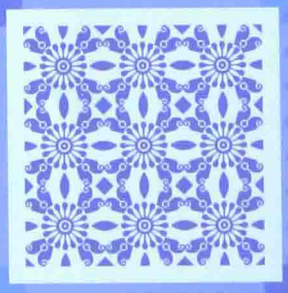
HZ BOOKS
华章教育

高等院校计算机教材系列

C LANGUAGE AND PROGRAMMING

C语言与程序设计

王瑞民 卢红星 主编
姬波 陈静 朱真峰 柳宏川 参编



机械工业出版社
China Machine Press

高等院校计算机教材系列

C LANGUAGE AND PROGRAMMING

C语言与程序设计

王瑞民 卢红星 主编
姬波 陈静 朱真峰 柳宏川 参编

机械工业出版社
China Machine Press

图书在版编目（CIP）数据

C 语言与程序设计 / 王瑞民，卢红星主编 . —北京：机械工业出版社，2015.2
（高等院校计算机教材系列）

ISBN 978-7-111-48860-6

Ⅰ. C… Ⅱ. ①王… ②卢… Ⅲ. C 语言 - 程序设计 - 高等学校 - 教材 Ⅳ. TP312

中国版本图书馆 CIP 数据核字（2014）第 295624 号

本书是学习 C 语言与程序设计的基础教材，主要包括绪论、C 语言程序的开发过程、基本数据类型、算术运算符与算术表达式、C 语言程序的输出与输入、选择结构程序设计、循环结构程序设计、函数、数组、结构体与共用体、指针、文件等内容，并且每章的后面附有习题供读者练习。

本书可作为高等院校计算机、信息、电子等相关专业本科生教材，也可供对 C 语言及程序设计感兴趣的读者自学使用。

出版发行：机械工业出版社（北京市西城区百万庄大街 22 号 邮政编码：100037）
责任编辑：李 艺　　责任校对：董纪丽
印　　刷：三河市宏图印务有限公司　　版　　次：2015 年 2 月第 1 版第 1 次印刷
开　　本：185mm × 260mm 1/16　　印　　张：16.75
书　　号：ISBN 978-7-111-48860-6　　定　　价：35.00 元

凡购本书，如有缺页、倒页、脱页，由本社发行部调换
客服热线：(010) 88378991 88361066　　投稿热线：(010) 88379604
购书热线：(010) 68326294 88379649 68995259　　读者信箱：hzjsj@hzbook.com

前　言

C 语言是一种比较流行的程序设计语言，也是全国计算机软件水平考试及 NIT 考试中的重要内容之一。本书是根据计算机相关专业本科生的培养目标和对“程序设计类”课程的基本要求，结合编者多年 C 语言与程序设计教学的实践经验编写而成的。

目前多数高校都将 C 语言与程序设计课程作为大学一、二年级计算机等相关专业的基础课程。本书基于程序设计的过程，详细介绍 C 语言的语法规则与程序设计的应用。针对初学者的特点，全书采用由浅入深、循序渐进的方式，并结合大量实例分析 C 语言的特点和程序设计方法：先讲一些容易理解的要点，难于理解的知识点放在后面章节介绍，而对于函数、指针这些既是重点又是难点的内容，则分成多个章节分别介绍。本书力求做到概念清晰、内容新颖、实用性强，使初学者既能理解 C 语言的语法和语义，又能掌握程序设计的方法。

本书具有以下 5 个特点：

1）内容表述深入浅出，通俗易懂。由于没有其他高级语言及相关程序设计的基础，难入门是大多数初学者所面临的主要问题。本书先讲述初学者容易掌握并且需要掌握的基础知识，而对于一些 C 语言的难点和程序设计的综合运用技巧，则放在后面章节中介绍。

2）在内容组织形式上，采用案例教学和启发式教学的方法。每章的例题和习题都是在以往的教学过程中针对基础概念、语法规则、程序设计基础等要点精心设计的，都是对必备知识点的有益补充，具有一定的连续性。为了让读者深入领会、运用所学知识，每道选择题都至少设置了一个正确答案。

