



本书附赠由**虎奔教育**提供的学习卡一张

**2015**年考试专用  
根据教育部最新大纲编写

National Computer Rank Examination  
**全国计算机等级考试**  
**无纸化专用教材**  
**二级C语言**

王希更 路谨铭 编著

■ 全国计算机等级考试命题研究室 虎奔教育教研中心 审定



**手机版学习软件**

题库试题，一网打尽，覆盖**99%**最新真题题库  
按关键字快速查找指定试题，随时随地查看解析，**你懂的**



**虎奔科举网**

一学就懂，学完就会的课程  
10分钟1个知识点，所学即所考  
全职老师在线答疑，不懂就问  
量身制定学习计划，定时催促学习进度

**赠 428元**  
**等考大礼包**

手机软件  
PC版软件  
虎奔科举网体验班和优惠券  
随身学

清华大学出版社

全国计算机等级考试专业辅导用书

# 全国计算机等级考试 无纸化专用教材

## 二级 C 语言

王希更 路谨铭 编著



清华大学出版社  
北 京

## 内 容 简 介

本书严格依据最新颁布的《全国计算机等级考试大纲》编写,并结合了历年考题的特点、考题的分布和解题的方法。

本书分为10章,包括C语言概述、数据基础知识、顺序程序设计、选择程序设计、循环程序设计、函数、数组、指针、结构体与共同体、文件等内容。

本书的光盘提供强化练习、真考模拟环境、评分与视频解析、名师讲堂等模块。

本书适合报考全国计算机等级考试“二级C语言”科目的考生选用,也可作为大中专院校相关专业的教学辅导用书或相关培训课程的教材

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

版权所有,侵权必究。侵权举报电话:010-62782989 13701121933

### 图书在版编目(CIP)数据

全国计算机等级考试无纸化专用教材. 二级C语言 / 王希更、路谨铭编著. —北京:清华大学出版社,2015

全国计算机等级考试专业辅导用书

ISBN 978-7-302-38365-9

I. ①全… II. ①王… ②路… III. ①电子计算机—水平考试—自学参考资料②C语言—程序设计—水平考试—自学参考资料 IV. ①TP3

中国版本图书馆CIP数据核字(2014)第247051号

责任编辑:袁金敏

封面设计:傅瑞学

责任校对:徐俊伟

责任印制:杨 艳

出版发行:清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址:北京清华大学学研大厦A座

邮 编:100084

社总机:010-62770175

邮 购:010-62786544

投稿与读者服务:010-62776969, [c-service@tup.tsinghua.edu.cn](mailto:c-service@tup.tsinghua.edu.cn)

质量反馈:010-62772015, [zhiliang@tup.tsinghua.edu.cn](mailto:zhiliang@tup.tsinghua.edu.cn)

印 刷 者:三河市君旺印务有限公司

装 订 者:三河市新茂装订有限公司

经 销:全国新华书店

开 本:185mm×260mm 印 张:13.5 插页:4 字 数:353千字  
(附光盘1张)

版 次:2015年1月第1版

印 次:2015年1月第1次印刷

印 数:1~5000

定 价:35.00元

---

产品编号:062190-01

# 前言



全国计算机等级考试(National Computer Rank Examination, NCRE)是经原国家教育委员会(现教育部)批准,由教育部考试中心主办,面向社会,用于考查应试人员计算机应用知识与技能的全国性计算机水平考试体系。计算机等级考试相应证书的取得,已经逐渐成为考生计算机操作水平的衡量标准,另一方面,也为考生以后的学习和工作打下良好的基础。

随着教育信息化步伐的加快,按教育部要求,从2013年上半年开始,全国计算机等级考试将完全采用无纸化考试的形式。为了使教师授课和考生备考尽快适应考试形式的变化,本书编写组织具有多年教学和命题经验的各方专业人士,结合最新考试大纲,深入分析最新无纸化考试形式和题库,精心编写了本套无纸化专用教程。

本书具有以下特点。

## 1. 直击真考知识点

深入分析和研究历年考试真题,结合最新考试大纲和无纸化考试的命题规律,知识点的安排完全依据真考考点,并将典型真考试题作为例题讲解,使考生在初学时就能掌握知识点的考试形式。

## 2. 课后题帮助考生查缺补漏

为巩固考生对重要知识点的把握,本书每章均配有课后习题。习题均出自无纸化真考试题库,具有典型性和很强的针对性。

## 3. 提供无纸化真考环境

本书配套软件完全模拟真实考试环境,其中包括4大功能模块:选择题、操作题日常练习系统,强化练习系统,完全仿真的模拟考试系统以及真人高清名师讲堂系统。同时软件中配有所有试题的答案,方便有需要的考生查阅或打印。

