

计算机等级考试丛书('98 大纲)

谭浩强 主编

C 程序设计(二级)教程

谭浩强 薛淑斌 崔武子 编著

清华大学出版社

(京)新登字 158 号

内 容 简 介

本书是根据教育部考试中心 1998 年制定的《全国计算机等级考试 考试大纲》中关于 C 语言程序设计的考试要求而编写的。内容包括:C 语言的基本概念、C 语言的语法规则以及利用 C 语言进行程序设计的有关知识。

本书的内容覆盖了考试大纲所要求的范围,系统而简明地介绍了考生应该掌握的内容。每章之后均附有精选的练习题,以帮助读者学习和掌握。最后还附有考试大纲和一份笔试模拟试题。

本书可以作为参加计算机等级考试人员的自学或培训教材,也可以作为大专院校 C 语言程序设计课程的教学用书。

版权所有,翻印必究。

本书封面贴有清华大学出版社激光防伪标签,无标签者不得销售。

书 名: C 程序设计(二级)教程

作 者: 谭浩强 等

出版者: 清华大学出版社(北京清华大学学研楼,邮编 100084)

印刷者:

发行者: 新华书店总店北京发行所

开 本: 787 × 1092 1/16 印张:18.25 字数:415 千字

版 次: 2000 年 3 月 第 1 版 2000 年 3 月 第 1 次印刷

书 号: ISBN 7-302-01166-4/ TP·432

印 数: 0001 ~ 6000

定 价: 22.00 元

计算机等级考试丛书('98 大纲)

序

进入 20 世纪 90 年代以来,我国掀起了第二次计算机普及高潮。人们已经认识到,要建设一个现代国家,离开计算机将寸步难行;无论从事什么工作,都必须学习计算机知识,掌握计算机应用。不少单位已经把通过计算机考试作为任职条件。

为了适应社会的需要,原国家教委考试中心于 1994 年推出了“全国计算机等级考试”,受到社会各界的热烈欢迎。至 1998 年底,已有 160 多万人报名考试,其中 70 多万人获得了等级证书。全国许多地区和部门也组织了本地区或本系统的计算机统一测试。许多同志认为,“学历”是从整体上反映了一个人的知识水平,而“证书”则反映了一个人在某一方面的能力。证书制度是学历制度的必要补充,是人才市场的需要,因而受到各方面的欢迎。

当然,计算机统一考试无论从内容上还是形式上都有一些问题需要进一步探索,例如,如何更好地反映计算机新技术的发展,怎样才能测试出应试者的实际应用能力等。相信随着时间的推移和经验的累积,这项制度会日趋完善。

全国和地区性的计算机等级考试在内容上、分级上、考试形式上大体相似。全国计算机等级考试目前分为四个等级。一级:具有计算机的初步知识和使用微机系统的初步能力。一级(B):面向公务员和在职干部,要求大致与一级相当,侧重应用能力。二级:具有计算机软件、硬件的基础知识和使用一种高级语言编制程序、上机调试的能力,可以从 QBASIC, FORTRAN, PASCAL, C, FoxBASE 五种语言中任选一种应试。三级分为两类:三级(A):具有计算机应用基础知识和计算机硬件系统开发的初步能力;三级(B):具有计算机应用基础知识和软件系统开发的初步能力。四级:具备深入而系统的计算机知识和较高的计算机应用能力。

为了帮助广大读者准备参加等级考试,1995 年我们组织编写了“计算机等级考试辅导”丛书,由清华大学出版社出版。该丛书包括一级、一级 B、二级(五种)的考试辅导共 8 本书,在出版后三年多时间内已发行 120 多万册。

根据计算机应用的发展,1998 年教育部考试中心修改了全国计算机等级考试大纲。和原大纲相比,新大纲有以下改动:一级除了可以选择 DOS 环境应试外,还可以选择 Windows 环境应试;二级的 BASIC 语言改为 QBASIC 语言;在二级的计算机基础知识部分中增加了对网络和多媒体的要求。

