

全国计算机等级考试用书 | 新考纲 |

二级C语言 程序设计

侯东昌
宋智玲 编著
李雪梅

com

uter



中国水利水电出版社
www.waterpub.com.cn

全国计算机等级考试用书[新考纲]

二级C语言程序设计

侯东昌
宋智玲 编著
李雪梅



中国水利水电出版社
www.waterpub.com.cn

内 容 提 要

本书共分为 14 章, 内容包括: 计算机软件基础知识 (包括数据结构、软件设计基础、软件工程及数据库的基础知识), Turbo C 2.0 IDE 环境的应用, 程序设计的基础知识, C 程序语言概述, C 语言的基本数据类型与表达式, 简单的 C 语言程序设计, C 语言中的分支结构, C 语言循环控制语句, 函数与变量类型, 数组, 指针, 结构体与共用体, 文件, 以及 C 语言中的常见错误分析和程序调试等内容。

这套等级考试用书系由在全国计算机等级考试第一线从事教学、辅导、培训的教师及试题研究人员分工编写的, 层次清晰, 导向正确, 针对性强, 可供考生考前实战, 感受全真训练。

本书可作为全国计算机等级考试的参考用书, 也可作为大学本科教学用书, 并可作为大中专、高职高专与各类培训人员的培训教材, 以及相关技术人员的参考用书。

图书在版编目 (CIP) 数据

二级 C 语言程序设计: 新考纲/侯东昌, 宋智玲, 李雪梅编著. —北京: 中国水利水电出版社, 2005
全国计算机等级考试用书
ISBN 7-5084-2633-9
I. 二... II. ①侯...②宋...③李... III. C 语言—程序设计—水平考试—自学参考资料 IV. TP312

中国版本图书馆 CIP 数据核字 (2004) 第 141613 号

书 名	全国计算机等级考试用书 [新考纲] 二级 C 语言程序设计
作 者	侯东昌 宋智玲 李雪梅 编著
出版 发行	中国水利水电出版社 (北京市三里河路 6 号 100044) 网址: www.waterpub.com.cn E-mail: sales@waterpub.com.cn 电话: (010) 63202266 (总机)、68331835 (营销中心)
经 售	全国各地新华书店和相关出版物销售网点
排 版	中国水利水电出版社微机排版中心
印 刷	北京市兴怀印刷厂
规 格	787mm×1092mm 16 开本 21.5 印张 510 千字
版 次	2005 年 1 月第 1 版 2005 年 1 月第 1 次印刷
印 数	0001—5000 册
定 价	30.00 元

凡购买我社图书, 如有缺页、倒页、脱页的, 本社营销中心负责调换

版权所有·侵权必究

丛书编委会名单

主 编 贾小珠

副 主 编 夏方道 宋立智

成 员	茹俊丽	周 伟	王会恩	蔺德军	侯东昌
	李 云	袁 琪	张 倩	宋智玲	王新华
	史英侃	綦伟青	李雪梅	杜祥军	司奇杰
	曹振楠	周广利	王吉茂		

前 言

全国计算机等级考试是目前国内影响最大、参加人数最多的计算机类水平考试，由教育部考试中心主办。自 1994 年开考以来，截止到 2004 年下半年，已顺利考过 20 次，考生累计人数接近 1000 多万。目前，高校大学生的择业、单位职工的职称评定、干部的选拔以及下岗人员的再就业等，都可能需要提供由教育部颁发的计算机等级考试证书。因此，参加全国计算机等级考试不仅成为一个热点，而且已成为许多人的必然需要。

全国计算机等级考试根据计算机应用水平的不同分为三个等级，包括：一级、二级、三级，人们可以根据自己的实际水平参加不同级别的考试。为适应计算机应用技术的飞速发展，使计算机等级考试的科目及内容更加接近目前较为先进的计算机应用技术，教育部考试中心先后几次对计算机等级考试的科目、内容以及考试时间进行了大幅度的调整，并于 2004 年通过了经全国计算机等级考试委员会审定的最新考试大纲。