## 4. 自助式全程服务

虎奔培训、虎奔官网、手机软件、YY讲座、虎奔网校、免费答疑热线、专业QQ群等互动平台,随时为考生答疑解惑;考前一周冲刺专题,更可以通过虎奔软件自动获取考前密押试卷;考后第一时间点评专题,帮助考生提前预测考试成绩。

本书由王希更和路谨铭担任主编,并完成了第1、2、7、8章的主要编写工作和全书的统稿工作,李媛、王小平、张永刚、石永煊担任副主编,李媛完成第3、4章的编写工作,王小平完成第5章的编写工作,张永刚完成第6章的编写工作,石永煊完成第9、10章的编写工作。参加本书编著工作的还有刘爱格、李鹏、戚海英、刘欣苗等。

由于时间仓促,书中难免存在疏漏之处,我们真诚希望得到广大读者的批评指正。

编者

# 目 录

|                          |    |                                                   |    |
|--------------------------|----|---------------------------------------------------|----|
| 第 1 章 初识 C 概述 .....      | 1  | 2.4.3 逻辑运算符与逻辑表达式 .....                           | 27 |
| 1.1 初识 C 语言 .....        | 1  | 2.4.4 关系运算符与关系表达式 .....                           | 27 |
| 1.1.1 C 语言的由来 .....      | 1  | 2.4.5 其他运算符 .....                                 | 28 |
| 1.1.2 C 语言的特点 .....      | 1  | 本章小结 .....                                        | 31 |
| 1.2 C 语言程序设计 .....       | 2  | 真题演练 .....                                        | 32 |
| 1.2.1 程序设计的基本概念 .....    | 2  | 巩固练习 .....                                        | 32 |
| 1.2.2 C 语言程序的构成 .....    | 3  | 第 3 章 顺序程序设计 .....                                | 34 |
| 1.2.3 C 程序书写格式的要点 .....  | 3  | 3.1 语句 .....                                      | 34 |
| 1.3 C 程序的开发过程 .....      | 4  | 3.2 数据的输入与输出 .....                                | 37 |
| 1.4 算法 .....             | 4  | 3.2.1 格式化输入/输出函数 .....                            | 37 |
| 1.4.1 算法的概念 .....        | 4  | 3.2.2 字符的输入/输出函数 .....                            | 48 |
| 1.4.2 算法的特性 .....        | 5  | 3.3 顺序结构程序设计 .....                                | 49 |
| 1.4.3 流程图 .....          | 5  | 本章小结 .....                                        | 51 |
| 1.5 结构化程序设计 .....        | 5  | 真题演练 .....                                        | 52 |
| 本章小结 .....               | 7  | 巩固练习 .....                                        | 54 |
| 真题演练 .....               | 7  | 第 4 章 选择程序设计 .....                                | 55 |
| 巩固练习 .....               | 8  | 4.1 概述 .....                                      | 55 |
| 第 2 章 数据类型、运算符与表达式 ..... | 10 | 4.2 条件选择语句——if 结构 .....                           | 55 |
| 2.1 标识符 .....            | 10 | 4.2.1 if 语句的第 1 种形式——<br>if 结构 .....              | 55 |
| 2.1.1 标识符的概念 .....       | 10 | 4.2.2 if 语句的第 2 种形式——<br>if-else 结构 .....         | 56 |
| 2.1.2 标识符的分类 .....       | 11 | 4.2.3 if 语句的第 3 种形式——<br>if-else if-else 结构 ..... | 57 |
| 2.2 常量和变量 .....          | 11 | 4.3 if 语句的嵌套使用 .....                              | 59 |
| 2.2.1 常量 .....           | 11 | 4.3.1 在 if 语句中嵌套具有<br>else 子句的 if 语句 .....        | 60 |
| 2.2.2 变量 .....           | 12 | 4.3.2 在 if 语句中嵌套没有<br>else 子句的 if 语句 .....        | 61 |
| 2.3 基本数据类型 .....         | 13 | 4.3.3 在 else 子句语中嵌套 if 语句 .....                   | 62 |
| 2.3.1 整型数据 .....         | 14 | 4.4 多分支选择语句——switch 结构 .....                      | 63 |
| 2.3.2 实型数据 .....         | 17 | 4.4.1 switch 结构的基本形式 .....                        | 63 |
| 2.3.3 字符型数据 .....        | 18 |                                                   |    |
| 2.3.4 类型的混合运算 .....      | 20 |                                                   |    |
| 2.4 运算符与表达式 .....        | 22 |                                                   |    |
| 2.4.1 赋值运算符与赋值表达式 .....  | 23 |                                                   |    |
| 2.4.2 算术运算符与算术表达式 .....  | 26 |                                                   |    |