根据广大读者的要求,我们在原来的“计算机等级考试辅导”丛书的基础上,组织出版这套“计算机等级考试丛书('98 大纲)”。该丛书由以下三个系列构成:(1)计算机等级考试教程:全面而系统地介绍考试大纲所规定的内容,是便于自学的教材。(2)计算机等级考试辅导:用来帮助已学过该课程的读者复习和准备考试,每本书的内容均包括各章要点、各章难点、例题分析、思考题,并附有模拟试题。(3)计算机等级考试样题汇编:按照计

计算机等级考试的内容和试题形式,提供了 500 ~ 600 道样题,供应试者准备考试参考。

本丛中各书的作者都是高等学校或计算机应用部门中具有丰富教学经验并对计算机等级考试有较深入研究的教授、专家。相信该丛书的出版一定会受到广大准备参加计算机等级考试的读者的欢迎。

欢迎读者对本丛书提出宝贵意见。

“计算机等级考试丛书('98 大纲)”主编

全国计算机等级考试委员会副主任

谭浩强

1999 年 3 月

前 言

全国计算机等级考试第二级为程序设计。要求考生掌握计算机的基础知识,并且具有用一种高级语言编写程序的能力。C语言是二级考试大纲规定的五种语言之一。

C语言是近年来国内外得到迅速推广使用的一种现代计算机语言,深受广大计算机应用人员的喜爱。现在愈来愈多的人希望学习和使用C语言。C语言功能强大,使用灵活,既具有高级语言的特点,又具有低级语言的某些特点;它既用于编写系统软件,又可用于编写应用软件。掌握C语言是软件开发人员的一项基本功。

学习C语言,最好有一种计算机高级语言的基础。在编写本书时,考虑到大多数学习C语言的读者已学过一种计算机高级语言,因此在学习其它高级语言时已经接触过的一些内容,本书不再重复详述。此外,本书的内容是紧扣教育部考试中心1998年修订颁布的《全国计算机等级考试考试大纲》(二级C语言程序设计)编写的。凡考试大纲中不包括的部分,本书也忍痛割爱。在本书中,全面而扼要地介绍了C语言的语法规则和在使用中应注意的问题,并提供了较多的例题,供读者分析。在每一章的后面都提供了一定数量的习题供读者练习和自我检查。题目的类型包括选择题、填空题和编程题。在本书的附录中提供了各章习题的全部参考答案。这些习题和解答(尤其是编程题的解答)可以作为例题来分析和学习,以加深对C语言的理解和掌握。请读者一定要仔细阅读程序,分析算法,尤其要多做练习,上机调试程序,决不要满足于“知道了”语法规则,更不应死记硬背某些现成结论。

考虑到统一阅卷的条件,计算机等级考试的笔试题采用了标准题形式(选择题和填空题),此外,还有上机题。但是读者千万不要满足于会答选择题和填空题。学习程序设计,衡量学得好坏的标准是能否编写和调试程序。学习程序设计,一定要学活用活,在使用上多下功夫。

二级考试包括计算机基本知识和计算机语言程序设计两个部分。本书只包括C语言程序设计的内容,不包括计算机基本知识的内容,有关计算机基本知识的内容可参阅本丛书中的《计算机基本知识(二级)教程》一书。

本书是根据目前在国内使用最广泛的《C程序设计》(谭浩强编著,清华大学出版社出版)一书结合等级考试的特点改写而成的。参加编写的人员有谭浩强、薛淑斌、崔武子。在学习本书时,如有不清楚的问题,可以参阅《C程序设计》(第二版)一书,该书有更详尽的说明。

本书适用于准备参加全国计算机等级考试和各地区、各部门组织的计算机统一考试,这些考试的要求、内容和形式是大同小异的。

由于时间仓促,加之水平有限,有不足之处,敬请批评指正。

谭浩强

1999年11月于北京

目 录

第 1 章 C 语言的初步知识	1
1.1 C 程序的构成	1
1.2 函数的构成	2
1.3 源程序的书写格式和程序风格	2
习题	3
第 2 章 数据类型及其运算	5
2.1 C 语言的数据类型	5
2.2 常量与变量	5
2.2.1 常量和符号常量	5
2.2.2 变量	6
2.2.3 整型数据	6
2.2.4 实型数据	8
2.2.5 字符型数据	9
2.3 数值型数据间的混合运算	10
2.4 C 语言的运算符和表达式	11
2.4.1 C 语言运算符	11
2.4.2 C 语言表达式的类型	12
2.4.3 表达式的求值规则	12
2.4.4 算术表达式	12
2.4.5 赋值表达式	13
2.4.6 逗号表达式	16
2.4.7 关系表达式	17
2.4.8 逻辑表达式	18
2.4.9 条件表达式	19
2.5 程序举例	20
习题	21
第 3 章 基本语句	24
3.1 C 语句概述	24
3.2 赋值语句	25
3.3 数据输入与输出	25
3.3.1 字符输入输出函数	26
3.3.2 格式输出函数	27
3.3.3 格式输入函数	30