3）以“程序设计”为主线，从程序设计的角度阐述 C 语言的基本概念、语法规则及基本操作方法，注重对读者进行程序设计方法及算法的训练，力图做到严格的理论与具体方法、算法的有机结合。

4）良好的源程序书写风格、程序设计风格和算法优化思想贯穿全书。全书注重程序书写风格的规范化和算法设计的优化，且所有源程序都通过了调试，读者可以直接编译、链接和运行。

5）吸取了往届学生提出的建议，其中包括 C 语言初学者、计算机软件水平考试的考生及 ACM 竞赛的选手。同时，编者还对读者学习和应用过程中容易混淆的概念和语法设计了相应的案例，并进行了深入的剖析。

本书由郑州大学的王瑞民、卢红星、姬波、陈静、朱真峰、柳宏川等共同编写。其中，第 1、2、3 章由朱真峰执笔；第 4、5、6 章由陈静执笔；第 7、8、16 章由卢红星执笔；第 9、10、13 章由王瑞民执笔；第 11、14、15 章由姬波执笔；第 12、17 章及附录由柳宏川执笔。此外，卢红星对全书的架构做了整体设计，王瑞民对全书进行了修改和统稿。

本书的编写得到了郑州大学信息工程学院各位同事的指导和帮助，往届的学生也对本

书的草稿提出了建设性的意见，编者在此对他们表示由衷的感谢。

本书的出版还参考了前辈和同行们的一些观点、资料和书籍，在此对相关的作者表示诚挚的感谢。感谢机械工业出版社的各位编辑，正是有了他们的帮助，本书才得以顺利出版。

由于时间仓促及编者水平有限，书中难免存在错误和不足，恳请各位读者批评指正。

编者

2014 年 11 月

教学建议

教学内容	教学要求	课时
第 1 章 绪论	程序设计语言、C 语言的发展与特点、算法基础、结构化程序设计、程序设计风格	2（理论）
第 2 章 C 语言程序的开发过程	C 语言程序的开发流程、C 语言程序的集成开发环境、C 语言程序的结构和语法规则	2（理论）+2（实践）
第 3 章 基本数据类型	标识符、C 语言的数据类型、常量、变量	2（理论）+2（实践）
第 4 章 算术运算符与算术表达式	算术运算符及优先级、算术表达式、类型转换	2（理论）+2（实践）
第 5 章 C 语言程序的输出与输入	C 语言语句、格式化输出、格式化输入、单个字符的输出与输入、顺序结构程序设计举例	4（理论）+2（实践）
第 6 章 选择结构程序设计	关系运算符与关系表达式、逻辑运算符与逻辑表达式、选择控制语句、选择结构程序设计举例	4（理论）+2（实践）
第 7 章 循环结构程序设计	循环控制语句、辅助控制语句、循环结构程序设计举例	4（理论）+2（实践）
第 8 章 函数	函数的定义、函数的值、函数的调用	4（理论）+2（实践）
第 9 章 数组	一维数组、二维数组、数组作为函数参数、数组应用举例	6（理论）+2（实践）
第 10 章 结构体与共用体	结构体类型的定义、结构体变量的定义与引用、结构体数组、共用体	4（理论）+2（实践）
第 11 章 指针	指针与指针变量、指针变量的定义与引用、指针变量应用举例	4（理论）+2（实践）
第 12 章 文件	文件概述、文件的打开与关闭、文件的读 / 写操作	2（理论）+2（实践）
第 13 章 其他运算符与表达式	运算符与表达式概述、赋值运算符与赋值表达式、逗号运算符与逗号表达式、条件运算符与条件表达式、自增自减运算符、位运算、表达式应用举例	4（理论）+2（实践）
第 14 章 变量的存储	变量的内存单元、变量的作用域、变量的生存期	4（理论）+2（实践）
第 15 章 再论指针	指针与数组、指针数组、函数指针、结构体指针、存储管理函数、链表	6（理论）+2（实践）
第 16 章 再论函数	函数的活动与存储空间的布局、参数的传递机制、递归函数的应用	4（理论）+2（实践）
第 17 章 预处理命令	宏定义、文件包含、条件编译	2（理论）+2（实践）
总课时	第 1 ～ 17 章建议课时	60（理论）
	课外实践建议课时	32（实践）