|                                 |           |                               |            |
|---------------------------------|-----------|-------------------------------|------------|
| 4.4.2 带 break 语句的 switch 语句     | 65        | 6.6.2 静态局部变量 static           | 103        |
| 4.4.3 switch 语句的嵌套              | 66        | 6.6.3 寄存器变量 register          | 103        |
| 4.4.4 if 语句和 switch 语句的<br>相互转换 | 67        | 6.6.4 全局变量 extern             | 104        |
| 本章小结                            | 67        | 本章小结                          | 105        |
| 真题演练                            | 67        | 真题演练                          | 106        |
| 巩固练习                            | 69        | 巩固练习                          | 109        |
| <b>第 5 章 循环程序设计</b>             | <b>71</b> | <b>第 7 章 数组</b>               | <b>112</b> |
| 5.1 概述                          | 71        | 7.1 概述                        | 112        |
| 5.2 while 和 do-while 循环         | 71        | 7.2 一维数组                      | 113        |
| 5.2.1 while 循环                  | 71        | 7.2.1 一维数组的定义                 | 113        |
| 5.2.2 do-while 循环               | 73        | 7.2.2 一维数组元素的引用               | 114        |
| 5.2.3 while 循环和 do-while 循环     | 74        | 7.2.3 一维数组中元素的地址              | 115        |
| 5.3 for 循环                      | 74        | 7.2.4 一维数组的初始化                | 116        |
| 5.3.1 for 循环的一般形式               | 74        | 7.2.5 一维数组应用举例                | 118        |
| 5.3.2 for 循环的特殊形式               | 75        | 7.3 二维数组                      | 121        |
| 5.4 循环的嵌套                       | 77        | 7.3.1 二维数组概述                  | 121        |
| 5.5 for 循环与 while 循环的转换         | 79        | 7.3.2 二维数组的定义                 | 122        |
| 5.6 break 语句与 continue 语句       | 80        | 7.3.3 二维数组元素的引用               | 123        |
| 5.6.1 break 语句                  | 80        | 7.3.4 二维数组元素的地址               | 123        |
| 5.6.2 continue 语句               | 81        | 7.3.5 二维数组的初始化                | 124        |
| 5.6.3 break 和 continue 的区别      | 82        | 7.4 数组在程序中的应用                 | 125        |
| 本章小结                            | 83        | 7.5 字符数组                      | 127        |
| 真题演练                            | 83        | 7.5.1 字符数组的定义                 | 127        |
| 巩固练习                            | 85        | 7.5.2 字符数组元素的引用               | 128        |
| <b>第 6 章 函数</b>                 | <b>87</b> | 7.5.3 字符数组的初始化                | 128        |
| 6.1 概述                          | 87        | 7.5.4 字符数组的输入/输出              | 130        |
| 6.2 函数的定义                       | 89        | 7.6 字符串                       | 132        |
| 6.2.1 无参函数的定义                   | 90        | 7.6.1 字符串的输入函数<br>scanf、gets  | 132        |
| 6.2.2 有参函数的定义                   | 90        | 7.6.2 字符串的输出函数<br>printf、puts | 133        |
| 6.3 函数的参数和返回值                   | 91        | 7.6.3 求字符串长度的函数 strlen        | 134        |
| 6.3.1 函数的参数                     | 92        | 7.6.4 字符串复制函数 strcpy          | 134        |
| 6.3.2 函数的返回值                    | 94        | 7.6.5 字符串比较函数 strcmp          | 135        |
| 6.4 函数的调用                       | 95        | 7.6.6 字符函数的综合应用               | 136        |
| 6.4.1 函数调用的一般形式                 | 95        | 本章小结                          | 138        |
| 6.4.2 函数调用的方式                   | 97        | 真题演练                          | 138        |
| 6.4.3 函数的声明                     | 97        | 巩固练习                          | 140        |
| 6.5 变量的作用范围                     | 99        | <b>第 8 章 指针</b>               | <b>142</b> |
| 6.5.1 局部变量                      | 99        | 8.1 概述                        | 142        |
| 6.5.2 全局变量                      | 100       | 8.2 指针的定义                     | 142        |
| 6.6 变量的存储类别                     | 101       | 8.2.1 地址的概念                   | 143        |
| 6.6.1 自动变量 auto                 | 102       |                               |            |

