

全国计算机等级考试辅导丛书

等级考试学习笔记——二级 C 语言

李琳 主编

人 民 邮 电 出 版 社

图书在版编目 (CIP) 数据

等级考试学习笔记. 二级 C 语言 / 李琳主编. —北京: 人民邮电出版社, 2006.1
(全国计算机等级考试辅导丛书)

ISBN 7-115-14189-4

I. 等... II. 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 (2005) 第 142316 号

内 容 提 要

本书本着精讲、精解、精练的原则, 对全国计算机等级考试——二级 C 语言的考试内容进行了全面的复习。主要内容包括考试指南、知识点分析、典型例题解析、练习题及答案、2005 年 4 月和 9 月的考试试题精解, 以及 2 套全真预测试题。

本书采用学习笔记的形式帮助考生进行复习, 每一章先给出知识结构图, 然后将大纲中的知识点通过问答的形式进行归纳讲解; 在每小节后精选了例题, 给出了详细的分析解答; 每章的最后对本章的重点难点进行了总结; 并给出练习题和答案。另外, 本书配有光盘, 光盘中附有 10 套全真预测上机试题。

本书可作为参加全国计算机等级考试——二级 C 语言考试的复习参考书, 也可供 C 语言的初学者学习参考。

全国计算机等级考试辅导丛书 等级考试学习笔记——二级 C 语言

-
- ◆ 主 编 李 琳
责任编辑 王文娟
 - ◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街 14 号
邮编 100061 电子函件 315@ptpress.com.cn
网址 <http://www.ptpress.com.cn>
北京顺义振华印刷厂印刷
新华书店总店北京发行所经销
 - ◆ 开本: 787×1092 1/16
印张: 22.75
字数: 557 千字 2006 年 1 月第 2 版
印数: 8 001—13 000 册 2006 年 1 月北京第 1 次印刷

ISBN 7-115-14189-4/TP · 5081

定价: 32.00 元 (附光盘)

读者服务热线: (010)67132692 印装质量热线: (010)67129223

前 言

全国计算机等级考试是经原国家教育委员会批准，由教育部考试中心主办，用于考查应试人员计算机应用知识与能力的等级水平考试。

本书以新颁布的全国计算机等级考试大纲为依据，在综述大纲规定的主要知识点的基础上，按照新大纲对典型例题进行了分类和筛选，并给出了详细的解析。每章的最后对本章的重点、难点进行了总结，并给出可供考生举一反三的练习题和习题答案。本书包括笔试和上机（上机考试模拟系统在配套光盘中）两大部分，使考生一书在手即可完成全面的考前训练。

本书结构新颖，内容全面，针对性强，重点突出。充分研究了考生复习的特点和规律，全书的讲解形式：

考试指南  本章知识结构图  以问答形式给出大纲中的知识点  精选例题讲解  单元练习  习题答案  重点、难点总结  全套模拟试卷 + 上机光盘

全书共分为 15 章。

第 0 章是考试指南，内容包括二级 C 语言考试的考试说明、考试大纲、对新大纲变化的分析、对考试特点的分析，以及考前准备和注意事项。

第 1 章～第 13 章内容包括公共基础知识，C 程序设计的初步知识，数据类型及其运算，顺序结构程序设计，选择结构程序设计，循环结构程序设计，数组，函数，指针，结构体与共用体，位运算，文件操作和上机操作。

第 14 章给出了 2005 年 4 月和 9 月的考试试题和详细解析。

附录中给出了 2 套全真预测试题，便于考生了解试题形式、出题范围和试题难度，测试自己的水平，可以作为实战训练的全真模拟试卷。

本书采用学习笔记的形式帮助考生进行复习。另外，本书配有光盘，光盘中附有 10 套全真预测上机试题，读者可通过实际操作掌握上机考试内容，熟悉上机考试环境。

从历年考试试题来看，同一内容重复考查的情况比较多，试题也有规律可循。建议考生多做一些历年试题，以巩固基础知识，找出薄弱点，有针对性地加强复习。

衷心祝愿读者能够顺利通过考试，成为一名合格的计算机应用人才，也希望本书在备考的过程中能够助您一臂之力。

由于时间仓促，书中难免有疏漏之处，恳请读者批评指正。欢迎与本书责任编辑联系，您可发电子邮件至 wangwenjuan@ptpress.com.cn。

编者

全国计算机等级考试辅导丛书编委会

主 编:

李 琳

副主编:

崔江浩

刘长东

翟方国

编写人员:

宋业珏珏

赵广平

刘长东

张晓峰

高 洁

刘士才

李 响

于华明

目 录

第 0 章	考试指南	1
0.1	新大纲考试分析	1
0.2	新版考试大纲	2
0.3	考试技巧及经验总结	5
第 1 章	公共基础知识	7
1.1	数据结构基础知识	7
1.2	程序设计基础知识	18
1.3	软件工程基础知识	19
1.4	数据库的基础知识	25
第 2 章	C 语言概述	32
2.1	C 语言的特点	32
2.2	简单的 C 程序介绍	34
2.3	C 语言程序的编译和执行	37
2.4	C 程序的上机步骤	40
第 3 章	C 程序设计的初步知识	43
3.1	程序的构成	43
3.2	数据类型	45
3.3	数据类型转换	58
3.4	运算符和表达式	61
第 4 章	顺序结构	75
4.1	数据输出	75
4.2	数据输入	80
第 5 章	选择结构	86
5.1	if 语句	86
5.2	switch 语句	92
第 6 章	循环结构	99
6.1	while 和 do while 循环结构	99
6.2	for 循环结构	105
6.3	continue 语句和 break 语句	110
6.4	循环结构的嵌套	113
第 7 章	数组	118
7.1	一维数组	118
7.2	二维数组	124
7.3	字符数组	127

第 8 章 函数	138
8.1 库函数	138
8.2 函数的定义和返回值	139
8.3 函数的参数	142
8.4 函数的调用	149
8.5 局部变量和全局变量	154
8.6 动态存储变量和静态存储变量	156
8.7 内部函数与外部函数	160
第 9 章 指针	165
9.1 变量的指针和指向变量的指针变量	165
9.2 数组的指针和指向数组的指针变量	171
9.3 字符串的指针和指向字符串的指针变量	179
9.4 函数的指针和指向函数的指针变量	186
9.5 返回指针值的函数	186
9.6 指针数组和指向指针的指针	187
第 10 章 结构体与共用体	200
10.1 结构体	200
10.2 共用体	214
第 11 章 位运算	223
11.1 位运算概述	223
11.2 位运算符的含义及应用	224
第 12 章 文件	230
12.1 文件类型指针	230
12.2 文件的打开与关闭	231
12.3 文件的读写	234
12.4 文件的定位	243
第 13 章 上机试题精解	249
13.1 操作步骤	249
13.2 试题精解	251
第 14 章 2005 年二级 C 语言真题精解	272
14.1 2005 年真题特点分析	272
14.2 2005 年 4 月份真题精解	276
14.3 2005 年 9 月份真题精解	305
附录 1 全真预测试题（一）	331
附录 2 全真预测试题（二）	341

第 0 章 考试指南

0.1 新大纲考试分析

新考试大纲对所考查的基本内容作了重新说明,对考试重点进行了重新界定。鉴于二级 C 语言考试内容的基础性,国家教育部在对考试内容进行部分改革的同时保留了大部分原有考试内容,并且也没有增大试题的难度,因此只要考生制定正确的学习和复习方法,认真地对待考试,最终会取得好成绩。

考核内容主要包括基础知识和程序设计。基础知识主要涉及数据结构与算法、程序设计方法、软件工程、数据库基础知识共 4 个部分。二级基础知识在各科笔试中的比重为 30% (30 分),题型为选择题和填空题,其余为本考试科目试题,在笔试中所占比重为 70% (70 分)。二级上机考试中取消对 DOS 部分的考核。

新考试大纲中加入了数据结构、数据库、软件工程等基本知识,同时删去了与 DOS 和 Windows 有关的考查内容。虽然考试大纲作了这些变动,但考生完全没有必要为此担忧,因为很多新内容考生在学习 C 语言的时候都学过,只是过去没有单独拿出来作为独立的章节进行系统地学习而已,比如数据结构的排序和查找等内容。对于数据库和软件工程等新内容,考试的难度应该不会太大,所考内容以基本概念和基础知识为主,考生应认真阅读本书的有关章节,看懂相应部分的例题。

除了新添加的内容,其他的考查内容主要是 C 语言方面的知识,对于这部分内容的考查,新旧大纲变化不大,考试形式也大体相似,仍然分为笔试和上机两部分。要做好这部分题目,考生依然需要具备扎实的 C 语言基础知识,对 C 语言的数据类型、数据运算、语句、程序设计的基本结构(如顺序结构、选择结构和循环结构等)、函数定义、函数调用、数组、指针、结构体数据类型、共用体数据类型、位运算及文件的操作等基本知识都要熟记并会灵活应用。在复习过程中,建议考生在认真学习教材的同时,认真研究历年考试真题及有影响的模拟题,对于这些题目,考生不但要牢牢掌握,并且要多加体会,做到举一反三,在做题过程中不但要了解题目考查的知识点,进一步掌握基本知识,而且要弄清楚知识点考查的方式。

比基础知识更高一层的就是对 C 语言编程方面的考查,这主要考查运用 C 语言的基础知识编写程序的能力。对于这部分考查内容,考生除了自己要动手编写一些程序练习以外,最重要的是要研究考试中出题的可能类型,并找出相应的解决方法。从这几年的 C 语言编程题来看,考查内容主要集中于字符串的处理、数组的定义及应用和复杂数学公式的编写等几大类。字符串的处理又可以分为几个小类,如字符查找、字符替换以及两个字符串的比较和操作等,其中对于字符串的处理往往要用到指针,而指针又是 C 语言的难点和重点,所以对这部分内容考生更应该多加注意和揣摩;对于数组的处理,简单的考查内容主要有数组元素的加减,复杂的有数组的排序等;而复杂数学公式的编写主要考查循环结构,因为大部分复杂数学公式都是用循环结构来编写的。因此综合起来,要求考生具备熟练运用数学知识和编程知识对某些数学公式巧妙编程并能得出正确结果的能力。作为复习建议,考生可以对历年

