



全国1001所高校学子的明智选择

全国计算机等级考试 全能教程

二级C语言

全国计算机等级考试命题研究组◎编



赠送模拟光盘1张

- **权威性:**
命题研究组资深专家最新研究成果, 作者长期工作在教学一线。
- **全真性:**
所有试题均紧扣最新大纲的要求, 布局均与考试真题完全相当。
- **实战性:**
配书光盘提供现场做题环境, 便于考生在考前60天实战冲刺。
- **前瞻性:**
在深入研究历年真题的基础上, 提供严密的知识体例供学习及应试。



北京邮电大学出版社
www.buptpress.com

全国计算机等级考试全能教程

——二级 C 语言

全国计算机等级考试命题研究组 编

北京邮电大学出版社

·北京·

内 容 简 介

本书以教育部考试中心最新考试大纲和指定教程为依据,在研究历次真题(最新最全)的基础上编写而成。本书章节安排与官方教程一致,主要内容包括:程序设计的基本概念、C 程序设计的初步知识、顺序结构、选择结构、循环结构、字符型数据、函数、指针、数组、字符串、C 语言中用户标识符的作用域和存储类型、编译预处理和动态存储分配、结构体及共用体、位运算和文件等,每章末尾设置有“应试加油站”,该板块汇集重点提示、解题技巧等部分,引导考生掌握重点内容,增强考生的解题能力和综合应用能力。在正文中提供一章进行上机专题辅导。本书附有 3 套笔试模拟试卷和 3 套上机模拟试卷,供考生考前实战演练。

本书配有上机盘。盘中含有数套上机真题,上机题的考试环境、过程和真实考试完全相同。

本书以全国计算机等级考试考生为主要读者对象,适合于考生在等考前复习使用,也可作为相关考试培训班的辅助教材,以及大、中专院校师生的参考书。

图书在版编目(CIP)数据

全国计算机等级考试全能教程. 二级 C 语言/全国计算机等级考试命题研究组编. --北京:北京邮电大学出版社,2011.6
ISBN 978-7-5635-2612-3

I. ①全… II. ①全… III. ①电子计算机—水平考试—教材②C 语言—程序设计—水平考试—教材 IV. ①TP3

中国版本图书馆 CIP 数据核字(2011)第 068446 号

书 名: 全国计算机等级考试全能教程——二级 C 语言

作 者: 全国计算机等级考试命题研究组

责任编辑: 满志文

出版发行: 北京邮电大学出版社

社 址: 北京市海淀区西土城路 10 号(邮编:100876)

发 行 部: 电话: 010-62282185 传真: 010-62283578

E-mail: publish@bupt.edu.cn

经 销: 各地新华书店

印 刷: 北京忠信诚胶印厂

开 本: 889 mm×1 194 mm 1/16

印 张: 18.25

字 数: 617 千字

版 次: 2011 年 6 月第 1 版 2011 年 6 月第 1 次印刷

ISBN 978-7-5635-2612-3

定 价: 34.80 元

• 如有印装质量问题,请与北京邮电大学出版社发行部联系 •

前 言

全国计算机等级考试为国内影响最大、参加人数最多的计算机类水平考试,在推广、普及计算机应用知识和技术中发挥了重要作用,并为用人单位的人员考核提供了客观、公正的评价标准。

为了引导考生顺利通过全国计算机等级考试,我们根据新大纲的要求,结合典型试题,按教育部考试中心指定教材的篇章结构,由从事全国计算机等级考试试题研究人员及在等级考试第一线从事命题研究、教学、辅导和培训的老师精心编写了《全国计算机等级考试全能教程——二级 C 语言》。

1. 本书特色

☑ **突出实用性和高效性:**书的章名、节名与教育部考试中心指定教程同步,每章还设计了以下板块。

- **考题链接:**精选出常考题型与历年真题穿插在知识点中的讲解,一方面有利于考生对知识点的理解,另一方面也让考生明白试题是如何考的。
- **应试加油站:**该板块汇集考试重点整理、解题技巧等部分,引导考生掌握重点内容,增强考生的解题能力和综合应用能力。
- **习题:**针对本节知识点设计一部分题目,方便读者一点一练,巩固提高。

☑ **突出标准性与严谨性:**本书由从事全国计算机等级考试试题研究人员及在等级考试第一线从事命题研究、教学、辅导和培训的老师分工编写,层次清晰,结构严谨,导向准确。

☑ **结构科学,实用性强:**紧扣新大纲要求,精讲考点、重点与难点,深入分析典型范例,抓住等级考试题眼,并提供实战训练。

☑ **注重上机考试的辅导:**针对上机考试的特点,本书在深入研究上机真题库的基础上,将上机真题进行分类,提炼出题型,按类型进行解析,便于考生专项攻克,提高复习效率。