|                                  |     |                                  |     |
|----------------------------------|-----|----------------------------------|-----|
| 8.2.2 指针的定义 .....                | 145 | 9.3.4 结构体变量成员的访问 .....           | 177 |
| 8.2.3 指针变量的操作 .....              | 145 | 9.4 共用体 .....                    | 179 |
| 8.2.4 指针的运算 .....                | 149 | 9.5 位运算符与位运算 .....               | 181 |
| 8.3 指针在数组中的应用 .....              | 150 | 本章小结 .....                       | 183 |
| 8.3.1 指针与一维数组 .....              | 150 | 真题演练 .....                       | 184 |
| 8.3.2 指向二维数组元素的<br>指针变量 .....    | 153 | 巩固练习 .....                       | 187 |
| 8.4 指针函数与函数指针 .....              | 153 | <b>第 10 章 文件</b> .....           | 190 |
| 8.4.1 指针作为函数参数 .....             | 154 | 10.1 文件概述 .....                  | 190 |
| 8.4.2 函数指针的定义 .....              | 156 | 10.2 文件指针 .....                  | 191 |
| 8.5 指针的其他用法 .....                | 156 | 10.3 文件的打开 .....                 | 191 |
| 本章小结 .....                       | 160 | 10.4 文件的关闭 .....                 | 192 |
| 真题演练 .....                       | 161 | 10.5 文件的读写 .....                 | 193 |
| 巩固练习 .....                       | 163 | 10.6 其他一些函数 .....                | 195 |
| <b>第 9 章 编译预处理、结构体和共用体</b> ..... | 166 | 本章小结 .....                       | 197 |
| 9.1 概述 .....                     | 166 | 真题演练 .....                       | 198 |
| 9.2 编译预处理 .....                  | 166 | 巩固练习 .....                       | 199 |
| 9.2.1 宏定义 .....                  | 167 | <b>附录一 C 语言的关键字</b> .....        | 200 |
| 9.2.2 文件包含与条件编译 .....            | 170 | <b>附录二 C 语言运算符的优先级和结合性</b> ..... | 201 |
| 9.3 结构体 .....                    | 172 | <b>附录三 C 常用函数库</b> .....         | 202 |
| 9.3.1 结构体的含义 .....               | 173 | <b>附录四 常用字符与 ASCII 码对照</b> ..... | 206 |
| 9.3.2 定义结构体变量 .....              | 174 | <b>附录五 巩固练习参考答案</b> .....        | 207 |
| 9.3.3 结构体变量初始化 .....             | 176 |                                  |     |

# 第 1 章 C 语言概述

C 语言是一种程序设计工具,用 C 语言进行编程的过程就是程序设计的过程。由于 C 语言既有高级语言的功能,又有低级语言的一些功能,因此既可以用它来编写系统软件,也可以用它来编写应用软件。

## 1.1 初识 C 语言

### 1.1.1 C 语言简介

C 语言是一种功能强大、应用广泛的编程语言,目前使用的很多应用软件及操作系统均由 C 语言程序设计开发。例如,我们用的操作系统,像 Linux 操作系统、UNIX 操作系统和 Windows 操作系统,都广泛应用了 C 语言的知识。再比如一些智能电器,它们的控制部分也大多是由 C 语言来进行处理的。C 语言深受广大编程人员的喜爱,在许多理工科类大学都开设了 C 语言程序设计课程。

当前,用于程序设计的高级语言数以百计,其中,C 语言以其强大的功能被称为是真正的程序员的编程语言。

C 语言虽然是高级语言,但它也可以完成许多只有低级语言才能完成的、面向机器的底层工作,因此也被称为“中间语言”。正是由于 C 语言的这些特性,决定了它能够成为一种重要的程序设计语言。在人们所使用的程序中,大多是由 C 语言编写的,例如,使用最多的计算机操作系统 Windows XP/Windows 7 中,有相当多的部分是由 C 语言编写的。

### 1.1.2 C 语言的特点

C 语言是结构化程序设计语言之一,和其他高级程序设计语言相比,C 语言具有以下几个显著的特点。

#### 1. 语言简练

C 语言中常用的关键字有 32 个,其中和数据类型有关的有 15 个,和存储类型有关的有 4 个,用于流程控制的有 12 个,以及一个运算符 `sizeof`。这些关键字压缩了其他高级语言中冗余的部分。

#### 2. 形式自由

C 语言程序书写形式自由,主要由小写字母表示。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。

#### 3. 数据类型丰富

C 语言中除了有整型、实型、字符型等基本数据类型以外,还提供了数组、指针和结构体等



构造数据类型,而且能够用来实现更复杂的数据结构,如线性表、链表、栈、队列、树和图等。

#### 4. 运算符丰富

C 语言中提供了 34 种运算符,表达式类型多样化,灵活使用 C 语言中的各种运算符可以实现在其他高级语言中难以实现的运算。

#### 5. 结构化语言

C 语言中有 if-else、while、switch 等结构化控制语句,并用函数作为程序的模块,便于实现程序的模块化。C 语言是完全模块化和结构化的语言。

#### 6. 程序设计灵活

C 语言程序书写格式自由,编程的自由度大,语法限制不太严格,如对数组下标越界不作检查,各种类型的变量可以通用。

#### 7. 直接访问物理地址和位运算

C 语言具有汇编语言的一些功能,比如允许直接访问物理内存,这是其他高级语言不具备的。它还能对数据进行按位操作。

## 1.2 C 语言程序设计

### 1.2.1 程序设计的基本概念

程序是指可以被计算机连续执行的多条指令的集合,是人与计算机进行“对话”的语言集合。人们将需要计算机做的工作写成一定形式的指令,并把它们存储在计算机的内部存储器中,接收到指令后,它就被计算机按指令操作顺序自动运行。

