

二级C语言一本通

陈业纲 黄波 著

吉林大学出版社

二级C语言一本通

陈业纲 黄波 著

本书是“二级C语言一本通”系列中的一本，旨在帮助读者快速掌握C语言编程。本书内容涵盖了C语言的基础知识、数据类型、运算符、表达式、语句、函数、数组、指针、结构体、文件操作等。本书采用循序渐进的教学方法，通过大量的实例和习题，帮助读者理解和掌握C语言的编程技巧。本书可作为高等院校计算机专业及相关专业的教材，也可供从事计算机工作的工程技术人员参考。

吉林大学出版社

图书在版编目 (CIP) 数据

二级C语言一本通 / 陈业纲, 黄波著. —长春: 吉林大学出版社, 2010. 3

ISBN 978-7-5601-5488-6

I. ①二… II. ①陈…②黄… III. ①C语言—程序设计—高等学校—教材 IV. ①TP312

中国版本图书馆CIP数据核字(2010)第032543号

书 名: 二级C语言一本通
作 者: 陈业纲 黄波 著

责任编辑、责任校对: 张显吉 李天白	封面设计: 创意广告
吉林大学出版社出版、发行	北京德富泰印务有限公司 印刷
开本: 880 × 1230 毫米 1/16	2010年03月 第1版
印张: 16.5 字数: 400千字	2010年03月 第1次印刷
ISBN 978-7-5601-5488-6	定价: 29.00元

版权所有 翻印必究

社址: 长春市明德路421号 邮编: 130021

发行部电话: 0431-88499826

网址: <http://www.jlup.com.cn>

E-mail: jlup@mail.jlu.edu.cn

内容提要

本书是依据最新推出的《全国计算机等级考试(二级公共基础知识和 C 语言程序设计)考试大纲》编写的,全书共分为 3 篇:上篇是公共基础部分,包括数据结构与算法、程序设计基础、软件工程和数据库基础;中篇是 C 语言的理论部分,包括 C 语言概述、程序设计的初步知识、控制结构、数组、函数、指针、编译预处理、结构体、共用体、文件和位运算、C 语言常见算法;下篇是 C 语言上机部分,实验包括错误改正、程序填空、程序设计等内容。

本书可以作为参加全国计算机等级考试——二级 C 语言考试教材,也可作为学习 C 语言的参考书。

序

我们知道人与人交流,依靠语言和文字。我国经历 5000 年的发展与融合,多民族的国家形成了以汉语为主体的语言和文字,它不仅在国内流传,而且也在加速走向世界。

和人类语言类似,计算机这个人类最先进的工具,自它被创建以来,就在进行着“机器语言”的研究和发展,正是借助这种语言,我们不断改进对计算机的应用,将它广泛应用于我们的各种活动,使之成为我们最有用的工具。因此,我们今天要使用计算机,就必须掌握它的语言。

C 语言是 UNIX 的研制者丹尼斯·里奇(Dennis Ritchie)和肯·汤普逊(Ken Thompson)于 1970 年研制 B 语言的基础上发展和完善起来的程序设计语言。1980 年产生 ANSI C,1990 年产生 ISO C90,1999 年产生 ISO C99,也被 ANSI 于 2000 年 3 月采用,它在不断的发展中不断地被标准化。C 语言提供了基础数据、数组、指针、字符串等数据类型,提供了复合、条件语句、循环、转移等种语句,它在注重结构程序设计的同时,又保留了转移语句,在适应常规计算的同时,也十分重视字位级计算,它具有高效、灵活、功能丰富、表达力强和较高的移植性等特点,因而广泛用于系统与应用软件的开发。C 语言编译程序普遍存在于多种操作系统,例如 UNIX、Microsoft Windows、Linux 等。C 语言的设计影响了很多后来的程序语言,例如 C++、Java、C# 等。

本书包含了(线性表、线性链表、栈、队列、树与二叉树等)数据结构与(查找、排序等)算法、(结构、面向对象)程序设计、(结构化分析与设计、软件测试与调试等)软件工程、和(数据模型、R-R 图、关系代数与运算、数据库设计与管理等)数据库基础知识的讲解,奠定了学习 C 程序设计所必需的一些基础知识。