目　录

第1章 绪 论

简单地说，程序设计就是利用计算机语言编写程序的过程。每一个程序必须由特定的计算机语言编写，并需要一定的环境支持。设计程序时，还需要遵循一定的程序设计方法。本书着重介绍如何使用C语言设计高质量的源程序的方法。

本章将介绍程序设计语言的分类、C语言的发展简史与特点、算法基础、结构化程序设计和程序设计风格等几个方面的内容。

1.1 程序设计语言的分类

计算机语言是指计算机能够接受和处理的、具有一定格式的语言，是进行程序设计时最重要的工具之一。计算机语言分为低级语言和高级语言。

1.1.1 低级语言

低级语言依赖于所在的计算机系统，也称为面向机器的语言。由于计算机系统不同，使用的指令系统可能不同，因此，使用低级语言编写的程序的移植性较差。低级语言主要包括机器语言和汇编语言。

机器语言是由二进制代码“0”、“1”组成的机器指令序列。用机器语言编写的程序称为机器语言程序，机器语言程序能够被计算机直接识别并执行。但是，程序员直接编写或维护机器语言程序是比较困难的。

汇编语言是一种借用助记符表示的程序设计语言。汇编语言的每条指令都对应着一条机器语言代码。汇编语言也是面向机器的，即不同类型的计算机系统使用的汇编语言也不同。用汇编语言编写的程序，称为汇编语言程序。计算机不能直接识别和执行汇编语言程序。汇编语言程序必须由“汇编程序”翻译成机器语言程序，才能够在计算机上运行。这种“汇编程序”称为汇编语言的翻译程序。汇编语言适用于编写直接控制机器操作的底层程序。汇编语言与机器的联系仍然比较紧密，所以在使用中有明显的局限性。

1.1.2 高级语言

高级语言编写的程序易读、易修改、移植性好。但使用高级语言编写的程序不能直接在机器上运行，必须经过语言处理程序的转换，才能被计算机识别。按照转换方式的不同，可将高级语言分为解释型和编译型语言两大类。

所谓解释型转换，是将编写的程序逐句翻译，翻译一句执行一句，即边翻译边执行。转换工作是由解释器自动完成的。常见的解释性语言包括 BASIC 语言和 Perl 语言。解释型转换方式的优点是比较灵活，可以动态地调整、修改程序；缺点是效率比较低下，不能生成独立的可执行文件，即程序的运行不能脱离其解释器。

编译型语言编写的程序，经过翻译等处理后，可以脱离其语言环境而独立地执行。C 语言、Pascal 语言等大多数编程语言都属于编译型语言。

高级语言按其发展过程，又可分为面向过程、面向对象和面向构件三类语言。

面向过程语言的特点是：

1）采用模块分解与功能抽象的方法，自顶向下，逐步求精。

2）按功能划分为若干个基本的功能模块，形成一个树状结构。各模块间的关系尽可能的简单，功能上应相对独立。每一个功能模块内部都是由顺序、选择或循环等基本结构组成。

面向过程的语言能有效地将一个比较复杂的任务，分解成若干个易于控制和处理的子任务。任务的分解有利于程序的设计与维护。C 语言即属于面向过程的语言。

面向过程的程序是按照流水线方式执行的，即一个模块执行结束前，不能执行其他的模块，也无法动态地改变程序的执行方向。而人们实际处理事务时，总期望每发生一件事情就能处理一件事情，即程序应该从面向过程改为面向具体的应用功能。20 世纪 80 年代初期，非过程化的程序设计语言开始出现。非过程化程序设计语言的目标是实现软件的集成化，把相互联系的数据以及对数据的操作，封装成通用的功能模块。各功能模块可以相互组合，完成具体的应用。各功能模块还可以重复使用，而用户不必关心其功能是如何实现的。C++、Java 等，是典型的面向对象的语言。

