键是掌握 C 函数的组成。

从数学角度来说，一个函数有自变量和因变量，也有其定义域和值域，但从 C 程序设计语言的角度来说，我们将其分为函数首部和函数体两个部分，函数的首部描述该函数的外观(即该函数的名字，该函数需要外部提供哪些信息，执行完毕后能为外部提供哪些信息等)，函数体则定义该函数的内涵(即该函数将对什么数据执行何种操作)。各种语句用来在函数体中定义各种数据及对这些数据进行操作。

因此，正确答案应是选项 A。

【填空题】

例 1.5 C 语言源程序文件的后缀是_____，经过编译后生成文件的后缀是_____，经过连接后生成文件的后缀是_____。

解 文件的后缀即文件的扩展名，主要用来表示该文件的类型。一般情况下，C 语言源程序文件的扩展名为“C”；C 语言源程序经过编译生成目标程序，其扩展名为“OBJ”；目

标程序必须经过连接生成可执行程序才能运行,其扩展名为“EXE”。因此,本题应分别填“C”,“OBJ”,“EXE”。

例 1.6 在事先不知道需要连续输入多少个数据的情况下,可用一个_____值控制输入循环的终止。

解 这是一道关于算法设计方面的题。算法设计中,经常会在某种条件下重复执行某些操作,这种情况称为循环。如,要连续输入多个数据,就可以用循环来实现,循环次数有时是固定数,有时是通过条件来控制的。就本题来说,由于不知道需要连续输入多少个数据,此时通常采用的办法是设置一个标记值(flag)来控制循环结束,当输入值为标记值时表示所有数据已输入完毕。显然该标记值应该小心选择,其值不应该在合法输入数据的范围内。类似的问题在循环中经常会遇到。因此本题应填“标记”。

例 1.7 _____是算法的图形表示。

解 解决问题的步骤称为算法。算法有多种表示方法,其中流程图是图形表示的算法,它可以清晰地反映控制结构的操作过程。因此本题应填“流程图”。

例 1.8 结构化程序设计方法使用的3种基本结构是_____、_____和_____。

解 这是一道基本概念题。结构化程序设计方法要求程序中只使用顺序、选择、循环3种基本结构。这3种基本结构构成的算法不存在无规律的转向,属于结构化的算法,而且已经证明,由以上3种基本结构构成的算法可以解决任何复杂的问题。因此,本题应分别填“顺序”、“选择”、“循环”。

【阅读题】

例 1.9 下述算法用于求 $1 * 2 * 3 * 4 * 5$ 。设两个变量,一个变量p代表被乘数(同时用于存放每一步骤的当前所得乘积),一个变量i代表乘数。算法如下:

S1:使 $p=1$

S2:使 $i=1$

S3:使 $p * i$,乘积仍放在变量p中,可表示为 $p * i \rightarrow p$

S4:使i的值加1,即 $i+1 \rightarrow i$ 。

S5:如果i不大于5,返回重新执行步骤S3以及其后的步骤S4和S5,否则,算法结束。最后得到p的值就是5!的值。

如何改动使其成为求 $1 * 3 * 5 * 7 * 9 * 11$ 。如果要求 $1+2+\dots+100$,应如何修改?

解 我们首先分析一下上述算法中每一步的作用。步骤S1、S2两步是对变量赋初值,即设置初始乘积为1,第一个乘数为1;S3是进行乘法运算,并将结果存放到当前所得乘积p中;S4通过改变i的值得到下一个乘数;S5判断是否所有的运算已执行完毕。

求 $1 * 3 * 5 * 7 * 9 * 11$ 与求 $1 * 2 * 3 * 4 * 5$ 的差别主要在于,下一个乘数比前一个乘数多2,且最后一个乘数是11。通过以上分析可知,只需作很少的改动即可:

S1: $1 \rightarrow p$

S2: $1 \rightarrow i$

S3: $p * i \rightarrow p$

S4: $i+2 \rightarrow i$

S5:若 $i \leq 11$,返回S3;否则,结束

可以看出,用这种方法表示的算法具有通用性、灵活性。S3 到 S5 组成一个循环,在实现算法时,要反复多次执行 S3、S4、S5 等步骤,直到某一时刻,执行 S5 步骤时经过判断,乘数 i 已超过规定的数值而不返回 S3 步骤为止。此时算法结束,变量 p 的值就是所求结果。