本书的主体内容包含 C 语言的概述、程序设计初步,程序设计的(顺序、选择、循环、函数等)控制结构,(常量、变量、数组、指针、结构、联合、文件和位运算等)数据结构,编译处理等内容作了简洁的归纳和总结,进而解析了大量的示例,提供了大量的习题。正如用范文有利于讲解自然语言一样,用程序片段和完整的程序讲解程序语言是一种好方法,它可把很多抽象的理论、新颖的定义,讲得直观、具体,使读者容易理解和掌握。

本书还提供了大量的上机操作程序,使学生能在计算机上实验程序改错、程序填空、编写程序并编译和执行程序。通过循序渐进的实验,使得学生可系统地掌握相关的基础知识,较快掌握程序设计的方法和技巧,锻炼实际的程序设计能力。

本书依据普通高等院校教学大纲、全国二级等级考试新大纲、以及实际应用的需求编写,注重了实验内容和教材知识的结合,有利于理论联系实际的教学,有利于培养学生的实际编程能力。特别是,本书以新考试大纲规定的考核知识点及能力层次为主线,按新试题结构分类、分章节编写,试题解析力求翔实,并侧重于解题思路及步骤的讲解,这决定了本书具有较强的应试能力,而有助于备试者复习有关内容、了解试题形式并提高应试能力。

本书还提供了应试策略。除了作者的意见之外,我认为应试策略的根本是学生要重视平时学习与练习。平时学习好了,考试自然会轻松、愉快,这才是上策。平时未努力,“临阵磨枪”只能是中策。

我国信息化的发展,需要众多的程序设计人员,学习计算机科学的学生,应当重视计算机程序设计技能培养,从而进可贡献社会,退可求职谋生。总之,国家的信息化是全民的信息化,也是高水平的信息化,我们应该努力而为。

中国科学院数学研究所 徐泽同

2010年1月1日于北京回龙观

目 录

序	1
应试策略	1

第一部分 基础知识

第一章 数据结构与算法	5
1.1 算法	5
1.2 数据结构基础	5
1.3 线性表及存储结构	6
1.4 线性链表	7
1.5 栈和队列	8
1.6 树与二叉树	9
1.7 查找	12
1.8 排序	13
1.9 练习题	16
第二章 程序设计基础	19
2.1 程序设计方法和风格	19
2.2 结构化程序设计	19
2.3 面向对象程序设计	20
2.4 练习题	24
第三章 软件工程基础	26
3.1 基本概念	26
3.2 结构化解析方法	27
3.3 结构化设计方法	28
3.4 软件测试	29
3.5 程序调试	29
3.6 练习题	30

第四章	数据库设计基础	33
4.1	基本概念	33
4.2	数据模型	34
4.3	关系代数	41
4.4	数据库设计与管理	44
4.5	练习题	46
第五章	综合练习	49
5.1	典型问题解析	49
5.2	经典例题	51

第二部分 C 语言

第一章	程序设计基本概念	58
1.1	知识点	58
1.2	典型问题解析	59
1.3	练习题	60
第二章	C 程序设计的初步知识	62
2.1	知识点	62
2.2	典型问题解析	67
2.3	练习题	70
第三章	顺序结构	73
3.1	知识点	73
3.2	典型问题解析	75
3.3	练习题	77
第四章	选择结构	80
4.1	知识点	80
4.2	典型问题解析	83
4.3	练习题	86
第五章	循环结构	89
5.1	知识点	89
5.2	典型问题解析	90
5.3	练习题	92