真题中的编程题进行汇总和模拟练习,在亲身实践的基础上仔细思考和揣摩,进行分类梳理,对于一些常用的经典方法可以记住。这样,考生就会对每一类问题都做到心中有数,脑中有物,也就能够自如地应付考试了。

综观近几年全国计算机等级考试二级 C 语言基础部分和程序设计部分的题目,笔试中大多数考题与大纲要求的基本内容一致,难度也不高,但内容却十分广泛。这反映出二级考试的重点不在试题的难度,而在于考生对基础知识的掌握,所以考生在复习时要多做、多练、多思考,全面的系统复习。对于前面的基础知识,虽然新考试大纲将数据结构等新知识纳入考试范围,这对于广大考生来说,确实是一个挑战,但正如前面讲过的,这部分内容会以概念性的基础知识为主,对于后面十几个和编程有关的题目,形式大多是给出一段程序,其中留出一个空(笔试题部分)或留出一段空(上机部分)要考生填写,而且往往是填写关键的相关语句,因此需要考生扎实地掌握编程的基础知识。另外笔者建议考生能将上机题中常出的考试类型的算法熟练写出来,因为上机考试一般比较紧张,调试的时候很容易因为微小的错误造成考试失败。

总之,考生要扎实地一部分一部分地进行学习和全面复习,尤其是要按照上面所讲的去做,这样更容易有的放矢,迅速地掌握 C 语言以及与其他科目相关的基础知识,掌握 C 语言编程的精华,做到事半功倍。

本书按照新大纲进行编排,精选了历年真题并配以详细解析,对于没有考过的部分,也按照 C 语言二级考试的难度给出了部分模拟题。我们相信只要考生认真准备,一定会在二级考试中取得理想的成绩。

0.2 新版考试大纲

0.2.1 公共基础知识

1. 基本要求

- (1) 掌握算法的基本概念。
- (2) 掌握基本数据结构及其操作。
- (3) 掌握基本排序和查找算法。
- (4) 掌握逐步求精的结构化程序设计方法。
- (5) 掌握软件工程的基本方法,具有初步应用相关技术进行软件开发的能力。
- (6) 掌握数据库的基本知识,了解关系数据库的设计。

2. 考试内容

- (1) 基本数据结构与算法。
 - ① 算法的基本概念;算法复杂度的概念和意义(时间复杂度与空间复杂度)。
 - ② 数据结构的定义;数据的逻辑结构与存储结构;数据结构的图形表示;线性结构与非线性结构的概念。
 - ③ 线性表的定义;线性表的顺序存储结构及其插入与删除运算。
 - ④ 栈和队列的定义;栈和队列的顺序存储结构及其基本运算。
 - ⑤ 线性单链表、双向链表与循环链表的结构及其基本运算。
 - ⑥ 树的基本概念;二叉树的定义及其存储结构;二叉树的前序、中序和后序遍历。
 - ⑦ 顺序查找与二分法查找算法;基本排序算法(交换类排序、选择类排序、插入类排

序)。

(2) 程序设计基础。

- ① 程序设计方法与风格。
- ② 结构化程序设计。
- ③ 面向对象的程序设计方法。

(3) 软件工程基础。

- ① 软件工程基本概念, 软件生命周期概念, 软件工具与软件开发环境。
- ② 结构化分析方法, 数据流图, 数据字典, 软件需求规格说明书。
- ③ 结构化设计方法, 总体设计与详细设计。
- ④ 软件测试的方法, 白盒测试与黑盒测试, 测试用例设计, 软件测试的实施, 单元测试、集成测试和系统测试。
- ⑤ 程序的调试, 静态调试与动态调试。

(4) 数据库设计基础。

- ① 数据库的基本概念: 数据库、数据库管理系统、数据库系统。
- ② 数据模型, 实体联系模型及 E-R 图, 从 E-R 图导出关系数据模型。
- ③ 关系代数运算, 包括集合运算及选择、投影、连接运算, 数据库规范化理论。
- ④ 数据库设计方法和步骤: 需求分析、概念设计、逻辑设计和物理设计的相关策略。

3. 考试方式

① 公共基础知识的考试方法为笔试, 与 C 语言程序设计的笔试部分合为一张试卷。公共基础知识部分占全卷的 30 分。

② 公共基础知识有 10 道选择题和 5 道填空题。

0.2.2 C 语言程序设计

1. 基本要求

- (1) 熟悉 TURBO C 集成环境。
- (2) 熟练掌握结构化程序设计的方法, 具有良好的程序设计风格。
- (3) 掌握程序设计中简单的数据结构和算法。
- (4) TURBO C 集成环境下, 能够编写简单的 C 程序, 并具有基本的纠错和调试程序的能力。