☑ **全真模拟,实战提高:**根据新大纲、新考点、新题型进行最新命题,书末提供 3 套笔试与 3 套上机全真模拟题,供考生考前实战演练。

☑ **书盘结合,一本速通:**光盘中包括考试模拟系统,提供数套上机真题供考生练习,上机题的考试环境、过程和真实考试完全相同,以帮助考生顺利过关。

2. 读者对象

本书以全国计算机等级考试考生为主要读者对象,适合于考生在等考前复习使用,也可作为相关考试培训班的辅助教材,以及大、中专院校师生的参考书。

3. 本书作者

本书由时道波、尹静、阮凌主编,参与本书编写、校对的还有张凌云、刘思平、钱阳勇、范荣钢、陈芳、许勇、许娟、何光明、江梅、赵传申、赵明、陈海燕、丁婷等。

由于作者水平有限,书中难免存在疏漏和错误之处,恳请专家和广大读者批评指正。在学习过程中,遇到疑难问题,可以通过以下方式与我们联系:bjbaba@263.net。

编 者

全国计算机等级考试全能教程丛书

顾问委员会

成员名单 (排名不分先后)

陈 畅 陈海燕 迟冬祥 邓达平 丁为民
江家宝 焦风杰 李 海 刘家琪 卢振侠
骆 健 盛 可 史春联 史国川 孙 虹
唐瑞华 王 钢 王继水 王景胜 吴 婷
吴成林 吴晓维 谢书玉 杨 晋 杨张静
尹 静 应艳杰 张 博 张 剑 张居晓
赵 明 钟志水 谭 红 林 莉 徐文娟
王 强

本书主编 时道波 尹 静 阮 凌

目 录

第 1 章 程序设计的基本概念	1	算符(一)	19
1.1 程序和程序设计	1	2.7.2 逗号运算符和逗号表达式	19
1.2 算法	2	2.8 应试加油站	20
1.3 结构化程序设计和模块化结构	3	2.8.1 考试重点整理	20
1.3.1 结构化程序	3	2.8.2 解题技巧	20
1.3.2 模块化结构	4	2.9 过关练习与答案	22
1.4 应试加油站	5	2.9.1 过关练习	22
1.4.1 考试重点整理	5	2.9.2 参考答案	23
1.4.2 解题技巧	5	第 3 章 顺序结构	24
1.5 过关练习与答案	6	3.1 赋值语句	24
1.5.1 过关练习	6	3.2 数据输出	24
1.5.2 参考答案	6	3.2.1 printf 函数的一般调用形式	24
第 2 章 C 程序设计的初步知识	7	3.2.2 printf 函数中常用的格式说明	24
2.1 简单 C 语言程序的构成和格式	7	3.2.3 调用 printf 函数时的考试重点整理	27
2.2 标识符、常量和变量	9	3.3 数据输入	28
2.2.1 标识符	9	3.3.1 scanf 函数的一般调用格式	28
2.2.2 常量	10	3.3.2 scanf 函数中常用的格式说明	29
2.2.3 变量	11	3.3.3 通过 scanf 函数从键盘输入数据	29
2.3 整型数据	11	3.4 复合语句和空语句	30
2.3.1 整型常量	11	3.4.1 复合语句	30
2.3.2 整型变量	12	3.4.2 空语句	31
2.3.3 整型数据的分类	12	3.5 程序举例	31
2.3.4 整数在内存中的存储形式	13	3.6 应试加油站	33
2.4 实型数据	13	3.6.1 考试重点整理	33
2.4.1 实型常量	13	3.6.2 解题技巧	33
2.4.2 实型变量	14	3.7 过关练习与答案	34
2.5 算术表达式	15	3.7.1 过关练习	34
2.5.1 基本的算术运算符	15	3.7.2 参考答案	36
2.5.2 运算符的优先级、结合性和算术表达式	15	第 4 章 选择结构	37
2.5.3 强制类型转换表达式	16	4.1 关系运算和逻辑运算	37
2.6 赋值表达式	17	4.1.1 关系运算符和关系表达式	37
2.6.1 赋值运算符和赋值表达式	17	4.1.2 逻辑运算符和逻辑表达式	38
2.6.2 复合的赋值表达式	18	4.2 if 语句和嵌套 if 语句	40
2.6.3 赋值运算中的类型转换	18	4.2.1 if 语句	40
2.7 自增、自减运算符和逗号运算符	19	4.2.2 嵌套 if 语句	42
2.7.1 自增运算符(++)和自减运		4.3 条件运算符和条件表达	45
		4.4 switch 语句	46