如果对这个算法适当修改实现求累加和,只需改动以下几点:

- (1) 使 p 的初值为 0(为什么? 请读者考虑)。
- (2) S3 中乘法运算改为加法运算。
- (3) S5 中改为与 100 比较。

由于计算机是进行高速运算的自动机器,可以高速地、不厌其烦地进行重复性操作。计算机中多次重复执行同类操作称为循环(或迭代)。所有计算机高级语言中都有实现循环的语句。因此,上述算法不仅是正确的,而且是计算机能实现的较好的算法。

【算法设计题】

例 1.10 开发一个程序求全班学生某门课程的平均成绩。要求程序每次运行时班级人数可为任意。

解 我们用结构化程序设计方法(“自顶向下、逐步求精”的方法)来解决这个问题并用伪代码来表示求精过程中得到的算法。为使初学者比较容易读懂,我们用中文作为伪代码。首先,顶层描述程序的整个功能,然后将其分解为一系列较小的任务,并按照执行顺序列出这些任务。每一步的求精都是一个完整的算法说明,只是每层的细节在变化。重复上述求精过程直到表示每一任务的伪代码能够转化为程序设计语言(如 C 语言)指令或程序段为止。

对于上述问题,顶层可用一句伪代码来表示:

求未知人数班级的平均成绩

首先我们考虑需要使用的数据量(变量)。显然,本例中我们需要使用的数据量有:总成绩(记为 $total$),成绩个数计数器(记为 $counter$),每次输入的成绩(记为 $score$),全班平均成绩(记为 $average$)。对数据进行分析后,我们再来考虑程序应该如何完成上述功能。许多程序可从逻辑上分为 3 个阶段:①初始化阶段(对程序变量初始化);②处理阶段(输入数据值并对程序中的变量作相应调整);③终止阶段(计算和打印最终结果)。本程序也不例外,因此我们可以将顶层分解为以下 3 个小任务:

初始化变量;

输入、求和及对输入的成绩计数;

计算并打印班级平均成绩;

对初始化阶段,由于每次输入的成绩值 $score$ 和平均成绩 $average$ 会被后面的读入过程覆盖,故不需要初始化。我们做如下求精:

初始化总成绩 $total$ 为 0;

初始化计数器 $counter$ 为 0;

由于成绩个数是不定的,我们使用一个专门的值来指示数据输入的结束,这个值称为“标记值”(flag)。显然标记值必须小心选择,使它有别于准备接收的成绩值。因为考试成绩通常是非负的,我们可以选择 -1 作为标记值。因此我们可以让程序重复读入成绩值,当遇到标记值时结束读入。每读入一个成绩值,就将其加到总成绩 $total$ 中,同时计数器加

1。

因此第二步求精如下：

```
    读入第一个成绩放到 score 中；
    当 score 不等于标记值时，重复以下工作：
    {
        将 score 加到 total 中；
        计数器值 counter 加 1；
        读入下一个数据放到 score 中；
    }
```

显然，只有读入有效的成绩值后，才能求出平均值。因此第三步求精如下：

```
    如果计数器值 counter 非零，那么执行以下工作：
    {
        将 total 除以 counter 的结果放到 average 中；
        打印 average；
    }
```

否则执行

打印信息“没有输入成绩”

经过上述求精后，所有分解的任务已可以用C语言描述出来。如此实现的程序结构清晰，简单明了。本例的C语言程序如下：

```
/* 用标记值控制循环求全班平均成绩 */
#include <stdio.h>
main()
{
    float average,total;
    int counter,score;
    /* 初始化 */
    total=0.0;
    counter=0;
    /* 处理阶段 */
    printf("输入成绩,-1 表示输入结束");          /* 屏幕提示 */
    scanf("%d",&score);
    while (score!= -1)
    {
        total=total+score;
        counter++;
        printf("输入下一个成绩,-1 表示输入结束");    /* 屏幕提示 */
        scanf("%d",&score);
    }
    /* 终止阶段 */
    if(counter != 0)
    {
        average=total/counter;
```

```

        printf("全班平均成绩为%.2f\n", average);
    }
    else
        printf("没有输入成绩\n");
    return 0;
}
/* 表明程序成功结束 */