2. 考试内容

(1) C 语言的结构。

- 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。

(2) 数据类型及其运算。

- ① C 语言的数据类型(基本类型、构造类型、指针类型和空类型)及其定义方法。
- ② C 语言运算符的种类、运算优先级和结合性。
- ③ 不同类型数据间的转换与运算。
- ④ C 语言表达式类型(赋值表达式、算术表达式、关系表达式、逻辑表达式、条件表达

式和逗号表达式)和求值规则。

(3) 基本语句。

- ① 表达式语句、空语句和复合语句。
- ② 数据的输入与输出,输入输出函数的调用。
- ③ 复合语句。
- ④ goto 语句和语句标号的使用。

(4) 选择结构程序设计。

- ① 用 if 语句实现选择结构。
- ② 用 switch 语句实现多分支选择结构。
- ③ 选择结构的嵌套。

(5) 循环结构程序设计。

- ① for 循环结构。
- ② while 和 do while 循环结构。
- ③ continue 语句和 break 语句。
- ④ 循环的嵌套。

(6) 数组的定义和引用。

- ① 一维数组和多维数组的定义、初始化和引用。
- ② 字符串与字符数组。

(7) 函数。

- ① 库函数的正确调用。
- ② 函数的定义方法。
- ③ 函数的类型和返回值。
- ④ 形式参数与实在参数,参数值的传递。
- ⑤ 函数的正确调用、嵌套调用和递归调用。
- ⑥ 局部变量和全局变量。
- ⑦ 变量的存储类别(自动、静态、寄存器和外部),变量的作用域和生存期。
- ⑧ 内部函数与外部函数。

(8) 编译预处理。

- ① 宏定义:不带参数的宏定义;带参数的宏定义。
- ② “文件包含”处理。

(9) 指针。

- ① 指针与指针变量的概念,指针与地址运算符。
 - ② 变量、数组、字符串、函数和结构体的指针以及指向变量、数组、字符串、函数、结构体的指针变量。通过指针引用以上各类型数据。
 - ③ 用指针作函数参数。
 - ④ 返回指针值的指针函数。
 - ⑤ 指针数组,指向指针的指针,main 函数的命令行参数。
- (10) 结构体(即“结构”)与共用体(即“联合”)。
- ① 结构体和共用体类型数据的定义方法和引用方法。

② 用指针和结构体构成链表，单向链表的建立、输出、删除与插入。

(11) 位运算。

① 位运算符的含义及使用。

② 简单的位运算。

(12) 文件操作。

只要求缓冲文件系统（即高级磁盘 I/O 系统），对非标准缓冲文件系统（即低级磁盘 I/O 系统）不要求。

① 文件类型指针（FILE 类型指针）。

② 文件的打开与关闭（fopen、fclose）。

③ 文件的读写（putc、getc、puts、gets、fread、fwrite、fprintf、fscanf），文件的定位（rewind、fseek 函数）。

0.3 考试技巧及经验总结

0.3.1 笔试技巧

考生在备考时要深化对基本概念和知识点的理解，并注重实践，同时要做一些针对性的练习，尤其是历年真题，从中找出规律以及解题技巧。

1. 选择题

这种类型的题要求考生从 4 个待选答案中选择 1 个正确答案。考生可综合运用直选法、排除法等多种方法。这类题目考的知识点往往比较多且细，容易出错，考生在做这类题时，切忌不加分析，一看就选，从而漏掉正确的答案。

2. 填空题

填空题主要考查考生对基础知识的准确理解。对于这种类型的题，考生应深刻理解题意，明确题目要求，运用相关知识做出正确回答。在历年考试中，这部分试题的得分对考生考试成绩有重要影响。在阅卷过程中发现考生常犯的错误有以下几个方面。

(1) 基础知识掌握不扎实，概念理解不准确。

(2) 答案表述不准确，会做的题却不得分或少得分。

(3) 注意答题卡的号码并不是试题卷上的题号，而是填空的顺序号。许多考生答题时将答案张冠李戴，位置写错，白白丢了分。

总之，只要考生能准确理解基本知识点，善于动手动脑多练习，举一反三，触类旁通，就能从中找出规律，轻松通过笔试。

0.3.2 上机技巧

上机考试的学习过程是一个由生疏到精通的渐进过程，应先打下扎实的 C 语言基础，对 C 语言的一些基本结构和用法有清晰的了解。

1. 变量的使用语法。C 语言规定，各种变量在使用前要先定义，但有部分考生在使用前不对变量定义，从而产生错误，不能通过编译环节。还有些考生在定义后，往往对变量不进行初始化，就拿到程序中应用。由于这种作法并没有语法上的错误，所以 C 语言的编辑器查不出错误，但是这可能会导致结果与题目要求相差很大，且在考生排除错误时，不容易找出错误的所在。建议考生在定义变量后，在紧接的代码行中进行初始化，养成好的编程习惯。