4.5 语句标号和 goto 语句	51	6.5 应试加油站	84
4.5.1 语句标号	51	6.5.1 考试重点整理	84
4.5.2 goto 语句	51	6.5.2 解题技巧	84
4.6 应试加油站	51	6.6 过关练习与答案	85
4.6.1 考试重点整理	51	6.6.1 过关练习	85
4.6.2 解题技巧	51	6.6.2 参考答案	86
4.7 过关练习与答案	54	第7章 函数	87
4.7.1 过关练习	54	7.1 库函数	87
4.7.2 参考答案	55	7.2 函数的定义和返回值	88
第5章 循环结构	56	7.2.1 函数的定义	88
5.1 while 语句构成的循环结构	56	7.2.2 函数的返回值	89
5.1.1 while 循环的一般形式	56	7.3 函数的调用	90
5.1.2 while 循环的执行过程	56	7.3.1 函数的两种调用方式	90
5.2 do-while 语句构成的循环结构	59	7.3.2 函数调用时的语法要求	91
5.2.1 do-while 语句构成的循环结构	59	7.4 函数的说明	93
5.2.2 do-while 循环的执行过程	59	7.4.1 函数说明的形式	93
5.3 for 语句构成的循环结构	61	7.4.2 函数说明的位置	94
5.3.1 for 语句构成的循环结构	61	7.5 函数间的数据传递	95
5.3.2 for 循环的执行过程	61	7.6 程序举例	98
5.3.3 有关 for 语句的说明	61	7.7 应试加油站	100
5.4 循环结构的嵌套	64	7.7.1 考试重点整理	100
5.5 break 和 continue 语句的作用	68	7.7.2 解题技巧	100
5.5.1 break 语句	68	7.8 过关练习与答案	102
5.5.2 continue 语句	69	7.8.1 过关练习	102
5.6 程序举例	69	7.8.2 参考答案	103
5.7 应试加油站	72	第8章 指针	104
5.7.1 考试重点整理	72	8.1 变量的地址和指针	104
5.7.2 解题技巧	72	8.2 指针变量的定义和指针变量的基类型	105
5.8 过关练习与答案	75	8.3 给指针变量赋值	106
5.8.1 过关练习	75	8.3.1 给指针变量赋地址值	106
5.8.2 参考答案	77	8.3.2 给指针变量赋“空”值	106
第6章 字符型数据	78	8.4 对指针变量的操作	107
6.1 字符型常量	78	8.4.1 通过指针来引用一个存储单元	107
6.1.1 字符常量	78	8.4.2 移动指针	109
6.1.2 转义字符常量	78	8.4.3 指针比较	110
6.1.3 字符串常量	79	8.5 函数之间地址值的传递	110
6.1.4 对字符常量进行的运算	79	8.6 应试加油站	113
6.2 字符变量	80	8.6.1 考试重点整理	113
6.3 字符的输入和输出	81	8.6.2 解题技巧	114
6.3.1 调用 printf 和 scanf 函数输出和输入	81	8.7 过关练习与答案	116
字符	81	8.7.1 过关练习	116
6.3.2 调用 putchar 和 getchar 函数输出和输入	81	8.7.2 参考答案	117
字符	81	第9章 数组	118
6.4 程序举例	83	9.1 一维数组的定义和一维数组元素的引用	118



9.1.1 一维数组的定义	118	10.2 使指针指向一个字符串	150
9.1.2 一维数组元素的引用	119	10.3 字符串的输入和输出	151
9.1.3 一维数组的初始化	119	10.3.1 字符串的输入	152
9.1.4 通过赋初值定义数组的大小	119	10.3.2 字符串的输出	153
9.1.5 一维数组的定义和数组元素引用举例 ..	120	10.4 字符串数组	153
9.2 一维数组和指针	121	10.5 用于字符串处理的函数	154
9.2.1 一维数组和数组元素的地址	121	10.6 字符数组和字符串应用举例	158
9.2.2 通过数组的首地址引用数组元素	122	10.7 应试加油站	159
9.2.3 通过指针及带下标的指针变量引用一 维数组元素	122	10.7.1 考试重点整理	159
9.2.4 指针变量的运算	123	10.7.2 解题技巧	159
9.3 函数之间对一维数组和数组元素的引用	125	10.8 过关练习与答案	163
9.3.1 数组名作为实参	125	10.8.1 过关练习	163
9.3.2 数组元素地址作为实参	126	10.8.2 参考答案	165
9.3.3 函数的指针形参和函数体中数组的 区别	127	第 11 章 对函数的进一步讨论	166
9.4 一维数组应用举例	128	11.1 传给 main 函数的参数	166
9.5 二维数组的定义和二维数组元素的引用	134	11.2 通过实参向函数传递函数名或指向函数的 指针变量	167
9.5.1 二维数组的定义	134	11.3 函数的递归调用	168
9.5.2 二维数组元素的引用	135	11.4 函数的嵌套调用	172
9.5.3 二维数组的初始化	135	11.5 应试加油站	173
9.5.4 通过赋初值定义二维数组的大小	135	11.5.1 考试重点整理	173
9.6 二维数组和指针	136	11.5.2 解题技巧	173
9.6.1 二维数组和数组元素的地址	136	11.6 过关练习与答案	174
9.6.2 通过地址来引用二维数组元素	137	11.6.1 过关练习	174
9.6.3 通过建立一个指针数组来引用二维数组 元素	137	11.6.2 参考答案	176
9.6.4 通过建立一个行指针来引用二维数组 元素	138	第 12 章 C 语言中用户标识符的作用域和存储类	177
9.7 二维数组名和指针数组作为实参	139	12.1 局部变量和全局变量	177
9.7.1 二维数组名作为实参	139	12.2 局部变量的作用域和生存期	179
9.7.2 指针数组作为实参	139	12.2.1 auto 变量	179
9.8 二维数组应用举例	140	12.2.2 register 变量	179
9.9 应试加油站	142	12.2.3 静态存储类的局部变量	180
9.9.1 考试重点整理	142	12.3 全局变量的作用域和生存期	181
9.9.2 解题技巧	142	12.3.1 在同一编译单位内扩展全局变量的 作用域	182
9.10 过关练习与答案	146	12.3.2 在不同编译单位内扩展全局变量的 作用域	182
9.10.1 过关练习	146	12.3.3 静态全局变量	183
9.10.2 参考答案	147	12.4 函数的存储分类	184
第 10 章 字符串	148	12.4.1 利用 extern 说明函数	184
10.1 利用一维字符数组存放字符串	148	12.4.2 利用 static 说明函数	184
10.1.1 字符串的相关概念	148	12.5 应试加油站	184
10.1.2 通过赋初值的方式给一维字符数组赋 字符串	148	12.5.1 考试重点整理	184
10.1.3 在赋初值时直接赋字符串常量	149	12.5.2 解题技巧	184
		12.6 过关练习与答案	186