3.4	程序举例.....	33
	习题	34
第4章	控制语句	39
4.1	if 语句	39
4.1.1	if 语句的三种形式	39
4.1.2	if 语句的嵌套	41
4.1.3	用 if 语句实现选择结构	42
4.2	switch 语句	43
4.3	goto 语句	46
4.4	while 和 do while 语句	46
4.4.1	while 语句	46
4.4.2	do-while 语句	47
4.4.3	while 和 do-while 循环的比较	47
4.5	for 语句	48
4.6	break 语句和 continue 语句	51
4.6.1	break 语句	51
4.6.2	continue 语句	51
4.7	循环的嵌套	52
4.8	程序举例	53
	习题	61
第5章	数组	68
5.1	数组的定义和引用	68
5.1.1	数组的定义	68
5.1.2	数组的初始化	69
5.1.3	数组元素的引用	71
5.2	数组应用举例	71
5.3	字符数组	79
5.3.1	字符数组的定义	79
5.3.2	字符数组的初始化	79
5.3.3	字符串结束标志	80
5.3.4	字符数组的输入输出	81
5.3.5	字符串处理函数	82
5.3.6	程序举例	85
	习题	89
第6章	函数	95
6.1	概述	95
6.2	函数定义的一般形式	96
6.3	函数参数和函数的值	97

6.3.1	形式参数和实际参数	97
6.3.2	函数的返回值	98
6.4	函数的调用	99
6.4.1	函数调用的一般形式	99
6.4.2	函数调用的方式	99
6.4.3	对被调用函数的声明	100
6.4.4	程序举例	102
6.5	函数的嵌套调用	104
6.6	函数的递归调用	107
6.7	局部变量和全局变量	109
6.7.1	局部变量	109
6.7.2	全局变量	110
6.8	动态存储变量与静态存储变量	111
6.8.1	变量的存储类别	111
6.8.2	局部变量的存储方式	112
6.8.3	全局变量的存储方式	115
6.8.4	存储类别小结	115
6.9	内部函数和外部函数	116
6.9.1	内部函数	117
6.9.2	外部函数	117
	习题	117
第7章	编译预处理	122
7.1	宏定义	122
7.1.1	不带参数的宏定义	122
7.1.2	带参数的宏定义	123
7.2	“文件包含”处理	126
	习题	127
第8章	指针	132
8.1	指针的概念	132
8.2	变量的指针和指向变量的指针变量	132
8.2.1	指针变量的定义	133
8.2.2	指针变量的引用	133
8.2.3	指针变量作为函数参数	135
8.3	数组的指针和指向数组的指针变量	136
8.3.1	指向数组元素的指针变量的定义与赋值	137
8.3.2	通过指针引用数组元素	137
8.3.3	数组名作函数参数	140
8.3.4	指向多维数组的指针和指针变量	144

8.4	字符串的指针和指向字符串的指针变量	145
8.4.1	字符串的表示形式	145
8.4.2	字符指针变量与字符数组	147
8.4.3	字符串指针作函数参数	149
8.5	函数的指针和指向函数的指针变量	152
8.5.1	用函数指针变量调用函数	152
8.5.2	把指向函数的指针变量作函数参数	154
8.6	返回指针值的函数	154
8.7	指针数组和指向指针的指针	155
8.7.1	指针数组的概念	155
8.7.2	指向指针的指针	156
8.7.3	main 函数的命令行参数	157
	习题	158
第 9 章	结构体与共用体	163
9.1	概述	163
9.2	定义结构体类型变量的方法	163
9.3	结构体类型变量的引用	165
9.4	结构体数组	167
9.5	指向结构体类型数据的指针	169
9.5.1	指向结构体变量的指针	169
9.5.2	指向结构体数组的指针	170
9.5.3	用指向结构体的指针作函数参数	171
9.6	用指针处理链表	172
9.6.1	链表概述	172
9.6.2	用 typedef 定义类型	173
9.6.3	链表的建立、输出、插入和删除	174
9.7	共用体	181
9.7.1	共用体的概念	181
9.7.2	共用体变量的引用方式	182
9.7.3	共用体类型数据的特点	182
	习题	183
第 10 章	位运算	189
10.1	“按位与”运算	189
10.2	“按位或”运算	189
10.3	“异或”运算	190
10.4	“取反”运算	190
10.5	左移运算	190
10.6	右移运算	191