运算符中要注意赋值运算符“=”和关系运算符“==”的区别，根据人们常有的思维定式，常把数学运算符中的“=”当作 C 语言中的关系运算符“==”。

2. 选择结构和循环结构。常用的选择语句是 `if()` 语句和 `switch()` 语句，当 `if()` 语句后多个 `else` 时，应当注意 `else` 总是与它上面、离它最近的一个 `if` 相配对，这点在读、写程序时都应注意。`switch()` 语句常与 `case`、`break` 等关键字结合使用，应该注意 `switch(表达式)` 中的“表达式”必须有一个定值，即可以是常量、常量表达式、字符常量等，而不能是一个“变动着的值或表达式”；在执行相应的 `case` 分支时，如果分支语句后没有出现“`break`”，则即便 `case` “表达式”中“表达式”的值与 `switch(表达式)` 中“表达式”的值不等，程序也不会跳出分支，仍然会继续逐条执行。在一些占用内存的操作中，如果出现死循环，则很容易导致死机，因此应注意，循环结构中循环条件的设定一定要杜绝死循环。

3. 数组的应用。例如从文件中读取英文文章，根据行列关系存入二维数组中、读整数存入一维或二维数组中等情况。对于这类问题，需要借助一些中转数组来完成，考生使用自己定义的数组时，应该特别注意，数组在定义后一定要初始化，尤其是字符数组。考生可以用 `memset(string, 0, count)` 函数来初始化数组，也可用循环语句来初始化数组。否则当定义数组的列数大于字符串的长度时，后面一部分没有赋初值的数组元素将产生乱码或特殊字符。

4. 函数。考生在解题前应看懂各函数间的互调关系，注意形参和实参的数据类型，确保形参和实参的数据类型一致。

5. 指针。指针是 C 语言的精华，也是机试考查的重点内容。提醒考生应该注意，指针的类型必须与所指元素的类型相同。另外指针的初始化比变量和数组的初始化更重要，没有初值的指针，当指针越界后，会引起死机，如果说死循环带来的死机是“假死”，大部分可以用“`Ctrl+Alt+Del`”来恢复，但指针越界引起的死机却是“真死”，很难进行恢复。

6. 结构体。结构体也是每年必考的内容，题型大多以“产品问题”、“学生问题”的形式出现。考生在复习结构体这部分内容时，要注意结构体名称和结构体变量之间的区别。考试内容还会涉及结构体数组，应该熟悉结构体元素的调用方法、结构体变量间的赋值和比较方式等。

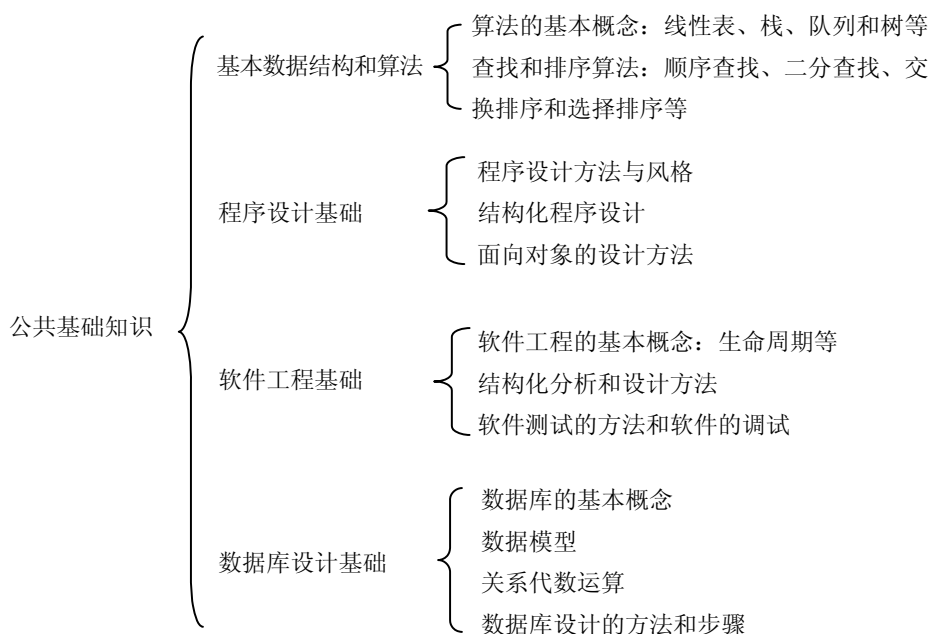
7. 位运算和文件的读写操作。这部分内容，考生要了解位的几种特殊运算，了解第 12 章文件中的读文件函数、写文件函数、打开文件函数和关闭文件函数等几个主要函数的知识。