12.6.1 过关练习	186	15.4.1 过关练习	229
12.6.2 参考答案	188	15.4.2 参考答案	230
第 13 章 编译预处理和动态存储分配	189	第 16 章 文件	231
13.1 编译预处理	189	16.1 C 语言文件的概念	231
13.1.1 宏替换	189	16.2 文件指针	232
13.1.2 文件包含	192	16.3 fopen 函数	232
13.2 动态存储分配	193	16.4 fclose 函数	234
13.2.1 malloc 函数和 free 函数	193	16.5 fgetc 和 fputc 函数	235
13.2.2 calloc 函数	194	16.6 feof 函数	237
13.3 应试加油站	195	16.7 fscanf 和 fprintf 函数	237
13.3.1 考试重点整理	195	16.8 fgets 和 fputs 函数	238
13.3.2 解题技巧	195	16.9 fread 和 fwrite 函数	238
13.4 过关练习与答案	196	16.10 文件定位函数	240
13.4.1 过关练习	196	16.10.1 fseek 函数	240
13.4.2 参考答案	197	16.10.2 ftell 函数	240
第 14 章 结构体、共用体和用户自定义类型	198	16.10.3 rewind 函数	241
14.1 利用 typedef 说明一种新类型名	198	16.11 应试加油站	241
14.2 结构体类型	199	16.11.1 考试重点整理	241
14.2.1 结构体类型的说明	199	16.11.2 解题技巧	241
14.2.2 结构体类型的变量定义	200	16.12 过关练习与答案	243
14.2.3 结构体类型变量的引用	202	16.12.1 过关练习	243
14.2.4 结构体类型变量的赋值和初始化	202	16.12.2 参考答案	245
14.2.5 结构体类型数组的定义和赋值	203	第 17 章 上机指导	246
14.2.6 结构体指针变量	206	17.1 考试重点整理	246
14.2.7 通过结构体变量构成链表	211	17.2 常考知识点指导	246
14.3 共用体	215	17.3 过关练习与答案	258
14.3.1 共用体类型的说明和变量定义	215	17.3.1 过关练习	258
14.3.2 共用体变量的引用	217	17.3.2 参考答案	260
14.4 应试加油站	218	第 18 章 模拟试题与解析	261
14.4.1 考试重点整理	218	18.1 笔试全真试题	261
14.4.2 解题技巧	218	18.1.1 笔试题题一	261
14.5 过关练习与答案	221	18.1.2 笔试题题二	266
14.5.1 过关练习	221	18.1.3 笔试题题一参考答案与解析	272
14.5.2 参考答案	224	18.1.4 笔试题题二参考答案与解析	275
第 15 章 位运算	225	18.2 上机全真模拟试题	278
15.1 位运算符	225	18.2.1 上机全真模拟试题一	278
15.2 位运算符的运算功能	226	18.2.2 上机全真模拟试题二	279
15.3 应试加油站	229	18.2.3 上机全真模拟试题三	280
15.3.1 考试重点整理	229	18.2.4 上机全真模拟试题一参考答案及解析	281
15.3.2 解题技巧	229	18.2.5 上机全真模拟试题二参考答案及解析	281
15.4 过关练习与答案	229	18.2.6 上机全真模拟试题三参考答案及解析	282

第 1 章 程序设计的基本概念

1.1 程序和程序设计

程序是指存储在计算机内部存储器中,可以连续执行的一条条指令的集合。它是人与机器进行“对话”的语言,也就是人们常说的“程序设计语言”。C语言是一种“高级语言”,人们把由高级语言编写的程序称为“源程序”,由二进制代码表示的程序称为“目标程序”,但对于计算机本身来说,它并不能直接识别由高级语言编写的程序,只能接受和处理由0和1的代码构成的二进制指令或数据。由于这种形式的指令是面向机器的,因此也称为“机器语言”。而“编译程序”是指可以把源程序转换成目标程序的软件,即可以把人们按规定语法写出的语句逐一翻译成二进制的机器指令。每一种高级语言都有它对应的编译程序。每条C语句经过编译(Compile)最终都将转换成二进制的机器指令。由C语言构成的指令序列称为C语言源程序;按C语言的语法编写C程序的过程,称为C语言的代码编写。C语言源程序经过编译后生成一个扩展名为.obj的二进制文件(目标文件),最后还要由“连接程序”(Link)将此.obj文件与C语言中提供的各种库函数连接起来生成一个扩展名为.exe的可执行文件。在DOS状态下,只需输入此文件的名字(不必输入扩展名.exe)就可以运行。