为了适应新的考试大纲的要求，帮助广大参考人员顺利通过计算机等级考试，并全面提高其计算机应用水平，我们在深入剖析最新考试大纲和历年考题的基础上，编写了本套用书，共 8 册：

- 《一级》
- 《二级 Visual Basic 程序设计》
- 《二级 Visual FoxPro 程序设计》
- 《二级 C 语言程序设计》
- 《三级 PC 技术》
- 《三级网络技术》
- 《三级数据库技术》
- 《三级信息管理技术》

本套按新考纲编写的“全国计算机等级考试用书”具有以下几个特点：

(1) 由在全国计算机等级考试第一线从事教学、辅导和培训的教师以及等级考试试题研究人员分工编写，层次清晰，结构严谨，导向正确。

(2) 紧扣考试大纲的要求，对大纲的各个考点进行仔细的分析，确保全

套书内容准确。

(3) 用言简意赅的语言精讲考试重点和难点,以帮助考生尽快达到考试大纲的基本要求。

(4) 书中含有大量的例题和练习题。其中,例题部分是在对历年考题进行详细分析的基础上给出的,覆盖了全国计算机等级考试的全部要点;练习题则能帮助考生在较短的时间内熟悉考试要点和考试题型。

(5) 在对历年全真考题研究的基础上精心设计的模拟试题可供考生考前实战,感受全真训练。

本套“全国计算机等级考试用书”以对考生进行综合指导为原则,具有极强的针对性,特别适合希望在较短的时间内取得较大收获的广大考生,也可作为各类全国计算机等级考试培训班的教材,以及大、中专院校师生的教学参考用书。

我们相信,本套用书对读者备考和全面掌握计算机应用知识将会有很大的帮助,可收到事半功倍的效果。

尽管我们力求精益求精,但书中的错漏或不妥之处仍然在所难免,敬请读者批评和指正。

编 者

2004 年 11 月

目 录

前 言

第 1 章 计算机软件基础知识

1.1 数据结构与算法	1
1.1.1 算法的基本概念	1
1.1.2 数据结构的定义	2
1.1.3 线性表结构	2
1.1.4 栈与队列结构	3
1.1.5 线性单链表、双向链表与循环链表结构	8
1.1.6 树与二叉树结构	9
1.1.7 顺序查找与二分查找	10
1.2 计算机软件设计基础	11
1.2.1 程序设计方法	11
1.2.2 程序设计风格	12
1.2.3 结构化程序设计	13
1.2.4 面向对象的程序设计方法简介	15
1.3 计算机软件工程基础	17
1.3.1 软件工程概念	17
1.3.2 结构化分析方法	17
1.3.3 结构化设计方法	19
1.3.4 软件测试方法	20
1.4 数据库设计基础	20
1.4.1 数据库的发展阶段	20
1.4.2 数据描述	22
1.4.3 数据模型	24
1.4.4 关系代数运算	28
1.4.5 数据库设计过程	30
1.4.6 数据库设计技巧	33
1.5 本章小结	37
1.6 练习题	37
1.6.1 选择题	37

1.6.2 填空题	37
-----------------	----

第2章 Turbo C 2.0 集成环境简介

2.1 Turbo C 2.0 界面详解	39
2.1.1 Turbo C 2.0 菜单详解	39
2.1.2 Turbo C 2.0 的配置文件	44
2.2 Turbo C 2.0 使用实例	44
2.3 本章小结	46
2.4 练习题	46

第3章 程序设计基础

3.1 概述	47
3.1.1 程序设计中的三个主要问题	47
3.1.2 问题分析与算法	49
3.1.3 算法描述工具	51
3.1.4 数据结构	53
3.1.5 程序编码调试运行	54
3.2 计算机语言概述	55
3.2.1 机器语言	55
3.2.2 汇编语言	56
3.2.3 高级语言	56
3.3 例题详解	57
3.4 本章小结	57
3.5 练习题	58
3.5.1 选择题	58
3.5.2 填空题	58

第4章 C 语言概述

4.1 C 语言的发展过程	59
4.2 C 语言的特点	59
4.3 C 源程序的结构	60
4.4 基本的输入与输出	62
4.5 C 语言的上机步骤	63
4.6 例题详解	64
4.7 本章小结	65
4.8 练习题	66
4.8.1 选择题	66
4.8.2 填空题	66