第六章	函数	96
6.1	知识点	96
6.2	典型问题解析	100
6.3	练习题	104
第七章	指针	110
7.1	知识点	110
7.2	典型例题解析	116
7.3	练习题	118
第八章	数组	122
8.1	知识点	122
8.2	典型问题解析	124
8.3	练习题	126
第九章	字符串	132
9.1	知识点	132
9.2	典型问题解析	134
9.3	练习题	135
第十章	文件	141
10.1	知识点	141
10.2	典型问题解析	144
10.3	练习题	147
第十一章	结构体和共用体	152
11.1	知识点	152
11.2	典型问题解析	155
11.3	练习题	158
第十二章	位运算	163
12.1	知识点	163
12.2	典型问题解析	165
12.3	练习题	166
第十三章	常见算法	169
13.1	知识点	169

13.2 典型问题解析	172
13.3 练习题	179

第三部分 上机部分

1.程序填空题	183
2.改错题	183
3.编程题	183
第一章 程序填空	184
1.1 填空必读	184
1.2 常见例题	186
1.3 常见程序填空	186
第二章 上机改错	203
2.1 上机改错必读	203
2.2 常见改错题	204
2.3 常见改错题	210
第三章 程序编写	219
3.1 程序编写题	219
3.2 常见编程题	226

第四部分 答案

4.1 第一部分	239
4.2 第二部分	240
4.3 第三部分	242

附 录

1. 全国计算机等级考试二级 C 语言考试大纲	245
2. 二级公共基础知识新大纲	247
3. 全国计算机等级考试二级 C 语言上机实际操作过程图解	248
4. 2009 年 9 月全国计算机等级考试	250
5. 参考文献	261

应试策略

参加计算机等级考试,应把握这样的总体原则,即首先要了解考试要点,清楚考试的知识点,再在此基础上重点熟悉笔试考题和上机考题的考试题型与特点,熟读例题题解,并做相应的练习;考生只要遵循这样的原则有步骤、有重点地进行学习和复习,就一定能通过考试。

“计算机等级考试”二级考试分笔试和上机考试两部分。两部分的分数均为 100 分,及格分都是 60 分,只有在两部分考试成绩都及格后,整个二级考试才算及格。

一、笔试

分为选择题和填空题两种题型。这两种题型的解题方法有所不同,在学习时要深化对基本概念和知识点的理解,并注重实践,同时作一些针对性的练习,从历年真题中找出规律性的东西以及解题技巧。

1. 选择题要求从四个待选答案中选择一个正确答案。可综合运用直选法、排除法等多种方法。但是这类题目考的知识点往往都比较多且细,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。例如:

有以下程序

```
main( )
{int a[3][3],* p,i;
 p=&a[0][0];
 for (i=0;i<9;i++)p[i]=i+1;
 printf("%d",a[1][2]);
}
```

程序运行后的输出结果是

A)3 B)6 C)9 D)2

此题主要考的知识点有:(1)数组与指针的关系;(2)数组的初始化和数组元素的引用。在 C 语言中,数组与指针有着非常密切的联系。任何能用数组下标完成的操作也都可以用指针来实现。语句“p=&a[0][0];”使 p 指向数组中的第 0 号元素,即 a[0][0],指针变量 p 中存放了数组元素 a[0][0]的地址,由于数组元素在内存中是连续存储的,因此,我们就可以通过指针变量 p 访问数组中的任何一个元素。本题中通过“for (i=0;i<9;i++)p[i]=i+1;”语句使 a[3][3]=[1,2,3,4,5,6,7,8,9],所以 a[1][2]=6。这里要特别注意 C 语言中数组的下标从 0 开始,不要错选 a[1][2]=3。

2. 填空题

主要考查考生对基础知识的准确理解。对于这类型的题,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,考生这部分试题的得分直接决定考试成绩。在教学中中发现考生常犯的错误有以下两个方面:

(1) 基础知识掌握不扎实,概念理解不准确。

```
main()
{int a=1,b=3,c=5;
if(c=a+b)
printf("yes");
else
printf("no");
}
```

题目问程序运行后的输出结果,答案应是 yes。因为“if(c=a+b)printf(“yes”);”中的条件“c=a+b”是一个赋值语句,c 的值为 a+b=1+3=4,所以条件为真。而许多考生却把赋值语句“c=a+b”理解为关系表达式“c==a+b”即 5=1+3 为逻辑假,故答 no。