考生完成按章节复习后，应进行各种模拟练习，熟悉考试中的相关软件的使用方法和技巧，只有这样，才能在考试中不会因为对各种软件陌生而失分。

第 1 章 公共基础知识

公共基础知识是学习计算机知识的基础，是新大纲添加的内容，其中包括基本数据结构与算法、程序设计基础、软件工程基础和数据库设计基础 4 个部分。这部分题目在以前的考试中从来没有出现过，考生可能会感到陌生，其实对于二级考试而言，这些都是一些基本的概念。只要认真学习本章，就能够掌握这些知识，本章中的一些知识在学习 C 语言的时候已经涉及到，例如数据结构的知识，本章将系统地介绍这部分知识。

知识结构图



大纲中的知识点

1.1 数据结构基础知识

 问题 1：数据结构的定义，数据结构的逻辑结构和存储结构，数据结构的线性和非线性含义是什么？

数据结构是指数据对象及其相互关系和构造方法。一个数据结构 B ，形式上可以用一个二元组表示为 $B=(A,R)$ 。其中， A 是数据结构中数据（称为结点）的非空有限集合， R 是定义在 A 上的关系的非空有限集合。结构是指结点之间的关系，数据结构就是结点的有限集合

和关系的有限集合。数据结构中, 结点及结点间的相互关系是数据的逻辑结构。数据结构在计算机中的存储内容, 一般包括结点的值和结点间的关系, 数据结构的存储形式就是数据的存储结构。数据结构按逻辑关系的不同分为线性结构和非线性结构两大类, 其中非线性结构可分为树形结构和图结构, 树形结构又可分为树结构和二叉树结构。

问题 2: 什么是线性表?

数据结构中, 线性结构习惯称为线性表。线性表是最简单、最常用的一种数据结构, 它是由相同类型的结点组成的有限序列。一个由 n 个结点 $e_0, e_1, \dots, e_{n-1}$ 组成的线性表记为 $(e_0, e_1, \dots, e_{n-1})$ 。线性表的结点个数称为线性表的长度, 长度为 0 的线性表称为空的线性表, 简称空表。对于非空线性表, e_0 是线性表的第一个结点, e_{n-1} 是线性表的最后一个结点。线性表的结点构成一个序列, 对序列中两个相邻结点 e_i 和 e_{i+1} , 称前者是后者的前驱结点, 后者是前者的后继结点。线性表最重要的性质是线性表中结点的相对位置是确定的。

线性表的结点也称为表元, 或称为记录, 它们要求是同一类型的数据。线性表的结点可由若干个成分组成, 其中惟一能标识表元的成分称为关键字, 简称键。线性表包含的表元个数可以动态增加或减少, 可以在任何位置插入或删除表元。

线性表常用的运算可分成 4 类: 查找运算、插入运算、删除运算以及其他运算。每类又包括若干种运算, 如查找运算有两种, 一种是查找线性表中某个结点的值, 另一种是在线性表中查找具有给定键值的记录。

线性表的存储方式有很多种, 其中最常用的是顺序存储和链接存储。

1. 线性表的顺序存储

线性表的顺序存储是最简单的存储方式。通常使用一个足够大的数组, 从数组的第一个元素开始, 将线性表的结点依次存储在数组中。用数组元素的顺序存储体现线性表中结点的先后次序关系。其最大优点是能直接访问线性表中的任意一个结点。

2. 线性表的链接存储

用单链表存储线性表是线性表的链接存储方式。从单链表的第一个表元开始, 将线性表的结点依次存储在单链表的各表元中。单链表的每个表元除了要存储线性表结点的信息外, 还要有一个成分用来存储其后继结点的指针。

问题 3: 栈和队列的含义是什么?

栈是只允许在同一端进行插入和删除运算的线性表。允许插入和删除的那一端称为栈顶, 另一端称为栈底。若有栈 $S=(s_0, s_1, \dots, s_{n-1})$, 则 s_0 为栈底结点, s_{n-1} 为栈顶结点。习惯称插入结点为进栈, 删除结点为出栈。因为最后进栈的结点必定最先出栈, 所以栈具有后进先出的特性。可以用顺序存储线性表来表示栈, 也可以用链表实现栈, 用链表实现的栈称为链接栈。

队列是只允许在一端进行插入运算, 另一端进行删除运算的线性表。允许进行删除运算的一端称为队首, 允许进行插入运算的另一端称为队尾。习惯称插入结点为进队, 删除结点为出队。若有队列 $Q=(q_0, q_1, \dots, q_{n-1})$, 则 q_0 为队首结点, q_{n-1} 为队尾结点。因最先进入队列的结点将最先出队, 所以队列具有先进先出的特性。可以用顺序存储线性表来表示队列, 也可以用链表实现队列, 用链表实现的队列称为链接队列。

问题 4: 线性表的应用, 数组和字符串有哪些特点?