面向构件的语言则提倡最大限度地进行资源共享和软件重用。面向构件编程也称为 Web 服务编程，其目标是将应用服务提供商开发出的各种构件，放在自己的服务器上，供网络上的其他人使用。例如 C#，即是面向构件的语言。

1.2 C 语言的发展简史与特点

使用高级语言编写的程序易读、易修改，对机器的依赖性较小，通用性好，但高级语言程序难以实现对计算机硬件的直接操作。C 语言之所以能长盛不衰，是因为其具有低级语言和高级语言的双重优点。确切地说，C 语言是介于高级语言与低级语言之间的一种特殊的语言。因此，有时也称为中级语言。

1.2.1 C 语言发展简史

C 语言的前身是一种较为简单，并接近于硬件的 B 语言。B 语言的主要思想来源于 BCPL（Basic Combined Programming Language）。B 语言曾被用于编写 UNIX 操作系统。但 B 语言过于简单，并且不能区分数据的类型。

20 世纪 70 年代初，贝尔实验室的 Dennis Ritchie（1983 年图灵奖得主，C 语言之父，UNIX 之父）首次提出了 C 语言。C 语言既克服了 B 语言过于简单、无数据类型的缺点，又保持了 B 语言精炼、接近于硬件的优点。1973 年，Dennis Ritchie 与同事合作，用 C 语言重新改写了 UNIX 操作系统（改写部分占原代码的 90% 以上）。随着 UNIX 操作系统的广泛使用，C 语言得到了迅速推广。之后，C 语言又被多次改进，并出现了众多版本。1983 年，美国国家标准化协会（ANSI）对 C 语言进行了发展和扩充，制定了 ANSI C，并在 1989 年被正式采用，简称 ANSI C'89。本书就是以 ANSI C'89 为标准介绍 C 语言的。

1.2.2　C 语言的特点

C 语言既可以用于编写系统软件，也可以用于编写应用软件。C 语言具有表达能力强、代码质量高、移植性较好等特点。具体表现在以下五个方面。

1）语言简洁、紧凑，使用方便、灵活。C 语言一共有 32 个关键字、9 种控制语句。C 语言程序书写形式自由，主要用小写字母表示。表示方法上力求通俗易懂，使用花括号“{”与“}”分别表示复合语句的开始与结束。

2）C 语言允许进行位、字节和地址等操作，可以对硬件进行编程，能实现汇编语言的大部分功能。但与汇编语言程序相比，C 语言程序生成的目标代码质量更高，并且更容易移植。C 语言程序的源代码基本上不用修改，经重新编译，就能够用于各种操作系统。

3）C 语言共有 44 种运算符。C 语言将括号、赋值、强制类型转换等都作为运算符处理，增强了数据的处理能力。

4）数据结构丰富，具有现代高级语言提供的大多数数据结构。C 语言的数据类型有整型、实型、字符型等基本数据类型，并允许在基本数据类型的基础上，按层次构造各种新的构造型数据类型，如数组类型、结构体类型、共用体类型等。此外，C 语言的数据类型还包括指针类型，能够实现各种复杂数据结构的运算，如链表、树、栈等。

5）C 语言提供了直接实现顺序、选择和循环三种基本结构的语句，并以函数作为程序的模块单位，因此 C 语言是一种比较理想的结构化编程语言。

当然，C 语言也有不足之处。例如，C 语言的语法限制不太严格，在增加程序设计灵活性的同时，对程序设计人员提出了更高的要求。编写程序时，程序员应该仔细检查程序，保证其正确性，而不要过分依赖 C 语言编译系统来检查错误。

1.3　算法基础

1.3.1　算法概述

问题的求解过程，是对数据对象的加工过程。问题的求解过程包括两个方面的内

容，即对问题涉及的“数据”进行描述和对“加工过程”进行描述。对“数据”的描述，称为“数据结构”；对“加工过程”的描述，称为算法。

广义地讲，算法是一个计算过程，是为解决一个问题而采取的方法和步骤。算法具有以下特点：

1）有穷性。一个算法包含的操作步骤是有穷的，而不能是无穷的，并且操作时一般还要求算法是合理可行的。例如，让计算机执行一个历时 1 万年的算法，虽然是有穷的，但这是不可行的。