(2) 答案表述不准确,会做的题却不得分或少得分。

```
main()
{int i,n[ ]={0,0,0,0,0};
for(i=1;i<=4;i++)
{n[i]=n[i-1]*2+1;
printf("%d",n[i]);
}
```

}问程序运行后的输出结果是_____。

答案是“1 3 7 15”许多考生虽然写对了但却在每个数字后加了标点而不得分如“1,3,7,15”或“1.3.7.15”。

二、机试

主要考查考生编写程序和调试程序等实际操作能力。因此在平时的学习中既要注意理论的学习,也要重视上机操作能力的培养。只要考生熟练掌握了 C 语言程序设计的基本技术、程序编辑、编译工具和程序调试技巧,要顺利通过上机考试并不是难事。

1. 程序填空题的应试策略

上机考试的题目的难度是由浅入深,填空题主要考查考生对 C 语言基础知识的掌握情况和阅读程序的能力。程序并不复杂,只要考生掌握了 C 语言的语法规则,理解了题目的要求,掌握了程序的算法,就可以较容易的将程序补充完整。在一些情况下,结合填空位置的上下语句来猜测需要补充的语句的功能,可以节省一些时间。

2. 改错题的应试策略

改错题的难度就增加了一些,在没有提示的情况下,要求考生自己发现并改正错误。其实也是要求考生熟练掌握 C 语言的语法规则,理解程序的思路和算法。对于一些简单的语法和逻辑错误可以通过仔细阅读程序找到;对于一时不能发现的错误,还可以使用 C 语言编译器的编译命令来帮助发现错误,根据它的错误提示信息找到错误所在;一些语法错误,是可以通过编译的,编译器不会提示错误,这就要求考生做题时认真细致并运用一些调试技巧来发现错误。

3. 编程题的应试策略

编程题对考生能力的要求就更高了,应当先仔细阅读题目,了解题目要求,以及已给出的函数对要编写的函数起了哪些作用,应避免在不明题意的情况下盲目答题。不要急于编程。要理清思路,可以先将复杂的任务逐层分解,要看题目用到了 C 语言中的哪些数据类型,还要看运用了哪些结构。在编写程序的过程中要严格遵守 C 语言的语法规则,避免犯一些常见的语法错误。在程序编写完成后,要经过调试后再运行,避免一些隐性的逻辑错误导致计算机死机,影响考试。总之要顺利完成编程题,考生应注意在平时多学习和积累一些典型的例子和算法。

第一部分 基础知识

本部分是考试的基础,所占分值为 30 分,由于该章的试题比较灵活,因此在学习本章时,要以理解为主,切忌死记硬背,在学习过程中,要注意各个知识点之间的联系和区别,将盘根错节的知识点理顺成知识网络。具体知识点总结如下:

一、算法

该知识点在试卷中一般有 1~2 道题,考生要了解算法的定义、特征、组成要素、常用算法和算法复杂度,其中算法复杂度是考试重点,与之有密切联系的是:查找技术(第一章第 7 节)和排序技术(第一章第 8 节),考生最后复习时,要牢记六种排序方法的时间复杂度和两种查找方法的特点及最好、最坏、平均查找次数。

二、数据结构

该知识点在试卷中一般有 2~4 道题,是本章的重点和难点,考题中所涉及的解析一般并不是教材上的直接知识点,因此在学习过程中,要以是否提高了数据处理的效率(速度、空间)为主线,对每种逻辑结构和其对应的不同存储结构进行解析、比较和总结。

三、线性结构和非线性结构

该知识点是必考的知识点,在学习过程中,要深刻理解和掌握栈和队列的特点(包括逻辑结构特点和不同的存储结构的特点)以及进栈、退栈和入队、退队时指针的变化,对于二叉树的性质和遍历规则要牢记并灵活运用。

四、程序设计基础

要深刻理解面向过程的程序设计方法和面向对象的程序设计方法。前者主要了解其设计原则,后者是本章重点,要理解并掌握一些基本概念和术语,例如对象(类的实例)及其特点、属性、方法、事件、消息、类(对象的抽象)、封装、继承、多态等。