数组是最常用的数据结构之一, 一般用于描述顺序存储的线性表。数组由固定个数的元

素组成,全部元素的类型相同,元素按照顺序存储。每个元素对应一个下标,数组元素按数组名和下标引用,引用数组元素的下标个数称为数组的维数。在C语言中,约定数组的第1个元素的下标为0,其余依次类推,由 n 个元素组成的数组,其最后一个元素的下标为 $n-1$ 。数组元素可以是任何类型的,当元素本身也是数组时,就构成多维数组。多维数组是一维数组的推广,最常用的是二维数组。多维数组的所有元素并未排在一个线性序列里,要顺序存储多维数组就需要按一定次序把所有的数组元素排在一个线性序列里,常用的排列次序有行优先顺序和列优先顺序两种。对于多维数组,C语言按行优先顺序存放。

字符串是非数值处理应用中重要的处理对象。字符串是由某字符集上的字符所组成的有限字符序列。不包含任何字符的字符串称为空字符串。字符串所包含的有效字符数称为字符串长度。一个字符串中任意连续的子序列称为该字符串的子串。

问题5: 什么是树? 其存储结构如何? 树的遍历有几种?

树和二叉树是非线性数据结构,用它们能够很好地描述有分支和有层次的数据集合。习惯称树和二叉树数据结构为树形结构。许多算法常用树形结构描述问题的求解过程或求解对策等。二叉树有确定的分支个数,可以为空,有良好的递归性质,所以同一般树相比,更适合于程序设计,因此也常将树转换成二叉树进行处理。

1. 树的基本概念

树是一种多分支、多层次的数据结构,由一组结点组成。由于它呈现出与自然界的树类似的结构形式,所以称它为树。树是由一个或多个结点组成的有限集 T ,它满足以下两个条件:

- (1) 有一个特定的结点,称为根结点;
- (2) 其余的结点分成 $m(m \geq 0)$ 个互不相交的有限集 $T_0, T_1, \dots, T_{m-1}$ 。其中每个集合又都是一棵树,称 $T_0, T_1, \dots, T_{m-1}$ 为根结点的子树。

树的定义是递归的,即一棵树由子树构成,子树又由更小的子树构成。一棵树至少有一个结点。一个结点的子树的数目,称为结点的度。树中各结点的度的最大值称为树的度。

2. 树的存储结构

树是非线性的结构,不能简单地用线性表来表示。树有多种实用的存储结构,最常用的是标准存储形式和带逆存储形式。在树的标准存储结构中,树中结点的内容可分成两部分:结点的数据和指向子结点的指针。当程序需从结点返回到其父结点时,需要在树的结点中存储其父结点的位置信息,这种存储形式就是带逆存储结构。

3. 树的遍历

在应用树结构时,常要求按某种次序获得树中全部结点的信息,可通过树的遍历操作来实现。常用的树的遍历方法有以下4种。

- 树的前序遍历。首先访问根结点,然后从左到右遍历根结点的各棵子树。
- 树的后序遍历。首先从左到右按后序遍历根结点的各棵子树,然后访问根结点。
- 树的层次遍历。首先访问处于0层上的根结点,然后从左到右依次访问处于1层、2层……上的结点,即自上而下、从左到右逐层访问各层的结点。
- 访问树中所有叶子结点。

问题6: 什么是二叉树? 二叉树的遍历如何进行?

1. 二叉树的基本概念

与一般的树结构比较,二叉树在结构上更规范、更确定,二叉树的每个结点有两棵子二

叉树，分别简称为左子树和右子树。因二叉树可以为空，所以二叉树中的结点可能没有子结点，可能只有一个左子结点，可能只有一个右子结点，也可能同时有左右两个子结点。二叉树定义为：二叉树是一个有限的结点集合，该集合或者为空，或者由一个根结点及其两棵互不相交的左、右二叉子树组成。

二叉树与树不同，首先二叉树可以为空，空的二叉树没有结点。其次，在二叉树中，结点的子树是有序的，分左、右两棵子二叉树。一般情况下，二叉树常采用类似树的标准存储形式来存储。

2. 二叉树的遍历

树的所有遍历方法都适用于二叉树，常用的二叉树遍历方法有以下 3 种。

- 前序遍历。访问根结点→按前序遍历根结点的左子树→按前序遍历根结点的右子树
 - 中序遍历。按中序遍历根结点的左子树→访问根结点→按中序遍历根结点的右子树
 - 后序遍历。按后序遍历根结点的左子树→按后序遍历根结点的右子树→访问根结点
- 以上 3 种遍历方法都是递归定义的。

问题 7：什么是二叉查找树？

查找树便于链式存储，还能实现快速查找。作为一种特殊的二叉树，它或者为空，或者满足以下 3 个条件。

- 该树根结点的左子树非空，其左子树所有结点的键值都小于该树根结点的键值。
- 该树根结点的右子树非空，其右子树所有结点的键值都大于该树根结点的键值。
- 该树的根结点的左子树和右子树均为查找树。