运用 C 语言进行程序设计的过程就称为 C 程序设计。用 C 语言编写的程序,称为 C 语言源程序,简称 C 程序。

程序设计就是程序员设计程序的过程。广义上说,程序设计就是用计算机解决一个实际应用问题时的整个处理过程,包括提出问题、确定数据结构、确定算法、编程、调试程序及书写文档等。具体流程如下:

(1) 提出问题: 提出需要解决的问题,形成一个需求任务。

(2) 确定数据结构: 根据需求任务提出的要求、指定的输入数据和输出结果,确定存放数据的数据结构。

(3) 确定算法: 针对存放数据的数据结构确定解决问题、实现目标的步骤。

(4) 编写程序: 根据指定的数据结构和算法,使用某种计算机语言编写程序代码,输入到计算机中并保存在磁盘上,简称编程。

(5) 调试程序: 消除由于疏忽引起的语法错误或逻辑错误; 用各种可能的输入数据对程序进行测试,使之对各种合理的数据都能得到正确的结果,对不合理的数据都能进行适当处理。

(6) 书写文档: 整理并写出文档资料。

### 1.2.2 C语言程序的构成

【例 1.1】一个简单的 C 语言程序。

```
#include<stdio.h>           /* include 文件包含命令 */
main()
{
    printf("Hello Huben! \n"); /* 输出 Hello Huben! */
}
```

运行本程序将输出以下一行内容：

```
Hello Huben!
```

程序分析：

(1) C 程序共分为两部分：main()之前的预处理部分和 main 函数部分。这里的 include 称为文件包含命令，被包含的文件通常由系统提供。

(2) C 程序由一系列函数组成，这些函数中必须有一个且只能有一个以 main 命名的函数，这个函数称为主函数，整个程序从主函数开始执行。在此例中，只有一个主函数而无其他函数。注意，main 这个单词必须小写，并且后面一对圆括号不能省略。

(3) 函数体由大括号“{ }”括起来。一个函数中所有的执行语句都写在函数体中。左大括号“{”表示函数体的开始，右大括号“}”表示函数体的结束。例 1.1 中的函数体只有一个 printf 输出语句。printf 是 C 语言中的输出函数，其功能是把输出结果显示到屏幕上。语句中的双引号用来表示需要输出的内容，双引号内的字符串将按原样输出；“\n”是换行符，即在输出“Hello Huben!”后回车换行。

(4) 语句后的“;”表示该语句结束，不能省略。

(5) “/\* ... \*/”表示注释部分。注释主要用于说明变量的含义和程序段的功能等，以提高程序的可读性，对程序的运行不起作用。

通过对上述 C 程序的分析，可以总结出 C 语言源程序的构成，有以下几个要点：

- (1) 一个 C 语言源程序由一个或多个函数组成。
- (2) 每个源程序必须有且只能有一个 main() 函数。
- (3) 除了 main() 函数外，程序中还可以有若干个其他函数。

C 语言程序的执行过程如下：

- (1) 程序从 main() 函数开始执行。
- (2) 其他函数通过调用的方式被执行。
- (3) 程序最后在 main() 函数中结束。

### 1.2.3 C 程序书写格式的要点

C 语言程序的书写格式比较自由，主要有以下几点要注意。

- (1) 标识符严格区分大小写。例如，a 和 A 表示两个不同的变量。
- (2) C 程序语句用分号“;”结束，分号是 C 语句的必要组成部分，不能缺少。但是在预处理命令、函数头、大括号“{ }”之后不能有分号。
- (3) 一行可以写多个语句，一个语句可以分写在多行。
- (4) 可以在程序的任何位置用“/\* ... \*/”对程序或语句进行注释。“/\* ... \*/”为多行注

释语句,注释以“/ \* ”开始,直到遇到“ \* /”结束,中间的内容均为注释部分。“/ \* ”和“ \* /”必须成对出现,且“/”和“ \* ”不可以有空格,同时注释之间不可以再嵌套“/ \* ”和“ \* /”。

小提示:程序中涉及的“,”、“;”、“{ }”与标识符、字符等标点或符号,必须在英文半角状态下输入。

## 1.3 C 程序的开发过程

C 程序开发的基本过程如图 1-1 所示。

### 1. 编辑

选择适当的编辑程序,将 C 语言源程序通过键盘输入到计算机,并以文件的形式存储到磁盘中。在 C 语言程序的编辑过程中,源代码文件使用 Microsoft Visual C++ 6.0 编辑器创建,文件名由用户自行定义,扩展名为 .c。

### 2. 编译

选择适当的编译程序,将 C 语言源程序转换为目标程序的过程称为编译。在编译过程中,可能会发现源程序中的一些语法错误,系统会给出“出错信息”,此时用户需要回到编辑阶段进行修改,直到编译通过。目标程序文件的主文件名与源程序的主文件名相同,扩展名为 .obj。