第5章 基本数据类型和表达式

5.1 标识符和变量	67
5.1.1 标识符	67
5.1.2 变量	67
5.2 基本数据类型	68
5.3 常量	69
5.3.1 整型常量	69
5.3.2 实型常量	69
5.3.3 单字符常量	70
5.3.4 字符串常量	70
5.3.5 符号常量	71
5.4 表达式	71
5.4.1 算术运算符	71
5.4.2 关系运算符	72
5.4.3 逻辑运算符	72
5.4.4 自增自减运算符	73
5.4.5 赋值运算符	74
5.4.6 逗号运算符及表达式	75
5.4.7 条件运算符	75
5.4.8 位运算符	76
5.5 数据类型的转换	77
5.6 运算符的优先级和结合性	78
5.7 赋值语句	80
5.8 例题详解	80
5.9 本章小结	81
5.10 练习题	82
5.10.1 选择题	82
5.10.2 填空题	85

第6章 简单的C语言程序设计

6.1 语句概述	88
6.1.1 控制语句	88
6.1.2 函数调用语句	89
6.1.3 表达式语句	89
6.1.4 空语句	89
6.2 数据输出	89
6.2.1 字符输出函数 putchar	89

6.2.2 格式输出函数 printf	90
6.3 数据输入.....	94
6.3.1 字符输入函数 getchar ()	94
6.3.2 格式输入函数 scanf	94
6.4 例题详解.....	96
6.5 本章小结.....	99
6.6 练习题.....	99
6.6.1 选择题	99
6.6.2 填空题	104

第7章 分支结构

7.1 分支程序设计	105
7.1.1 if 语句	105
7.1.2 if 语句嵌套	107
7.1.3 条件运算符的作用	108
7.2 Switch 语句	109
7.3 goto 语句	110
7.4 例题详解	111
7.5 本章小结	114
7.6 练习题	114
7.6.1 选择题	114
7.6.2 填空题	116

第8章 循环控制

8.1 while 语句.....	117
8.2 do-while 语句	119
8.3 for 语句	120
8.3.1 for 语句的一般格式	120
8.3.2 条件表达式缺省的 for 语句.....	120
8.3.3 条件表达式中包含逗号运算符的 for 语句	121
8.4 三种循环语句的比较	121
8.5 break 语句和 continue 语句.....	122
8.5.1 break 语句.....	122
8.5.2 Continue 语句.....	123
8.6 例题详解	124
8.7 本章小结	128
8.8 练习题	128
8.8.1 选择题	128

8.8.2 填空题	131
-----------------	-----

第9章 函数与变量类型

9.1 函数	133
9.1.1 概述	133
9.1.2 函数定义的一般形式	134
9.1.3 有关函数的说明	135
9.1.4 函数的调用形式	138
9.1.5 函数的递归调用	141
9.1.6 库函数简介	144
9.2 变量类型	145
9.2.1 自动型变量(局部变量)	145
9.2.2 外部型变量(全局变量)	146
9.2.3 静态型变量	148
9.2.4 寄存器变量	149
9.3 变量初始化	150
9.4 编译预处理器	150
9.4.1 宏定义	150
9.4.2 文件包含	153
9.4.3 条件编译	154
9.5 例题详解	155
9.6 本章小结	158
9.7 练习题	158
9.7.1 选择题	158
9.7.2 填空题	163

第10章 数 组

10.1 一维数组	170
10.2 二维数组	173
10.3 字符数组和字符串	175
10.3.1 字符数组的定义和初始化	175
10.3.2 字符串和字符串结束标志	176
10.3.3 字符数组的输入和输出	177
10.3.4 字符串处理函数	178
10.4 例题详解	179
10.5 本章小结	180
10.6 练习题	181
10.6.1 选择题	181