依据以上定义，如果进行中序遍历，即可得到一个从小到大的结点序列。

问题 8：什么是排序与查找？什么是散列表？

对于有 n 个结点的线性表 $(e_0, e_1, \dots, e_{n-1})$ ，将结点中某些数据项的值按递增或递减的次序，重新排列线性表结点的过程，称为排序。排序时参照的数据项称为排序码，通常选择结点的键值作为排序码。在排序过程中，线性表的全部结点都在内存中，并在内存中调整它们在线性表中的存储顺序，称为内排序。在排序过程中，线性表只有部分结点被调入内存，并借助内存调整结点在外存中的存放顺序的排序方法称为外排序。常用排序方法有选择排序、直接插入排序、冒泡排序、希尔排序、堆排序、快速排序、合并排序和外排序，外排序是对大文件的排序。

查找就是在按某种数据结构形式存储的数据集合中，找出满足指定条件的结点。按查找的条件分类，有按结点的关键码查找、按关键码以外的其他数据项查找以及按其他数据项的组合查找等。按被查找数据在内存或在外存分为内存查找和外存查找。按查找的目的分类，有静态查找和动态查找，查找只是为了确定指定条件的结点存在与否，称为静态查找；查找是为了确定结点的插入位置或为了删除找到的结点，称为动态查找。

散列表又称杂凑表，是一种非常实用的查找技术。由于查找码与结点在数据结构中的位置不存在确定的关系，查找只能通过对查找码与结点的关键码的反复比较来实现。通过比较来缩小查找范围，惟一能提高查找效率的方法是通过一次比较来大幅减小查找范围。最理想的情况是直接利用查找码的信息，一次或几次比较便能存取所查结点。为此必须在结点的存储位置和它的关键码间之建立一个确定的关系，从而让查找码直接利用这个关系确定结点的位置。

例题讲解

试题 1 设有 4 个数据元素 a_1 、 a_2 、 a_3 和 a_4 ，对它们分别进行栈操作或队操作。在进栈或进队操作时，按 a_1 、 a_2 、 a_3 、 a_4 次序每次进入一个元素。假设栈或队的初始状态都是空。

现要进行的栈操作是进栈 2 次，出栈 1 次，再进栈 2 次，出栈 1 次。这时，第 1 次出栈得到的元素是 ，第 2 次出栈得到的元素是 。类似地，考虑对这 4 个数据元素进行的队操作是进队 2 次，出队 1 次，再进队 2 次，出队 1 次。这时，第 1 次出队得到的元素是 ，第 2 次出队得到的元素是 。经操作后，最后在栈中或队中的元素还有 个。

- (A) ~ (D) ① a_1 ② a_2 ③ a_3 ④ a_4
 (E) ① 1 ② 2 ③ 3 ④ 0

【解析】 本题考查栈和队的存储操作。

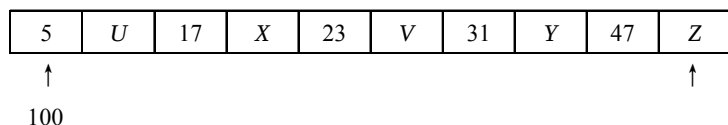
栈是具有先进后出特征的数据结构，队列则具有先进先出特征。

根据题意，元素 a_1 、 a_2 、 a_3 和 a_4 按次序进入。进栈 2 次的结果是 a_1 先进入栈底， a_2 处于栈顶；出栈 1 次的结果是栈顶元素 a_2 被弹出。再进栈 2 次分别将 a_3 和 a_4 压入栈中，此时从栈底到栈顶的元素依次为 a_1 、 a_3 和 a_4 ，这时再出栈 1 次，将 a_4 弹出。

进队列 2 次使 a_1 和 a_2 进队列，且 a_1 为队首元素，出队 1 次的结果是使队首元素 a_1 出队列；再进队列 2 次，使队列由首到尾的元素依次为 a_2 、 a_3 、 a_4 ，出队 1 次，得到的元素是 a_2 。空的栈或队列在被压入 4 个元素，弹出 2 个元素后，剩下 2 个元素。

【答案】 (A) ② (B) ④ (C) ① (D) ② (E) ②

试题 2 线性表具有两种存储方式，即顺序方式和链接方式。现有一个具有 5 个元素的线性表 $L=\{23, 17, 47, 05, 31\}$ ，若它以链接方式存储在下列 100~119 号地址空间中，每个结点由数据（占 2 个字节）和指针（占 2 个字节）组成，如：



其中指针 X、Y 和 Z 的值分别为 、 和 。

在上述链接方式存储时，该线性表的首结点的起始地址为 ，末结点的起始地址为 。

- (A) ~ (E) ① 100 ② 104 ③ 108 ④ 112 ⑤ 116
 ⑥ 120 ⑦ 0 ⑧ 102 ⑨ 106 ⑩ 110

【解析】 本题考查线性表的链式存储。

线性表是最简单、最常用的一种数据结构。在计算机存储器上存放线性表有两种方式：一种是按它们的有序关系依次连续存放，用存储的连续性体现其有序关系；另一种方式是独立地存放各个数据元素，再通过链接体现其有序关系。为了将存储的各个数据元素链接起来，先要知道各个数据元素存储的起始地址，在存储每个数据元素时，还要存储一个指向下一个