### 3. 连接

经过编译后生成的目标程序是不能直接执行的,需要经过连接之后才能生成可执行程序。连接是将编译形成的目标文件 .obj 和库函数、其他目录文件连接形成统一的、可执行的二进制文件 .exe。

### 4. 执行

经过编译、连接之后,源程序文件就生成了可执行文件,运行可执行的二进制文件 .exe 可得到程序的运行结果。如果在运行程序后得不到预期的结果,就要重复执行编辑、编译、连接和执行 4 个步骤。

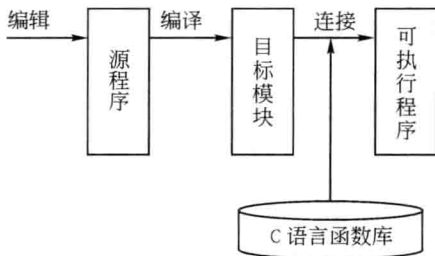


图 1-1 C 程序开发过程

## 1.4 算 法

### 1.4.1 算法的概念

在程序设计的过程中,“确定算法”是相当重要的。广义上讲,算法是为了解决问题而采取的方法和步骤。例如,描述跆拳道动作的图解,就是“跆拳道的算法”;一首歌曲的乐谱也可以称为该歌曲的算法。计算机科学中的算法是指为解决某个特定问题而采取的确定且有限的步骤,它是为了解决“做什么”和“怎么做”的问题。可以说,算法是程序设计的灵魂。著名科学家沃思(Nikiklaus Wirth)提出过一个公式:

数据结构 + 算法 = 程序

其中,数据结构是对数据的描述,也就是在程序中数据的类型和组织形式,而算法则是对操作步骤的描述,是使用程序设计语言来解决问题的“思路”。

### 1.4.2 算法的特性

一个算法应当具有以下5个特征。

(1) 确定性。算法中每一条指令必须有确切的含义,不能有二义性,对于相同的输入,必能得出相同的执行结果。

(2) 有穷性。一个算法应包含有限个操作步骤。也就是说,在执行若干个操作步骤之后,算法将结束,而且每一步都在合理的时间内完成。

(3) 可行性。算法中指定的操作,都可以通过已经实现的基本运算执行有限次后实现。

(4) 有零个或多个输入。在计算机上实现的算法,是用来处理数据对象的,在大多数情况下,这些数据对象需要通过输入来得到。

(5) 有一个或多个输出。算法的目的是为了求“解”,这些“解”只有通过输出才能得到。

算法可以用各种描述方法来进行描述,最常用的是伪代码和流程图。

### 1.4.3 流程图

算法一般比较抽象,可以用流程图来实现。流程图表示算法的好处是用图形来表示流程,形象直观,各种操作一目了然,而且不会产生歧义。正因为如此,流程图成为了描述算法的标准工具。

流程图的基本框架如图1-2所示。

图1-2(a)为流程图的起止框,表示算法的开始和结束。

图1-2(b)为流程图的处理框,主要用来表示“赋值”、“加减乘除”等操作。

图1-2(c)为流程图的判断框,用来根据给定的条件决定执行几条路径中的某一条路径。

图1-2(d)为流程图的输入/输出框,用来表示输入/输出操作。

图1-2(e)为流程图的流程线,用来表明程序流程的方向。

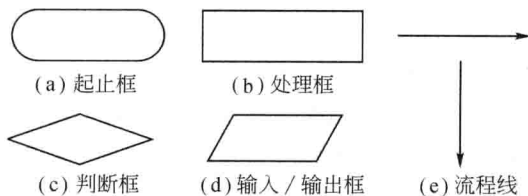


图 1-2 流程图的构成

## 1.5 结构化程序设计

C语言是一种结构化程序设计语言。程序主要由以下3种基本控制结构组成。

### 1. 顺序结构

赋值语句、输入/输出语句都可构成顺序结构(这些内容在本书第3章中进行介绍),当执行由这些语句构成的程序时,将按这些语句在程序中的先后顺序逐条执行,没有分支,没有转移。顺序结构可用图1-3所示的流程图表示。