简单的程序设计一般包括以下几个部分:

- (1) 根据任务书提出的要求、指定的输入数据和输出的结果,确定存放数据的数据结构;
- (2) 针对存放数据的数据结构来确定解决问题、完成任务的每一个步骤,即确定算法;
- (3) 根据确定的数据结构和算法,使用选定的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,即编程;
- (4) 在计算机上调试程序,消除由于疏忽而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出正确的结果,对不合理的数据能进行适当的处理;
- (5) 整理并写出文档资料。

考题链接

【例题 1-1】计算机能直接执行的程序是_____。(2008.09)

- A) 源程序 B) 目标程序 C) 汇编程序 D) 可执行程序

解 析: 源程序是由程序设计语言构成的指令序列,需要由特定的软件去执行。将源程序编译后可得到目标程序,计算机仍然不能直接执行。而在操作系统下,只须单击或输入可执行文件的名称,该可执行文件就可运行。

答 案: D



【例题 1-2】C 语言程序名的扩展名是_____。(2007.09)

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

解析：C 语言程序名的扩展名是.c。C 语言源编译程序经过编译后形成一个扩展名为.obj 的二进制文件。

答案：B

【例题 1-3】以下叙述正确的是_____。(2009.09)

- A) 程序设计的任务就是编写程序代码并上机调试
 B) 程序设计的任务就是确定所用数据结构
 C) 程序设计的任务就是确定所用算法
 D) 以上三种说法都不完整

解析：计算机程序设计的步骤有：确定数据结构、确定算法、编程、调试程序、整理书写文档资料，所以 A、B、C 三个选项都不全面。

答案：D

1.2 算 法

算法是为了解决某个特定问题而采取的确定的且有限的步骤。一个算法应当具有以下 5 个特性：

- 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。
- 确定性。算法中每条指令必须有明确的含义，不能有二义性。对于相同的输入必能得出相同的执行结果。
- 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。
- 有零个或多个输入。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。
- 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。

算法一般可以用以下 3 种方法来进行描述：

(1) 伪代码：一种近似高级语言但又不受语法约束的语言描述方式，这在英语国家中使用起来更为方便。

(2) 流程图：是描述算法中很好的工具。传统的流程图由如图 1-1 所示的几种基本框架组成。



图 1-1 流程图基本框架

用这些框和流程线组成的流程图来表示算法，形象直观，简单方便。但这种流程图对于流程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，既费时又费力，而且还不易阅读。

(3) N-S 流程图：随着结构化程序设计方法的出现，1973 年美国学者 I. Nassi 和 B. Shneiderman 提出了一种新的流程图形式。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。这种流程图用两位学者名字的第一个英文字母命名，称为 N-S 流程图。

一般情况下，最常用的描述方法是伪代码和流程图。

考题链接

【例题 1-4】程序流程图中的菱形框表示的是_____。(2009.09)

解 析：程序流程图中的菱形框是判断框，表示的是逻辑条件。

答 案：逻辑条件

【例题 1-5】算法中对需要执行的每一步操作，必须给出清楚、严格的规定，这属于算法的_____。
(2007.04)

A) 正当性 B) 可行性 C) 确定性 D) 有穷性

解 析：算法应该具有下列 5 个特性：有穷性、确定性、可行性、有零个或多个输入、有一个或多个输出。算法的确定性是指：算法中每一条指令必须有确切的含义，读者理解时不会产生二义性，并且在任何条件下，算法只有唯一的一条执行路径，即对于相同的输入只能得出相同的执行结果。

答 案：C

1.3 结构化程序设计和模块化结构

1.3.1 结构化程序

结构化程序由 3 种基本结构组成，即顺序结构、选择结构、循环结构。

1. 顺序结构

如赋值语句、输入、输出语句等都可以构成顺序结构。当执行这些语句构成的程序时，将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。顺序结构可用如图 1-2 所示的流程图来表示，(a)是一般的流程图，(b)是 N-S 流程图。

2. 选择结构

选择结构的程序要根据不同的条件去执行不同分支中的语句，如 if 语句、switch 语句等都可以构成选择结构。选择结构可用如图 1-3 所示的流程图表示，(a)是一般的流程图，(b)是 N-S 流程图。