10.6.2 填空题·····	183
-----------------	-----

第 11 章 指 针

11.1 指针的概念·····	188
11.2 指针和指针变量·····	189
11.2.1 指针变量定义 ·····	189
11.2.2 指针变量引用 ·····	190
11.2.3 指针作为函数参数引用 ·····	191
11.3 数组和指针·····	193
11.3.1 通过指针访问数组元素 ·····	193
11.3.2 数组作为函数参数 ·····	196
11.3.3 指向多维数组的指针和指针变量 ·····	200
11.4 字符串和指针·····	204
11.4.1 字符串的表达形式 ·····	204
11.4.2 字符串指针作函数参数 ·····	204
11.5 函数与指针·····	206
11.5.1 指针函数·····	206
11.5.2 函数指针·····	209
11.6 指针数组和指向指针的指针·····	210
11.6.1 指针数组·····	210
11.6.2 指向指针的指针 ·····	212
11.7 Turbo C 的内存分配函数 ·····	213
11.8 本章小结·····	215
11.8.1 有关指针的数据类型 ·····	215
11.8.2 指针运算·····	215
11.9 练习题·····	216
11.9.1 选择题·····	216
11.9.2 填空题·····	221

第 12 章 结构体与共用体

12.1 结构体的定义及其变量的初始化·····	228
12.1.1 结构体定义 ·····	228
12.1.2 结构体变量的初始化 ·····	230
12.2 结构体类型变量的引用·····	232
12.3 结构体数组·····	232
12.3.1 定义 ·····	232
12.3.2 结构体数组初始化 ·····	232
12.3.3 应用举例·····	233

12.4 指针和结构体	234
12.4.1 指向结构体变量的指针	234
12.4.2 指向结构体数组的指针	235
12.4.3 结构指针参数	236
12.5 用指针处理链表	236
12.5.1 链表	236
12.5.2 建立链表	237
12.5.3 链表输出	238
12.5.4 对链表中的元素进行删除	239
12.5.5 对链表插入结点	241
12.5.6 主函数	242
12.6 共用体 (联合)	243
12.6.1 概念	243
12.6.2 引用方式	244
12.6.3 共用体的特点	244
12.7 枚举	245
12.8 用 typedef 定义类型	245
12.9 例题详解	245
12.10 本章小结	248
12.11 练习题	249
12.11.1 选择题	249
12.11.2 填空题	253

第 13 章 文 件

13.1 文件概述	255
13.2 文件的处理	256
13.2.1 文件指针	256
13.2.2 文件的打开和关闭	256
13.2.3 文件的读和写	257
13.3 例题详解	265
13.4 本章小结	267
13.5 练习题	268
13.5.1 选择题	268
13.5.2 填空题	268

第 14 章 常见错误分析和程序调试

14.1 常见错误分析	270
14.1.1 遗漏分号或分号位置错误	270

14.1.2	路径表示的错误	270
14.1.3	混淆赋值号(=)与比较符(==)	270
14.1.4	遗漏花括号	270
14.1.5	括号不配对	271
14.1.6	大小写字母的区别	271
14.1.7	忘记定义变量	271
14.1.8	错误使用指针	271
14.1.9	开头语句中忘记中断语句 break	272
14.1.10	混淆字符和字符串的表示形式	273
14.1.11	自加(++)和自减(--)错误	273
14.1.12	地址传送失败	273
14.1.13	数组及数组下标	273
14.1.14	int 型数据的数值范围	274
14.1.15	函数的使用	274
14.1.16	混淆数组名及指针变量区别	277
14.1.17	混淆结构体类型和结构体变量区别	278
14.1.18	使用文件时忘记打开文件或打开文件方式不对	278
14.2	错误的检出与分离	279
14.3	程序调试	280
14.3.1	人工检查(静态检查)	280
14.3.2	上机调试	281
14.3.3	分析结果	281
附录 I	ASCII 码表	282
附录 II	Turbo C 常用库函数	283
附录 III	C 语言中的关键字	291
附录 IV	运算符和结合性	292
附录 V	习题参考答案	294
附录 VI	模拟试题	301
附录 VII	模拟试题参考答案	325