### 2. 选择结构

if语句、switch语句可构成选择结构(这些内容在本书第4章中进行介绍)。当执行到这

些语句时,将根据不同的条件去执行不同的分支中的语句。选择结构可用图 1-4 所示的流程图表示。

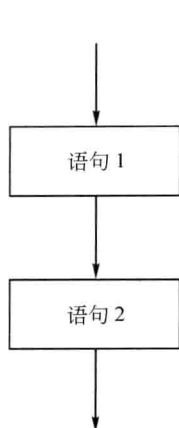


图 1-3 顺序结构

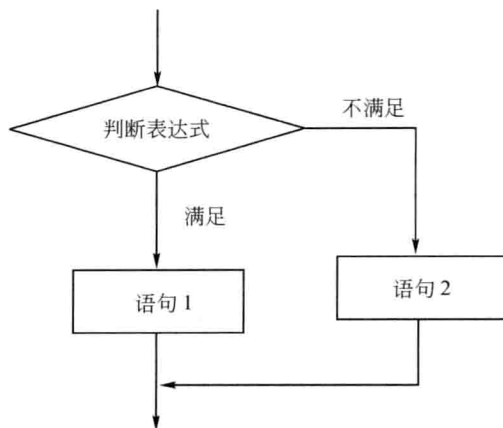


图 1-4 选择结构

### 3. 循环结构

在本书第 5 章中将介绍不同形式的循环结构。它们将根据各自的条件,使同一组语句重复执行多次或一次也不执行。循环结构的流程图如图 1-5 和图 1-6 所示。图 1-5 是当型循环,该循环的特点是:当指定的条件满足(成立)时,就执行循环体,否则就不执行。图 1-6 是直到型循环,该循环的特点是:执行循环体直到指定的条件满足(成立),就不再执行循环。

由 3 种基本结构组成的算法结构可以解决任何复杂的问题。由 3 种基本结构所构成的算法称为结构化算法;由 3 种基本结构所构成的程序称为结构化程序。

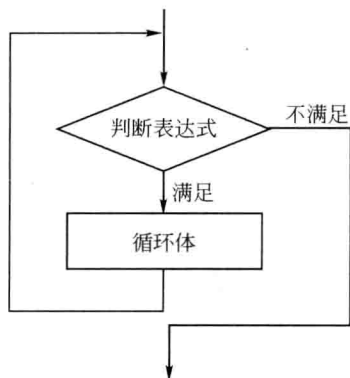


图 1-5 当型循环

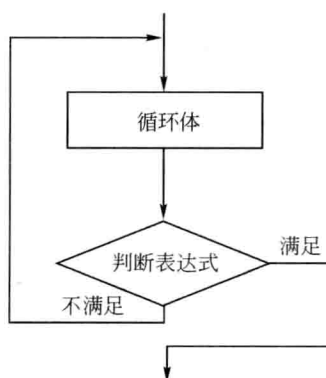


图 1-6 直到型循环

### 4. 结构化程序设计方法的要点

结构化程序设计方法的要点如下。

- (1) 要求程序清晰、可读性强。
- (2) 要求程序设计者按一定规范编写程序,而不能随心所欲地设计程序。

(3) 结构化程序设计方法规定了几种具有良好特性的“基本结构”，用它们作为构成程序的基本单元。“基本单元”应具有以下特点：

- ① 只有一个入口。
- ② 只有一个出口。
- ③ 没有永远执行不到的语句。
- ④ 没有死循环。

(4) 程序开发应当采取“自顶而下，逐步细化和模块化”的方法，即将一个大任务先分成若干个子任务，每个子任务就是一个模块，如果一个子任务还是太复杂，就再分解为若干个子任务，如此逐层分解，逐步细化。

## 本章小结

本章的内容主要讲述C语言的基础知识，程序设计的概念、算法以及结构化程序设计方法。本章内容是后面章节学习的基础，考生须掌握：

- (1) 程序的构成，main函数和其他函数。
- (2) 头文件，数据说明，函数的开始和结束标志。
- (3) 源程序的书写格式。
- (4) C语言的风格。

## 真题演练

### 选择题

(1) 以下叙述中错误的是( )。(2011年3月)

- A. C语言的可执行程序是由一系列机器指令构成的
- B. 用C语言编写的源程序不能直接在计算机上运行
- C. 通过编译得到的二进制目标程序需要连接才可以运行
- D. 在没有安装C语言集成开发环境的机器上不能运行C源程序生成的.exe文件

**【答案】D**

**【解析】**C语言程序生成的.exe文件是二进制的可执行文件，它不需要C语言集成开发环境的支持，可以直接运行。

(2) 以下叙述中正确的是( )。(2008年4月)

- A. C程序中的注释只能出现在程序的开始位置和语句的后面
- B. C程序书写格式严格，要求一行内只能写一个语句
- C. C程序书写格式自由，一个语句可以写在多行上
- D. 用C语言编写的程序只能放在一个程序文件中

**【答案】C**

【解析】在 C 语言中,注释可以加在程序中的任何位置,选项 A 错误。C 程序可以分模块写在不同的文件中,编译时再将其组合在一起,选项 D 错误。C 程序的书写风格很自由,不但一行可以写多个语句,还可以将一个语句写在多行中。所以正确答案为选项 C。

(3) C 语言源程序名的后缀是( )。(2007 年 9 月)

- A. .exe                      B. .c                      C. .obj                      D. .cp