10.7	位运算复合赋值运算.....	191
10.8	不同长度数据的位运算.....	192
	习题.....	192
第 11 章	文件	194
11.1	C 文件概述	194
11.2	文件类型指针.....	194
11.3	文件的打开与关闭.....	195
11.3.1	文件的打开(fopen 函数)	195
11.3.2	文件的关闭(close 函数)	196
11.4	文件的读写.....	196
11.4.1	fputc 函数和 fgetc 函数(putc 函数和 getc 函数).....	196
11.4.2	fputs 函数和 fgets 函数	198
11.4.3	fread 函数和 fwrite 函数	199
11.4.4	fprintf 函数和 fscanf 函数	200
11.5	文件的定位.....	200
11.5.1	rewind 函数	200
11.5.2	fseek 函数和随机读写	201
	习题.....	202
附录	206
附录 1	常用字符与 ASCII 代码对照表	206
附录 2	C 语言中的关键字	207
附录 3	运算符和结合性	207
附录 4	C 语言常用语法提要	208
附录 5	C 库函数	212
附录 6	全国计算机等级考试二级(基础知识和 C 语言程序设计)考试大纲	219
附录 7	全国计算机等级考试二级(基础知识和 C 语言程序设计)笔试样题	222
附录 8	本书各章习题参考答案	235
参考文献	280

第 1 章 C 语言的初步知识

1.1 C 程序的构成

下面是一个简单的 C 程序,它包含两个函数。

[例 1.1] 简单的 C 程序。

```
# include <stdio.h>          /* “文件包含”命令,它是一条预编译命令 */
main ( )                    /* 主函数 */
{ int a,b,c;                /* 定义变量 */
  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 */
}

int max (int x,int y) /* 定义 max 函数,函数值为整型,x、y 为形式参数,是整型 */
{ int z;              /* max 函数中用到的变量 z,也要加以定义 */
  if (x > y)
    z = x;
  else
    z = y;           /* 找出 x、y 中的最大者后存放 z 中 */
  return (z);       /* 将 z 的值返回,通过 max 带回调用处 */
}
```

说明:

(1) 一个 C 程序由一个或若干个函数构成。程序中至少应包含一个 main 函数。函数是 C 程序的基本单位。C 程序中的函数相当于其它语言中的子程序。用函数来实现特定的功能。因此,可以说 C 语言是函数式的语言,C 语言的这种特点容易实现程序的模块化。

(2) 一个 C 程序总是从 main 函数开始执行,而不论 main 函数在整个程序中的位置如何(main 函数可以放在程序最前头,也可以放在程序最后,或放在一些函数之前而放在另一些函数之后)。

(3) 被某一函数调用的函数可以是系统提供的库函数(例如 printf 和 scanf 函数),也可以是用户根据需要自己编制设计的函数(例如例 1.1 中的 max 函数)。

(4) 如果调用库函数,则要用 include 命令行,以向系统提供必要的信息(如例 1.1 中的第一行)。有关命令行将在第 7 章中详细介绍。

(5) 一个 C 程序可以由一个文件组成,也可以由若干个文件组成。例 1.1 的程序是

由一个文件组成的,在一个文件中包含两个函数。也可以将两个函数分别放在两个文件中(每一个文件有一个文件名),分别进行编译,然后通过联接(Link)把它们合并成一个可供执行的二进制目标文件,以供运行,图 1.1 表示 C 程序的结构。

图 1.1 C 程序的结构

1 2 函数的构成

一个函数由两部分组成:

(1) 函数的首部。它包括函数名、函数类型、函数属性、函数参数名、参数类型。

例 1.1 中的 max 函数的首部为:

一个函数名后面必须跟一对圆括弧,函数参数可以没有,但一对圆括弧不能省略,如 main()。

(2) 函数体。即函数首部下面的大括弧{ }内的部分。如果一个函数内有多个大括弧,则最外层的一对{ }为函数体的范围。

函数体一般包括数据声明部分和执行语句部分。

数据声明部分,对本函数中用到的变量或数组定义其数据类型。如例 1.1 中 main 函数中的“int a, b, c”和 max 函数中的“int z”,指定变量 a、b、c 和 z 为整型变量。

执行语句部分,由若干语句组成。每个语句的最后必须有一个分号,分号是 C 语句的必要组成部分。例如:

```
c = a + b;
```

分号不可少,即使是程序中最后一个语句也应包含分号。

1.3 源程序的书写格式和程序风格

源程序的书写格式和程序风格通常体现在以下几点:

(1) 函数的首部,后面不要加分号(见例 1.1 main 函数和 max 函数的第一行最后都没有分号)。

(2) C 程序书写格式自由,一行内可以写几个语句,一个语句可以分写在多行上。为清晰起见,建议一般在一行内写一个语句。

(3) 为了清晰地表现出程序的结构,建议采用锯齿形程序格式,例如将 if 语句的内嵌语句和循环结构中的循环体在书写时向右缩进几列。如多层嵌套则多层缩进。

(4) 可以用 `* */` 对 C 程序中的任何部分作注释。一个好的、有使用价值的源程序都应当加上必要的注释,以增加程序的可读性。

习 题

一、选择题

1.1 以下不正确的概念是

- A) 一个 C 程序由一个或多个函数组成
- B) 一个 C 程序必须包含一个 main 函数
- C) 在 C 程序中,可以只包括一条语句
- D) C 程序的每行上可写多条语句

1.2 下述源程序的书写格式不正确的是

- A) 一条语句可以写在几行上
- B) 一行上可以写几条语句
- C) 分号是语句的一部分
- D) 函数的首部,其后必须加分号

1.3 在 C 语言程序中

- A) main 函数必须放在程序的开始位置
- B) main 函数可以放在程序的任何位置
- C) main 函数必须放置在程序的最后
- D) main 函数只能出现在库函数之后

1.4 以下能正确构成 C 语言程序的是

- A) 一个或若干个函数,其中 main 函数是可选的
- B) 一个或若干个函数,其中至少应包含一个 main 函数
- C) 一个或若干个子程序,其中包括一个主程序
- D) 由若干个过程组成

第 2 章 数据类型及其运算

2.1 C 语言的数据类型

C 语言提供的数据类型如下:

	整型	
	字符型	
基本类型	实型(浮点型)	单精度型 双精度型
数据类型	枚举类型	
	数组类型	
	构造类型	结构体类型 共用体类型
	指针类型	
	空类型	

C 语言中的数据分为常量和变量,它们都有类型的区分。

2.2 常量与变量

2.2.1 常量和符号常量

在程序运行过程中其值不能被改变的量称为常量,常量区分为不同的类型,如 12、0、-3 为整型常量,4.6、-1.23 为实型常量, a、d 为字符型常量。常量一般从其字面形式即可判别。

也可以用一个标识符代表一个常量,如可以在一个文件的开头写以下一个命令行:

```
# define PRICE 30
```

用 # define 命令行定义 PRICE 代表常量 30,此后凡在该文件中出现的 PRICE 都代表 30,可以和常量一样进行运算。

这种用一个指定的名字代表一个常量的称为符号常量,即带名字的常量,注意,符号常量不同于变量,它的值在其作用域(一般说是在本文件范围内)内不能改变,也不能再被赋值。

习惯上,符号常量名用大写,变量名用小写,以示区别。

2 2 2 变量

在程序运行过程中其值可以改变的量称为变量。程序中的变量由用户取名。和其它高级语言一样,用来标识变量名、符号常量名、函数名、数组名、类型名、文件名的有效字符序列称为标识符(identifier)。简单地说,标识符就是一个名字。

C语言规定标识符只能由字母、数字和下划线三种字符组成,且第一个字符必须为字母或下划线。下面是合法的标识符,也是合法的变量名:

```
sum, average, class, day, month, student_name, _above, lotus_1_2_3, basic
```