五、软件工程基础

掌握软件工程中的一些基本概念,例如:软件的定义、特点和分类;软件危机的表现;软件工程的定义、要素、核心思想、原则和目标。然后以软件生命周期为主线,重点掌握每个阶段(需求解析、总体设计、详细设计、测试和调试)的主要任务、方法和常用工具,该知识点是本章的重点,建议以表格的形式进行归纳和总结,以明确各个知识点的联系,以有助于记忆。在归纳总结过程中,要注意以下知识点的区别,例如:测试(发现错误;本阶段涵盖整个软件生命周期的过程)、调试(诊断并改正程序中的错误,主要在开发阶段)的区别;白盒测试(内部的结构测试)和黑盒测试(外部接口的功能测试)的区别;每个阶段所用到的工具(DFD、DD、SC、PFD、N-S 等)的区别。

六、数据库设计基础

主要知识点有数据库系统组成、发展、特点、内部体系结构;数据模型;实体联系模型;关系模型、关系运算和数据库设计与管理,其中数据库系统、关系模型和关系运算是考试重点。

第一章 数据结构与算法

1.1 算法

算法:是一组有穷指令集,是解题方案的准确而完整的描述。通俗地说,算法就是计算机解题的过程。算法不等于程序,也不等于计算方法,程序的编制不可能优于算法的设计。

算法是一组严谨地定义运算顺序的规则,每一个规则都是有效的,且是明确的,此顺序将在有限的次数下终止。所以其四个基本特征包括:

- (1)确定性,算法中每一步骤都必须有明确定义,不允许有模棱两可的解释或多义性;
- (2)有穷性,算法必须能在有限的时间内做完,即能在执行有限个步骤后终止;
- (3)可行性,算法原则上能够精确地执行;
- (4)拥有足够的情报。

算法的基本要素包括:一是对数据对象的运算和操作;二是算法的控制结构。基本运算和操作包括:算术运算、逻辑运算、关系运算、数据传输;算法的三种基本控制结构:顺序结构、选择结构、循环结构。

算法基本设计方法:列举法、归纳法、递推、递归、减半递推技术、回溯法。

算法效率的度量——算法复杂度:算法时间复杂度和算法空间复杂度。算法时间复杂度:指执行算法所需要的计算工作量,即算法执行过程中所需要的基本运算次数。通常,一个算法所用的时间包括编译时间和运行时间,算法空间复杂度:指执行这个算法所需要的内存空间。包括算法程序所占的空间,输入的初始数据所占的空间,算法执行过程中所需的额外空间。

指令系统:一个计算机系统能执行的所有指令的集合。

1.2 数据结构基础

数据处理是指对数据集中的各元素以各种方式进行运算,包括插入、删除、查找、更改等运算,也包括对数据元素进行解析。

数据结构:指相互有关联的数据元素的集合。

数据结构研究的三个方面:

- (1)数据集中各数据元素之间所固有的逻辑关系,即数据的逻辑结构;
- (2)在对数据进行处理时,各数据元素在计算机中的存储关系,即数据的存储结构;
- (3)对各种数据结构进行的运算。

数据的逻辑结构应包含:

- (1)表示数据元素的信息;
- (2)表示各数据元素之间的前后件关系(指逻辑关系,与存储位置无关)。即一个数据结构可以表示成

$B=(D,R)$ 其中 B 表示数据结构。为了反映 D 中各数据元素之间的前后件关系,一般用二元组来表示。

数据的逻辑结构在计算机存储空间中的存放形式称为数据的存储结构,也称数据物理结构。数据的存储结构有顺序、链接、索引等。

一个数据结构除了用二元关系表示外,还可以直观地用图形表示。在数据结构的图形表示中,对于数据集合 D 中的每一个数据元素用中间标有元素值的方框表示,一般称之为数据结点,并简称为结点;为了进一步表示各数据元素之间的前后件关系,对于关系 R 中的每一个二元组,用一条有向线段从前件结点指向后件结点。例如,一年四季的数据结构可以用如图 1.1 所示的图形来表示。又如,反映家庭成员间辈分关系的数据结构可以用如图 1.2 所示的图形表示。