2）确定性。算法的每个操作都是确定的，而不应该具有二义性。例如，判断“A 是否为老人”。到底多大岁数的人算老人？是 60 岁以上的还是 50 岁以上的？不同的当事人，对于 A 是否为老人，可能会得到不同的结论。

3）有零个或多个输入。“输入”指的是执行算法时，需要从外界获取的必要信息。例如，求两个数的最大公约数，可能需要输入两个整数。

4）有一个或多个输出。算法的目的是为了得到计算的结果。一般地，计算结果即为输出。没有输出的算法是没有意义的。

5）有效性。算法的每一个操作都应能有效地执行，并得到确定的结果。例如，若 $a<0$，在实数域范围内求 a 的算术平方根的算法就不会被有效地执行。

1.3.2 算法的结构化描述

算法的描述方法有多种，常用的有自然语言描述法、传统流程图描述法、结构化流程图描述法、伪代码描述法、PAD 图描述法等。这里仅介绍 N–S 流程图描述法。

N–S 流程图是由美国学者 I. Nassi 和 B. Shneiderman 于 1973 年提出的，用于描述三种基本结构的表示方法。按照 N–S 流程图描述方法，所有结构化的算法都可以写在一个矩形框内，但在该框内还可以包含其他的框。

1）顺序结构。如图 1-1 所示，A 和 B 两个框组成一个顺序结构。即执行 A 框所指定的操作后，接着执行 B 框所指定的操作。顺序结构是一种最基本的结构。

2）选择结构。选择结构也称为分支结构。该结构必须包含一个判断框。根据给定的条件是否成立，执行不同的操作。如图 1-2 所示，当条件 p 成立时，执行 A 操作；p 不成立时，执行 B 操作。

注意： 无论条件 p 是否成立，只能执行 A 框或 B 框中的一个，不可能既执行 A 框又执行 B 框，也不可能 A 框和 B 框都不执行。

3）循环结构。循环结构也称为重复结构，即反复执行某一部分操作。循环结构又可以分为两种。

一种称为当型循环结构，如图 1-3 所示。当条件 p 成立时，执行 A 框。执行完 A 框后，判断条件 p 是否成立，如果成立，再执行 A 框……如此反复。也即，当条件 p 成立时，总执行 A 框。

另一种称为直到型循环结构，如图 1-4 所示。先执行 A 框，然后判断条件 p 是否成立。如果条件 p 不成立，则再执行 A 框，然后再对条件 p 判断。如果条件 p 仍然不成