下面是不合法的标识符和变量名:

```
M.D John, $123, #33, 3D64, a > b
```

注意,大写字母和小写字母被认为是两个不同的字符,因此 sum 和 SUM 是两个不同的变量名。习惯上,为增加可读性,变量名用小写字母表示。

C语言中标识符的长度(字符个数)无统一规定,随系统的不同而不同。许多系统(如IBM PC的MS C)取8个字符,假如程序中出现的变量名长度大于8个字符,则只有前面8个字符有效,后面的不被识别。例如,有两个变量: student_name 和 student_number,由于二者的前8个字符相同,系统认为这两个变量是一回事而不加区别。

在程序中,一个变量实质上代表某个存储单元。要注意变量的“名”和变量的“值”的区别,变量的“名”是指该变量所代表的存储单元的标志,而变量的“值”是指存储单元中的内容。

2 2 3 整型数据

1. 整型常量

在C语言中整型常量有以下三种表示形式:

- (1) 十进制形式,如 112、0、-38。
- (2) 八进制形式,以数字 0 开头,如 0123 表示八进制整数 123,即十进制整数 83。
- (3) 十六进制形式,以 0x 开头。如 0x123 代表十六进制数 123,即十进制整数 291。

2. 整型变量

分为基本型、短整型、长整型、无符号型。

例如:

```
int i;           /* 定义变量 i 为整型 */
long j,k;       /* 定义变量 j 和 k 为长整型 */
```

C标准没有具体规定以上各类数据所占内存字节数,各种机器处理上有所不同,一般以一个机器字(word)存放一个 int 型数据,而 long 型数据的字节数应不小于 int 型,short 型不长于 int 型。常用的 Turbo C 对各类型数据的设定见表 2.1。

表 2.1 Turbo C 对各类型整数的设定

类 型	类型标识符	在内存中所占字节数	数 值 范 围
基本类型	int	2	- 32768 ~ 32767 即 $- 2^{15} \sim - (2^{15} - 1)$
短整型	short[int]	2	- 32768 ~ 32767 即 $- 2^{15} \sim - (2^{15} - 1)$
长整型	long[int]	4	- 2147483648 ~ 2147483647 即 $- 2^{31} \sim - (2^{31} - 1)$
无符号整型	unsigned[int]	2	0 ~ 65535 即 $0 \sim 2^{16} - 1$
无符号短整型	unsigned short	2	0 ~ 65535 即 $0 \sim 2^{16} - 1$
无符号长整型	unsigned long	4	0 ~ 4294967295 即 $0 \sim 2^{32} - 1$

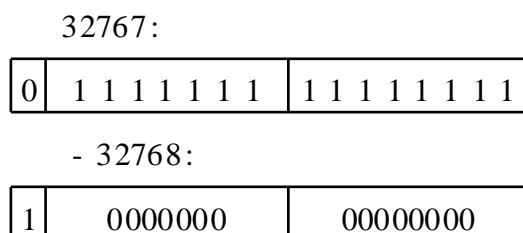
[例 2.1] 整型数据程序示例。

```
main( )
{ int a, b, c, d;
  a = 10; b = 5;
  c = a + b;
  d = 32767 + 1
  printf( "c = %d \n", c);
  printf( "d = %d " \n, d);
}
```

程序的运行结果如下：

```
c = 15
d = - 32768
```

程序中开辟了名为 a、b、c、d 的四个存储单元，它们在内存中各占两个字节。a 和 b 所代表的存储单元中存放 10 和 5，c 所代表的存储单元中存放 a 和 b 中值的和 15。d 中存放 32767 与 1 的和，由于受到 int 型变量取值范围的限制，d 中存放的值为 - 32768，而不是 32768。



符号位

图 2.1

从图 2.1 可以看到 32767 在内存中以二进制的表示形式，最左边一位是符号位(0 表示正, 1 表示负)。32767 再加 1 后 15 位全为 0，最左边一位为 1，它是 - 32768 的补码形式(数据在内存中都是以补码形式存放的)。有关补码的知识不必深究，不在考试范围内。我们只要知道使用某类型变量时，一定要注意该类型变量的取值范围。