第 1 章 计算机软件基础知识

1.1 数据结构与算法

借助于计算机解决问题，首先需要了解所处理对象的性质和特点，即所操作对象的数据结构，然后再设计解决问题的方法和步骤，即设计一个合理的算法，这就是通常所说的“程序 = 数据结构 + 算法”。

1.1.1 算法的基本概念

“算法” (Algorithm) 一词最早来自 9 世纪波斯数学家比阿勒·霍瓦里松的一本影响深远的著作《代数对话录》。20 世纪的英国数学家图灵提出了著名的图灵论点，并抽象出了一台机器，这台机器被我们称之为图灵机。图灵的思想对算法的发展起到了重要的作用。一般来说，算法是指完成一个任务或解决一个问题所需要的具体步骤和方法的描述。在这里我们说的算法是指计算机能执行的算法。

1. 算法分类

计算机算法可分为两大类，一类是数值运算算法，另一类是非数值运算算法。数值运算算法主要是求数值解，如求方程的解、求函数的定积分等，非数值运算的范围则非常广泛，如人事管理、图书检索等。

2. 算法特征

一个科学的算法必须具备以下特征：

(1) 有穷性。一个算法必须保证执行有限步之后结束，而不能是无限的。更进一步说，有穷性是指在合理的范围内结束运算。显而易见，如果一个算法需计算机执行几百年或更长时间才结束，显然是不合理的。

(2) 确定性。算法的每一步骤必须有确切的定义而不能模棱两可，算法中不能出现诸如“一个比较大的数”等模糊描述。

(3) 有零个或多个输入。

(4) 有一个或多个输出。算法的目的是为了解决问题，一个没有输出的算法是不能解决任何问题，因而是没有意义的。

(5) 有效性。算法中的每一个步骤都应当能有效地执行，并得到确定的结果。例如，“若 $n=0$ 则执行 m/n ”是无法有效执行的。

3. 算法表示

一个计算机算法可以用自然语言、流程图、N-S 图等来表示。

4. 算法分析

算法分析的任务是对设计出的每一个具体的算法，利用数学工具，讨论各种复杂度，以探讨某种具体算法适用于哪类问题，或某类问题宜采用哪种算法。

算法的复杂度分时间复杂度和空间复杂度。

- 时间复杂度：在运行算法时所耗费的时间为 $f(n)$ （是 n 的函数）。
 - 空间复杂度：实现算法所占用的空间为 $g(n)$ （也是 n 的函数）。
- $O(f(n))$ 和 $O(g(n))$ 称为该算法的复杂度。

1.1.2 数据结构的定义

数据结构是计算机科学与技术领域上广泛被使用的术语。尽管它至今还未有一个被一致公认的定义，但其内容已得到大家一致公认。数据结构用来反映一个数据的内部构成，即一个数据由哪些成分数据构成，以什么方式构成，呈什么结构。数据结构有逻辑上的数据结构和物理上的数据结构之分。逻辑上的数据结构反映成分数据之间的逻辑关系，而物理上的数据结构反映成分数据在计算机内部的存储安排。数据结构是数据存在的形式。

数据结构是信息的一种组织方式，其目的是为了提高算法的效率，它通常与一组算法的集合相对应，通过这组算法集合可以对数据结构中的数据进行某种操作。

一般数据结构可采用顺序存储结构和链式存储结构两类主要的存储方式，大多数数据结构的存储表示都采用其中的一类方式，或是两类方式的结合。

1. 顺序存储结构

这种存储方式主要用于线性数据结构，它把逻辑上相邻的数据元素存储在物理上相邻的存储单元内，结点之间的关系由存储单元的邻接关系来实现。

顺序存储结构的主要特点是：①结点中只有自身信息域，没有连接信息域，因此存储密度大，存储空间利用率高；②可以通过计算直接确定数据结构中第 i 个结点的存储地址 Li ，计算公式为 $Li = L_0 + (i - 1) * m$ ，其中 L_0 为第一个结点的存储地址， m 为每个结点所占用的存储单元个数；③插入、删除运算不便，会引起大量结点的移动。