图 1.1 一年四季数据结构的图形表示

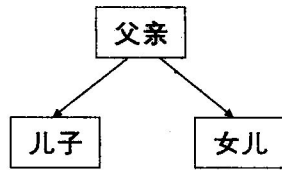


图 1.2 家庭成员间辈分关系数据结构的图形表示

根据数据结构中各数据元素之间前后件关系的复杂程度,一般将数据结构分为两大类型:线性结构与非线性结构。

如果一个非空的数据结构满足下列两个条件:

- (1)有且只有一个根结点;
- (2)每一个结点最多有一个前件,也最多有一个后件。

则称该数据结构为线性结构。线性结构又称线性表。

1.3 线性表及存储结构

线性表是由 $n(n \geq 0)$ 个数据元素 a_1, a_2, \dots, a_n 组成的一个有限序列,表中的每一个数据元素,除了第一个外,有且只有一个前件,除了最后一个外,有且只有一个后件。即线性表或是一个空表,或可以表示为

$$(a_1, a_2, \dots, a_i, \dots, a_n)$$

其中 $a_i(i=1, 2, \dots, n)$ 是属于数据对象的元素,通常也称其为线性表中的一个结点。

显然,线性表是一种线性结构。数据元素在线性表中的位置只取决于它们自己的序号,即数据元素之间的相对位置是线性的。

非空线性表有如下一些结构特征:

- (1)有且只有一个根结点 a_1 ,它无前件;
- (2)有且只有一个终端结点 a_n ,它无后件;
- (3)除根结点与终端结点外,其他所有结点有且只有一个前件,也有且只有一个后件。线性表中结点的个数 n 称为线性表的长度。当 $n=0$ 时,称为空表。

线性表由一组数据元素构成。数据元素的含义很广泛,在不同的具体情况下,它可以有不同的含义。例如,一年中的四个季节(春,夏,秋,冬)是一个长度为4的线性表,其中的每一个季节名就是一个数据元素。数据元素的位置只取决于自己的序号,元素之间的相对位置是线性的。在复杂线性表中,由若干项数据元素组成的数据元素称为记录;由多个记录构成的线性表称为文件。

线性表的顺序存储结构具有以下两个基本特点:

- (1)线性表中所有元素所占的存储空间是连续的;
- (2)线性表中各数据元素在存储空间中是按逻辑顺序依次存放的。

元素 a_i 的存储地址为: $ADR(a_i) = ADR(a_1) + (i-1)k$

$ADR(a_1)$ 为第一个元素的地址, k 代表每个元素占的字节数。

顺序表的运算:查找、插入、删除。

1.4 线性链表

线性表的链式存储结构称为线性链表。

为了适应线性表的链式存储结构,计算机存储空间被划分为若干个小块,每一小块占若干字节,通常称这些小块为存储结点。

为了存储线性表中的每一个元素,一方面要存储数据元素的值,另一方面要存储各数据元素之间的前后件关系。为此目的,将存储空间中的每一个存储结点分为两部分:一部分用于存储数据元素的值,称为数据域;另一部分用于存放下一个数据元素的存储序号(即存储结点的地址),即指向后件结点,称为指针域。由此可知,在线性链表中,存储空间的结构如图 1.3 所示。

存储序号	数据域	指针域
1		
2		
⋮		
i		
⋮		
m		

图 1.3 线性链表存储结构

在线性链表中,用一个专门的指针 HEAD 指向线性链表中第一个数据元素的结点(即存放线性表中第一个数据元素的存储结点的序号)。线性表中最后一个元素没有后件,因此,线性链表中最后一个结点的指针域为空(用 NULL 或 0 表示),表示链表终止。线性链表的逻辑结构如图 1.4 所示。

存储序号	数据域	指针域
i	$V(i)$	$NEXT(i)$

图 1.4 链表的逻辑结构