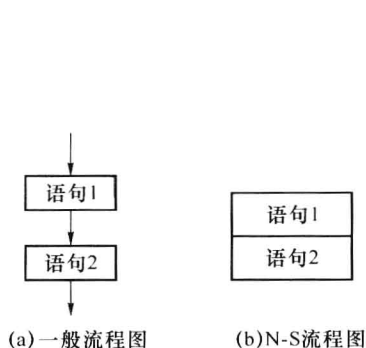


图 1-2 顺序结构流程图

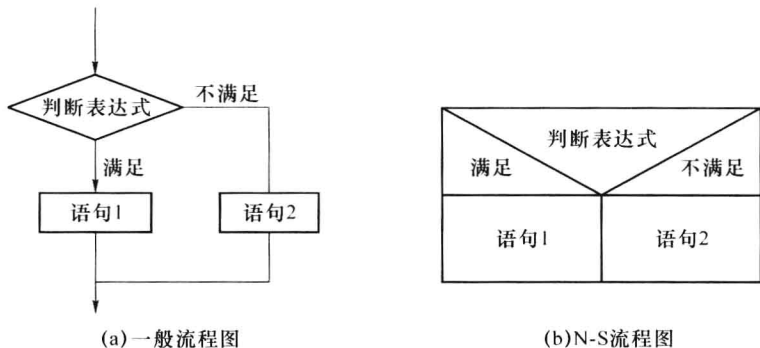


图 1-3 选择结构流程图

3. 循环结构

不同形式的循环结构要根据各自的条件，使同一组语句重复执行多次或一次也不执行。循环结构包括当型循环(如图 1-4 所示)和直到型循环(如图 1-5 所示)。当型循环的特点是：当指定的条件满足时，就

执行循环体,否则就不执行。直到型循环的特点是:执行循环体直到指定的条件满足,就不再执行循环。

事实已经证明,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种基本结构组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。

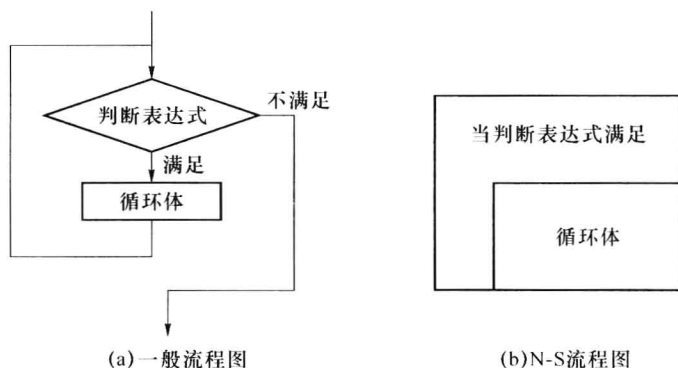


图 1-4 当型循环

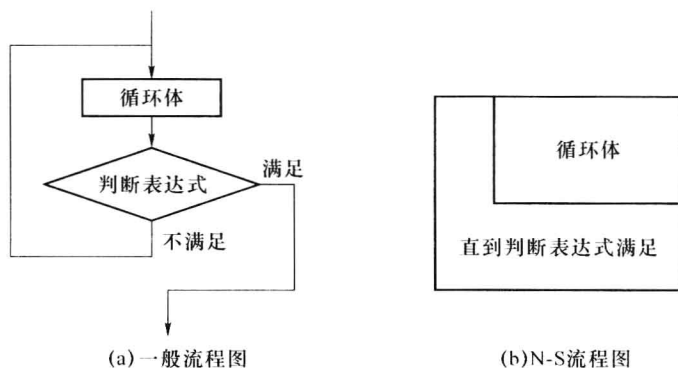


图 1-5 直到型循环

1.3.2 模块化结构

计算机在处理复杂任务时,所编写的应用程序经常由上万条语句组成,需要由许多人来完成。这时,常常要把一个大任务分解为若干个子任务,每个子任务又分成很多个小子任务,每个小子任务只完成一项简单的功能。在程序设计时,用一个个小模块来实现这些功能。程序设计人员分别完成一个或多个小模块,人们称这样的程序设计方法为“模块化”,由一个个功能模块构成的程序结构就称为模块化结构。模块化结构可以大大提高程序编写的效率。

软件人员在进行程序设计时,首先应集中考虑主程序中的算法,写出主程序后再动手逐步完成子程序的调用。而对这些子程序也可以利用主程序的方法逐步完成其下一层子程序的调用。这就是自顶向下、逐步细化、模块化的程序设计。

C语言是一种结构化程序设计语言。它直接提供了3种基本结构的语句,提供了定义“函数”的功能,在C语言中没有子程序的概念,它提供的函数可以完成子程序的所有功能;C语言中允许对函数单独进行编译,从而可以实现模块化;另外还提供了丰富的数据类型。这些都为结构化程序设计提供了有力的工具。



1.4 应试加油站

1.4.1 考试重点整理

- 高级语言编写的程序称为“源程序”，二进制代码表示的程序称为“目标程序”，而“编译程序”是指可以把源程序转换成目标程序的软件。每条C语句经过编译(Compile)最终都将转换成二进制的机器指令(.obj的文件)。最后由“连接程序”(Link)将此.obj文件与C语言提供的各种库函数连接起来生成一个扩展名为.exe的可执行文件。
- 算法的5个特性分别是：有穷性、确定性、可行性、有零个或多个输入以及有一个或多个输出。
- 结构化程序由3种基本结构组成：顺序结构、选择结构和循环结构。