2. 链式存储结构

链式存储结构就是在每个结点中至少包括一个指针域，用指针来体现数据元素之间逻辑上的联系。这种存储结构可把逻辑上相邻的两个元素存放在物理上不相邻的存储单元中；还可以在线性编址的计算机存储器中表示结点之间的非线性联系。

链式存储结构的主要特点是：①结点中除自身外，还有表示连接信息的指针域，因此比顺序结构的存储密度小，存储空间利用率低；②逻辑上相邻的结点物理上不必邻接，可用于线性表、树、图等多种逻辑结构的存储表示；③插入、删除操作灵活方便，不必移动结点，只要改变结点中的指针即可。

除上述两种主要存储方式外，散列法也是在线性表和集合的存储表示中常用的一种存储方式。

1.1.3 线性表结构

1. 线性表的定义

线性表（Linear List）是最常用并且最简单的一种数据结构。它是由 n ($n \geq 0$) 个数

据元素（结点） $a_1, a_2, \dots, a_n$ 组成的有限序列。

(1) 数据元素的个数 n 定义为表的长度 ($n=0$ 时称为空表)。

(2) 将非空的线性表 ($n>0$) 记作: $(a_1, a_2, \dots, a_n)$ 。

(3) 数据元素 a_i ($1 \leq i \leq n$) 只是个抽象符号, 其具体含义在不同情况下可以不同。

在一些比较复杂的线性表中, 一个数据元素可以由若干个数据项组成。在这种情况下, 一般把数据元素称为记录, 含有大量记录的线性表也称为文件。

例 1 英文字母表 ($A, B, \dots, Z$) 是线性表, 表中每个字母是一个数据元素 (结点)。

例 2 一副扑克牌的点数 ($2, 3, \dots, 10, J, Q, K, A$) 也是一个线性表, 其中数据元素是每张牌的点数。

2. 线性表的存储

线性表可采用顺序方式存储和链式方式存储。在各种高级语言中的一维数组就是用顺序方式存储的线性表, 因此也常用一维数组来称呼顺序表。下面主要讨论的线性表对象是指顺序表。

3. 线性表的基本操作

线性表是一种相当灵活的数据结构, 不仅可以查找访问其数据元素, 也可以根据需要增大或缩小其长度, 即可对线性表进行插入和删除数据元素运算。

常见的线性表的基本运算如下:

(1) $\text{InitList}(L)$ 。构造一个空的线性表 L , 即表的初始化。

(2) $\text{ListLength}(L)$ 。求线性表 L 中的结点个数, 即求表长。

(3) $\text{GetNode}(L, i)$ 。取线性表 L 中的第 i 个结点, 这里要求 $1 \leq i \leq \text{ListLength}(L)$ 。

(4) $\text{LocateNode}(L, x)$ 。在 L 中查找值为 x 的结点, 并返回该结点在 L 中的位置。若 L 中有多个结点的值和 x 相同, 则返回首次找到的结点位置; 若 L 中没有结点的值为 x , 则返回一个特殊值表示查找失败。

(5) $\text{InsertList}(L, x, i)$ 。在线性表 L 的第 i 个位置上插入一个值为 x 的新结点, 使得原编号为 $i, i+1, \dots, n$ 的结点变为编号为 $i+1, i+2, \dots, n+1$ 的结点。这里 $1 \leq i \leq n+1$, 而 n 是原表 L 的长度。插入后, 表 L 的长度加 1。

(6) $\text{DeleteList}(L, i)$ 。删除线性表 L 的第 i 个结点, 使得原编号为 $i+1, i+2, \dots, n$ 的结点变成编号为 $i, i+1, \dots, n-1$ 的结点。这里 $1 \leq i \leq n$, 而 n 是原表 L 的长度。删除后表 L 的长度减 1。具体程序实现可参考本书 C 语言相关章节。

1.1.4 栈与队列结构

1. 栈与队列的定义

栈是一种限定仅在表的一端进行插入与删除操作的线性表。允许进行插入与删除操作的这一端称为栈顶, 而另一端称为栈底。不含元素的空表称为空栈, 插入与删除分别称进栈与出栈。

由于插入与删除只能在同一端进行, 所以较先进入栈的元素, 在进行出栈操作时, 要