```

1.4 习 题

一、选择题

1. C 语言程序的基本单位是()。
 - A. 程序行
 - B. 语句
 - C. 函数
 - D. 字符
2. 在C程序中main()函数的位置()。
 - A. 必须作为第一个函数
 - B. 必须作为最后一个函数
 - C. 可以任意
 - D. 必须放在它所调用的函数之后
3. 一个C程序是由()。
 - A. 一个主函数和若干子程序组成
 - B. 一个或多个函数组成
 - C. 若干过程组成
 - D. 若干子程序组成
4. 编辑程序的功能是()。
 - A. 建立并修改源程序
 - B. 将C源程序翻译成目标程序
 - C. 调试程序
 - D. 命令计算机执行指定的操作
5. 运行一个C语言编写的程序,以下不需要的是()。
 - A. 编译程序
 - B. 汇编程序
 - C. 编辑程序
 - D. 连接程序
6. 以下描述正确的是()。
 - A. 解释方式不产生目标程序,编译方式产生目标程序
 - B. 解释方式执行速度快,编译方式执行速度慢
 - C. 解释方式是针对高级语言而言,编译方式则是针对汇编语言
 - D. 所有的高级语言全部采用解释方式执行
7. C程序中分号()。
 - A. 是语句之间的分隔符
 - B. 表示它前后的两个语句是并列关系
 - C. 是语句的结束标记
 - D. 可有可无

二、填空题

1. 函数体由符号 ① 开始,用符号 ② 结束。
2. 算法具有 ①、②、③ 和 ④ 等4个特点。
3. 一个C程序必须有一个 ① 函数。C程序都是从 ② 函数开始执行。
4. 任何问题的求解都是按指定的_____执行一系列的动作。
5. 结构化程序设计方法采用的3种基本结构具有 ①、②、③ 和 ④ 等四个共同特点。
6. 计算机能直接识别的语言是_____。

7. C 程序开发的 4 个步骤是 ①、②、③ 和 ④。
8. 用计算机解决实际问题一般经过 ①、②、③ 和 ④ 4 个步骤。
9. 在计算机内存储器中最小的存储单位称为 ①，其中只能存放 ② 数。

三、阅读题

说明以下 C 程序的各个组成部分。

```
main()
{
    int a,k,m;
    a=10;
    k=2;
    m=1;
    a=(k+m)*k/(k-m);
    printf("%d\n",a);
}
```

四、算法设计题

给出下列问题的算法，并画出流程图。

- (1) 求 A, B, C 三个数中的最大数。
- (2) 求 $1 \times 2 \times 3 \times \cdots \times 10$ 的值。
- (3) 交换 A 和 B 的值。
- (4) 求两个整数 A 和 B 的最大公约数和最小公倍数。
- (5) 判断整数 N 是否素数。
- (6) 计算函数：输入 X，输出 Y 的值。

$$Y = \begin{cases} 3X & (X < 1) \\ 4X - 1 & (X = 1) \\ 5(X - 1) + 6 & (1 < X < 5) \\ 6 - 3X & (X \geq 5) \end{cases}$$

1.5 部分答案

一、选择题

1. C 2. C 3. B 4. A 5. B 6. A 7. C

二、填空题

1. ① { ② }
2. ① 有穷性 ② 确定性 ③ 有效性 ④ 有输入输出
3. ① main() ② main()
4. 算法
5. ① 只有一个入口 ② 只有一个出口 ③ 结构内的每一部分都有机会被执行到
④ 结构内不存在“死循环”(无终止的循环)
6. 机器语言
7. ① 编辑 ② 编译 ③ 连接 ④ 运行

8. ① 确定要解决的问题 ② 算法设计与数据结构设计 ③ 编写程序和调试程序
④ 整理资料,交付使用
9. ① 位(bit) ② 一位二进制

三、阅读题

`main()` 是函数首部,“{ ”与“}” 之间的部分是函数体。在函数体中,第一行是变量定义,后 5 行为执行部分。

四、算法设计题

略