立，又执行 A 框……如此反复执行 A 框，直到给定条件 p 成立时，不再执行 A 框。

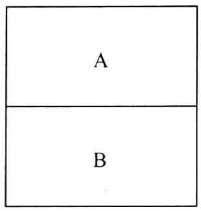


图 1-1　顺序结构

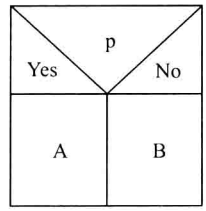


图 1-2　选择结构

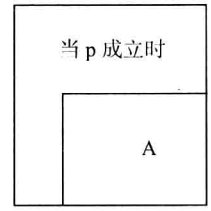


图 1-3　当型循环结构

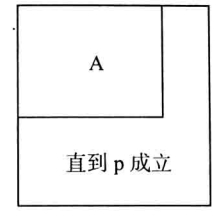


图 1-4　直到型循环结构

结构化的算法总能分解为若干个顺序、选择和循环等基本结构。因此，使用 N–S 流程图可描述任一结构化的算法。

1.4　结构化程序设计

1.4.1　程序设计

简单地说，程序设计就是使用计算机语言编写程序的过程。程序的设计过程可以细分为以下五个步骤。

1）分析问题：提出解决问题的可行方案。

2）确定算法：针对提出的可行方案，确定求解问题、完成任务的每一个细节步骤。

3）编写程序：使用计算机语言，把算法按照语法法则严格地描述出来，输入到计算机并存盘。

4）调试并运行程序：如果在运行过程中发现错误，需要仔细分析错误原因。更正后再次运行程序，直到程序准确无误，并得到正确的输出结果为止。

5）总结：写出书面报告。

其中，2）～4）三个步骤，即为一般所说的“程序设计”。

设计程序时，需要遵循一定的程序设计方法。每一个程序必须用某种计算机语言编写，并有必要的环境支持。因此可以认为：

程序 = 算法 + 数据结构 + 程序设计方法 + 语言工具和支持环境

算法设计、数据结构都有专门的课程介绍。本书着重介绍如何以 C 语言为工具进行程序设计的方法。

1.4.2　结构化程序设计概述

早期的非结构化语言中都包含 goto 语句，允许程序从一个地方直接转到另一地方。其优点是程序设计比较灵活方便，但不加限制地使用 goto 语句，会使得程序正确性的验证工作难以进行。同时，过多的 goto 语句，会使程序流程显得复杂和紊乱，维护成本急剧增加。这正是软件产生危机的根源之一。

结构化程序设计可以将任一复杂的求解算法都归结为三种特定的基本结构，即顺序结构、选择结构和循环结构。但明确规定，基本结构可以并列，可以嵌套，但不允许交叉。结构化的程序易于书写、阅读、修改和维护。

结构化程序设计对复杂问题的求解是分阶段进行的，每个阶段处理的问题都以人们容易理解和处理为准。在总体设计阶段，采用“自顶向下，逐步求精”的方法，将复杂问题的求解方案分解和细化为多个模块，按照层次结构将模块组装成一个软件系统。在详细设计和编码阶段，将每一模块的功能逐步分解和细化为一系列具体的处理步骤。

一个结构化的程序应具有以下特点：

1）程序符合“正确第一，效率第二”的质量标准。

2）程序由“模块”组成，模块之间可以跳转但不能随意地跳转。

3）程序只有一个入口、一个出口。

4）程序由顺序、选择和循环三种结构组成。

5）程序没有死循环。

1.5 程序设计风格

程序设计者只要遵循结构化程序设计的规定，就可以设计出结构化的程序。但由于C语言程序的语法规则比较自由，有较大的随意性，对于初学者来说，这种自由反而容易造成程序的条理不清、结构混乱。因此，建议初学者在开始学习程序设计时，应严格要求自己：遵循良好的程序设计风格，使得自己编写的程序易于调试和维护，不仅自己可以看得懂，而且别人也可以看得懂。

良好的程序设计风格包括源程序文档化、数据对象说明标准化、构造语句简单化、输入输出合理化、注重程序效率化等。

1）源程序文档化。编码的目的是产生源程序，但是为了提高程序的可维护性，源代码需要进行文档化。源程序文档化包括按照“见名知意”的准则给数据对象命名、安排适当的注释、按照标准格式书写源程序等。

2）数据对象说明标准化。按照一定的次序说明数据对象，有利于程序的测试、排错和维护。因此，数据对象的说明次序要固定；当用一个语句说明多个数据对象时，应按照数据对象的字母顺序排列；对于复杂的数据结构，增加注释以说明其特点及实现方法。

3）构造语句简单化。不能为了追求效率而使程序代码复杂化，应当尽量保证语句简单明了。为了便于阅读和理解，最好不要将多个语句写在同一行。不同层次的语句采用缩进形式，使程序的逻辑结构和功能特征更加清晰。避免复杂的判定条件，避免多重的循环嵌套。

4）输入输出合理化。即输入操作步骤和输入格式尽量简单；程序应能够自动检查输入数据的合法性、有效性，并报告必要的输入状态信息及错误信息；成批输入数据时，要使用数据或文件结束的标志，而不要用计数来控制；交互式输入时，应提供可用的选择和边界值；当程序设计语言有严格的格式要求时，应保持输入格式的一致性；输出的

数据应尽量表格化、图形化。

5）注重程序效率化。程序的效率主要指该程序对处理器和存储空间的使用效率，包括该程序的运行时间、对存储器的使用效率、输入输出操作的效率等。提高程序效率的根本途径在于选择良好的设计方法、良好的数据结构算法，而不是靠编程时对程序语句做调整。同时，提高效率不能损害程序的正确性、可读性和可靠性。