1.4.2 解题技巧

【典型题 1-1】下列叙述中错误的是_____。

- A) 计算机不能直接执行用C语言编写的源程序
- B) C程序经编译后,生成的扩展名为.obj的文件是一个二进制文件
- C) 扩展名为.obj的文件,经连接程序生成扩展名为.exe的文件是一个二进制文件
- D) 扩展名为.obj和.exe的二进制文件都可以直接运行

解析：一个C语言的源程序(扩展名为.c)在经过编译器编译后,首先生成一个汇编语言程序,然后由编译程序再将汇编语言程序翻译成机器指令程序,即目标程序(扩展名为.obj)。目标程序不可以直接运行,它要和库函数或其他目标程序连接成可执行文件(扩展名为.exe)后方可运行,故本题答案为D。

答案：D

【典型题 1-2】以下叙述中错误的是_____。

- A) 算法正确的程序最终一定会结束
- B) 算法正确的程序可以有零个输出
- C) 算法正确的程序可以有零个输入
- D) 算法正确的程序对于相同的输入一定有相同的结果

解析：根据算法的5个特性可知,一个有效的算法程序必须有一个或一个以上的输出,所以选项B错误。

答案：B

【典型题 1-3】以下叙述中错误的是_____。

- A) C语言是一种结构化程序设计语言
- B) 结构化程序由顺序、分支、循环3种基本结构组成
- C) 使用3种基本结构构成的程序只能解决简单问题
- D) 结构化程序设计提倡模块化的设计方法

解析：结构化程序设计是指在程序的构成上只使用顺序、选择(即分支)和循环3种结构组成的编程方式。它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。结构化程序设计提倡模块化的设计方法。

答案：C

【典型题 1-4】能将高级语言编写的源程序转换为目标程序的软件是_____。

- A) 汇编程序
- B) 编辑程序
- C) 解释程序
- D) 编译程序

解析：对于计算机来说,它并不能直接识别由高级语言编写的程序,它只能接受和处理由0和1的代码构成的二进制指令或数据。我们把由高级语言编写的程序称为“源程序”,把由二进制代码表示的程序称为“目标程序”。如何把源程序转换成机器能够接受的目标程序,软件工作者编制了一系列软件,通过这些软件可以把用户按规定语法写出的语句翻译成二进制的机器指令,这种具有翻译功能的软件称为“编译程序”。一般每一种高级语言都有与它对应的编译程序。

答案：D

【典型题 1-5】一个算法应该具有“确定性”等5个特性,下面对另外4个特性的描述中错误的是_____。

- A) 有零个或多个输入
- B) 有零个或多个输出
- C) 有穷性
- D) 可行性

解析：算法的5个特性为：有穷性、确定性、可行性、有零个或多个输入、有1个或多个输出。

答案：B

【典型题 1-6】C语言中用于结构化程序设计的三种基本结构是_____。

- A) 顺序结构、选择结构、循环结构
- B) if、switch、break
- C) for、while、do-while
- D) if、for、continue

解析：结构化程序由三种基本结构组成：顺序结构、选择结构和循环结构。在选择结构中,又分为if结构和switch结构,在循环结构中,又分为while型和do-while型和for型循环。

答案：A



1.5 过关练习与答案

1.5.1 过关练习

1. 能将高级语言编写的源程序转换成目标程序的是_____。
A) 编辑程序 B) 编译程序
C) 驱动程序 D) 连接程序
2. 下列 4 条叙述中,正确的一条是_____。
A) 计算机语言中,只有机器语言属于低级语言
B) 高级语言源程序可以被计算机直接执行
C) C 语言属于高级语言
D) 机器语言是与所用机器无关的
3. 算法具有 5 个特性,以下选项中不属于算法特性的是_____。
A) 有零个或多个输入 B) 可行性
C) 有穷性 D) 通用性
4. while、do-while 语句是用于_____基本结构。

- A) 顺序 B) 选择
C) 循环 D) 转移
5. 用 C 语言编写的代码_____。
A) 可立即执行 B) 是一个源程序
C) 经过编译即可执行 D) 经过编译解释才能执行
6. C 语言中的赋值、输入输出语句可以构成_____基本结构。
A) 分支 B) 顺序
C) 循环 D) 选择
7. 在循环结构中,_____可以使得同一组语句一次也不执行。
A) 当型循环 B) 直到型循环
C) 都不能实现 D) do-while 语句