【答案】B

【解析】C 语言中,源程序文件的后缀为“.c”,经过编译生成后缀为“.obj”的目标文件,再经过与 C 语言提供的各种库函数连接,生成后缀为“.exe”的可执行文件。故本题答案为 B。

(4) 以下叙述中正确的是( )。(2007 年 9 月)

- A. C 语言程序将从源程序中第一个函数开始执行  
B. 可以在程序中由用户指定任意一个函数作为主函数,程序将从此开始执行  
C. C 语言规定必须用 main 作为主函数名,程序将从此开始执行,在此结束  
D. main 可作为用户标识符,用以命名任意一个函数作为主函数

【答案】C

【解析】main 是主函数的函数名,表示这是一个主函数。每一个 C 源程序都必须有且只能有一个主函数。程序要从 main 函数开始执行,最后在 main 函数中结束。本题选 C。

## 巩固练习

(1) 以下叙述中正确的是( )。

- A. 简单 C 语句必须以分号结束  
B. C 程序中的每一行只能写一条语句  
C. C 语言程序中的注释必须与语句写在同一行  
D. C 语句必须在一行内写完

(2) 下列叙述中错误的是( )。

- A. C 程序可以由多个程序文件组成  
B. 一个 C 语言程序只能实现一种算法  
C. C 程序可以由一个或多个函数组成  
D. 一个 C 函数可以单独作为一个 C 程序文件存在

(3) C 语言源程序名的后缀是( )。

- A. .obj                      B. .exe                      C. .C                      D. .cp

(4) 我们所写的每条 C 语句,经过编译最终都将转换成二进制的机器指令。关于转换以下说法错误的是( )。

- A. 一条 C 语句对应转换成一条机器指令  
B. 一条 C 语句可能会被转换成多条机器指令  
C. 一条 C 语句可能会被转换成零条机器指令  
D. 某种类型和格式的 C 语句被转换成机器指令的条数是固定的

(5) 关于算法,以下叙述中错误的是( )。

- A. 任何算法都能转换成计算机高级语言的程序,并在有限时间内运行完毕
- B. 同一个算法对于相同的输入必能得出相同的结果
- C. 一个算法对于某个输入的循环次数是可以事先估计出来的
- D. 某个算法可能会没有输入

(6) 以下叙述中正确的是( )。

- A. 在C程序中,模块化主要是通过函数来实现的
- B. 程序的主函数名除 main 外,也可以使用 Main 或 \_main
- C. 程序可以包含多个主函数,但总是从第一个主函数处开始执行
- D. 书写源程序时,必须注意缩进格式,否则程序会有编译错误



# 第2章 数据类型、运算符与表达式

本章讲述 C 语言的基本数据类型、运算符和表达式。数据是语言程序处理的对象,如何编写出好的 C 语言程序和如何在程序中描述数据,是学好 C 语言的关键。下面先讲解 C 语言的数据类型。

简单地说,数据类型就是程序给其使用的数据指定某种数据组织形式。例如,可以使用数组来组织一系列的数,那么数组就是一种数据类型。数据类型是按被说明数据的性质、表示形式、占据存储空间多少、构造特点来划分的。在 C 语言中,数据类型可分为基本数据类型、构造数据类型、指针类型和空类型四大类,如表2-1所示。

表 2-1 数据类型

| 数据类型   | 包含的子类型             |
|--------|--------------------|
| 基本数据类型 | 整型、实型(又称浮点型)、字符型   |
| 构造数据类型 | 数组类型、结构类型、共用类型和枚举型 |
| 指针类型   |                    |
| 空类型    |                    |

本章将重点介绍整型(int)、浮点型(float)和字符型(char)这 3 种基本数据类型,并且详细讲解这些数据类型对应的常量和变量。接着介绍 C 语言的基本运算符和表达式。学习完这部分内容之后,读者应该掌握算术运算符、赋值运算符和逗号运算符的使用,并且要特别注意这些运算符的优先级和结合性。另外要结合基本的数据类型来掌握算术表达式、赋值表达式和逗号表达式的定义和使用。除此之外,还将介绍自动转换和强制转换两种类型转换方式。指针类型和构造类型分别在第 8 章和第 9 章介绍。

本章所讲解的知识是进行程序设计的基础,希望读者能够切实地掌握这些知识。

## 2.1 标识符

### 2.1.1 标识符的概念

简单地说,标识符就是一个名字。它用来标识变量的名称、常量的名称、函数的名称以及文件的名称等。构成标识符的字符有大小写字母、数字(0~9)和下划线“\_”,标识符的第一个字符只能是下划线或大小写字母。C 语言是一种大小写敏感的语言,所以 abc、aBc、Abc 是 3 种不同的标识符。不同的 C 语言编译系统对标识符的长度(即一个标识符的最大字符个数)规定可能会有所不同,但是一般是前 32 个字符为有效字符,多余的将