习题

一、选择题（每小题至少有一个正确选项）

1. 属于低级语言的计算机语言是（　　）。

A. C 语言　　B. 汇编语言　　C. Java 语言　　D. Pascal 语言

2. 面向过程语言的特点包括（　　）。

A. 自顶向下，逐步求精

B. 可以动态地改变程序的执行方向

C. 几个功能模块可以并行地执行

D. 能够将其他服务提供商开发的构件放在自己的服务器上，供网络用户使用

3. 与汇编程序相比，C 语言程序的优点包括（　　）。

A. 更容易移植　　B. 更容易阅读　　C. 目标代码质量较高　　D. 能够进行位操作

4. C 语言的运算符个数是（　　）。

A. 32　　B. 34　　C. 42　　D. 44

5. 有关结构化程序设计，正确的说法是（　　）。

A. 每个算法都必须包含三种基本结构

B. 每个结构化的算法都可以归结为三种基本结构

C. 三种基本结构可以相互嵌套

D. 三种基本结构可以交叉设计

6. 一个结构化的程序，所具有的特点包括（　　）。

A. 程序代码符合“正确第一，效率第二”的质量标准

B. 程序的入口是唯一的

C. 程序的出口是唯一的

D. 程序不包含死循环

7. 算法的特点包括（　　）。

A. 有穷性　　B. 确定性　　C. 可以没有输入　　D. 可以没有输出

二、简答题

1. 用 N-S 流程图描述算法：输入一个正整数，判断其是否为素数。

2. 简述什么是良好的程序设计风格。

3. 简述 C 语言的发展历史与特点。

第2章　C语言程序的开发过程

C语言程序的开发过程包括：使用编辑工具编写文本形式的C语言源文件，然后编译生成以机器代码为主的可执行程序。函数是C语言程序的基本组成单位，开发C语言程序的主要工作就是编写各个函数。

本章主要介绍C语言程序的开发过程、C语言程序的集成开发环境及C语言的结构和语法规则。

2.1　概述

给定一个可解的问题，首先要分析“需要做什么”和“如何做”。根据分析的结果，制定可行的解决方案，然后选用合适的算法描述方法，表示解决方案的每一个细节步骤。程序设计的主要工作正是从算法设计开始的。

开发一个C语言程序一般包括3个主要步骤，如图2-1所示。

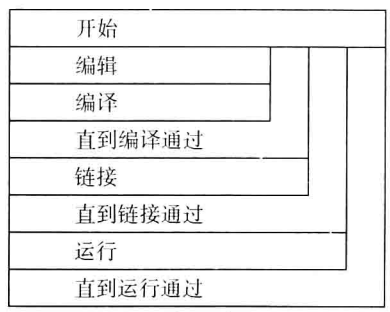


图2-1　C语言程序的开发过程

1）编辑。根据求解算法，开发人员使用任意一款文本编辑器编辑代码，生成源程序代码。

2）编译。编译时，编译器首先检查源程序中每条语句的词法和语法。当发现错误时，在屏幕上显示错误的位置、错误类型等相关信息。根据错误信息，重新使用编辑器进行查错并修改，然后重新编译，直到所有的词法和语法错误都被排除。

编译过程发现的错误可分为两类，一类是局部语法错误，例如，缺失了分号、逗号或者引用了错误的数据对象；另一类是程序内部上下文关系方面的错误，例如，需要使用的数据对象没有定义。

3）链接。编译后产生的目标文件是可重定位的程序模块，但不能直接运行。链接即将目标程序、库函数和其他目标程序链接到一起，生成可执行的程序。

源代码经过编译、链接，生成可执行程序后，还需要进行测试。测试的目的是发现程序的错误。一般通过输入一些实际数据来验证程序执行结果的正确性。如果程序执行过程中出现问题，或发现程序的输出结果不正确，需要设法找到出错的原因，并修改源程序，重新编译、链接，再测试，不断反复，直到程序正确无误。

当测试结果为正确的可执行程序时，才可以在操作系统的控制下使用。