1.5.2 参考答案

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

第 2 章 . C 程序设计的初步知识

2.1 简单 C 语言程序的构成和格式

下面是一个简单的 C 程序。

【例】比较两个数的大小。

```
main ()                                /* 主函数,功能为输入两个整数,输出其中较大的数 */
{
    int a,b,c;                          /* 定义 3 个整型变量 */
    scanf("%d, %d", &a, &b);           /* 输入变量 a 和 b 的值 */
    c = max(a, b);                       /* 调用 max 函数,将得到的值赋给 c */
    printf("max = %d", c);              /* 输出 c 的值 */
}

/* 空一行,使程序清晰可读 */
int max(x,y)                             /* 定义 max 函数,函数值为整数,x,y 为形式参数 */
{   int x, y;                             /* 定义形式参数 x,y 为整型变量 */
    int z;                                 /* 定义 max 函数内部用到的变量 z 为整型变量 */
    if (x>y) z = x;                       /* 将 x,y 中较大的值赋给 z */
    else z = y;
    return (z);                           /* 将 z 值返回,通过 max 函数带回调用处 */
}
```

本程序的作用是从键盘输入两个整数,然后在屏幕上输出它们的最大值。

该程序包括两个函数:一个主函数 main 和一个自定义函数 max。max 函数的作用是将 x 和 y 中较大的数赋给变量 z,再通过 return 语句将 z 的值返回给主函数 main。第 4 行中的 scanf 是 C 语言提供的标准输入函数(printf 是 C 语言中提供的标准输出函数),其作用是输入 a 和 b 的值。其中,"%d,%d"用于指定输入的两个数据按十进制整数形式输入。&a 和 &b 中的"&"表示"地址",此形式与其他语言不同,scanf 函数是将两个数值分别输入到变量 a 和 b 地址所标识的单元中来完成 a 和 b 值的输入。有关 scanf 函数的说明详见第 3 章。

程序的第 5 行是调用 max 函数,在调用过程中将实际参数 a 和 b 的值分别传递给 max 函数中的形式参数 x 和 y,然后得到一个返回值(max 函数中变量 z 的值),并把这个值赋给变量 c。第 6 行是输出 c 的值。程序的运行情况如下。

```
4,9<回车>          (输入 4 和 9 给 a 和 b)
max = 9             (输出 c 的值)
```




在以上程序中,可以看到 C 程序有以下结构特征:

(1) main 是主函数名,C 语言中规定必须用 main 作为主函数名。其后的“()”中间可以是空的,但“()”不能省略。一个 C 程序可以包含任意多个不同名的函数,但必须有且只有一个主函数,C 程序总是从主函数开始处执行。

(2) 在函数的起始行后面是函数体,是由一对大括号“{}”括起来的语句集合。函数体中一般包含变量定义语句和执行语句,且执行语句必须放在定义语句之后。

(3) 每个语句和变量定义的最后必须要有一个分号(即使是程序的最后一个语句也应包含分号)。分号是 C 语句的必要组成部分。

(4) C 语言中用“/*...*/”对程序进行注释。注意“/”和“*”之间不能有空格。注释可以出现在程序中任意合适的地方,但必须成对出现。它对程序的运行不起作用,但在注释中可以说明变量的含义、程序段的功能等。一个好的 C 程序应有必要的注释,以便阅读。

(5) “#include “stdio. h””通常称为命令行,必须用“#”号开头,最后不能加“;”,因为它不是 C 程序中的语句。“stdio”是系统提供的文件名,该文件中包含着有关输入输出函数的信息,调用不同的库函数,应当包含不同的头文件。

另外,为了增强程序的可读性,便于人们理解和查错,建议采用以下书写格式:

(1) C 程序习惯上使用小写英文字母,只有符号常量或其他特殊用途的符号才大写。

(2) 不使用行号,通常按语句顺序执行。

(3) C 程序不存在程序行的概念,一个语句可以占多行,一行也可以有多个语句(此时各语句之间要用分号隔开)。

(4) 不指定语句在一行中的起始位置,但建议同一结构层次的语句应左对齐。

(5) 用大括号“{}”表示各结构层次的范围,可以表示函数,也可以表示循环体等。

(6) 一个语句中不同成分之间应使用空格隔开。标识符、关键字之间至少加一个空格隔开。若已有明显的间隔,也可以不再加空格来间隔。

(7) 为了使程序清晰,可以加上空行(如上例中),空行不影响程序的执行,但不要在一个语句之间加空行。

(8) 建议多使用注释信息,以增加可读性。

考题链接

【例题 2-1】 以下叙述中正确的是_____。(2008.09)

- A) C 程序的基本组成单位是语句
- B) C 程序中的每一行只能写一条语句
- C) 简单 C 语句必须以分号结束
- D) C 语言必须在一行内写完

解 析: C 程序的基本组成单位是函数,一个或多个 C 函数组成一个 C 程序。C 程序允许一行写多条语句,也允许一条语句分几行书写,但每条语句必须以分号结束。

答 案: C

【例题 2-2】 以下叙述中正确的是_____。(2008.04)

- 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

解 析: C 程序的注释可以出现在 C 程序的任何位置,注释的内容放在符号“/*”和“*/”之间。C 程序书写格式自由,一行内可写多个语句,每条语句用分号“;”结束。用 C 语言编写的程序可以放在多个程序文件中,用“#include”命令行实现文件的包含功能。

答 案: C