

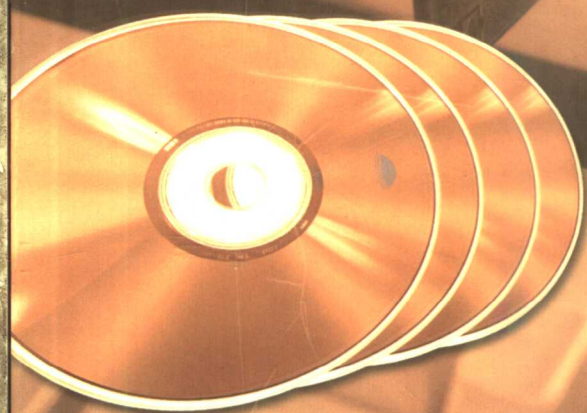
高等职业教育教材丛书

GAODENG ZHIYE
JIAOYU JIAOCAI CONGSHU

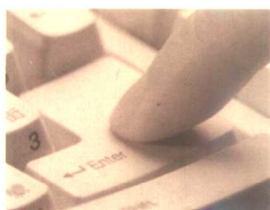
C语言与 数据结构

实习指导与模拟试题

边莫英 主编
高福成 曲建民 喻梅 赵喆 编著



.12



南开大学出版社

高等职业教育教材丛书

C 语言与数据结构

实习指导与模拟试题

边奠英

主编

高福成 曲建民 喻梅 赵喆 编著

南开大学出版社

天 津

内容提要

本书是天津市高等教育自学考试委员会组编的《C 语言程序设计》和《数据结构导论》的配套教材。依据《C 语言和数据结构自学考试大纲》的要求,参考实际考试的范围和题型,系统介绍了这两门课程的主要知识点、重点和难点,精心安排了上机实习及必要的指导,提供了大量的笔试和上机考试模拟试题,读者可将本书作为上机实习指导书和考前复习的主要参考书。

图书在版编目(CIP)数据

C 语言与数据结构实习指导与模拟试题 / 边莫英主编;
高福成等编著. —天津:南开大学出版社,2003.7
ISBN 7-310-01901-6

I. C... II. ①边...②高... III. ①C 语言—程序设计—高等学校—教学参考资料②数据结构—高等学校—教学参考资料 IV. ①TP312②TP311.12

中国版本图书馆 CIP 数据核字(2003)第 019583 号

出版发行 南开大学出版社

地址:天津市南开区卫津路 94 号 邮编:300071

营销部电话:(022)23508339 23500755

营销部传真:(022)23508542

邮购部电话:(022)23502200

出版人 肖占鹏

承印 天津宝坻第二印刷厂印刷

经销 全国各地新华书店

版次 2003 年 7 月第 1 版

印次 2003 年 7 月第 1 次印刷

开本 787mm×1092mm 1/16

印张 17.75

字数 448 千字

印数 1—5000

定价 25.00 元



前 言

本书是《C 语言程序设计》(天津市高等教育自学考试委员会组编,高福成编著,南开大学出版社出版)和《数据结构》(天津市高等教育自学考试委员会组编,陈有祺、辛运伟编著,南开大学出版社出版)的配套教材,目的是根据考试大纲的要求,指导读者有针对性地学习、上机和复习。

本书由两部分组成,第一部分为 C 语言程序设计,第二部分为数据结构,每部分各包含三编。第一编为本课程基本要求,分章节介绍本门课程的主要知识点、重点和难点,突出重点,剖析难点,使读者能从知识的层面上掌握核心内容;第二编为实习指导,将本课程的主要实习内容按学时分配和编排,每个实习都给出了实习要求、学时分配和必要的指导,以使读者能从实践的层面上提高动手能力。每个实习都安排了适量的自测题,以提高读者举一反三和灵活运用所学知识的能力;第三编为模拟试题,按实际考题的题型和难度,给出了大量的具有实战价值的模拟试题,以帮助读者了解和掌握考试的基本题型和要求;可作为阶段性复习和考前自我检测之用。

本书 C 语言程序设计部分的第一编和第二编由高福成编写,第三编由喻梅编写;数据结构部分第一编由曲建民编写,第二编和第三编由赵喆编写。

由于作者水平有限,书中肯定存在不少疏漏和错误,敬请读者批评指正。

编著者 谨识

2002 年 11 月 30 日

作者 E-mail 地址: gfceyou@eyou.com

目 录

第一部分 C 语言程序设计

第一编 本课程基本要求	3
第 1 章 C 语言概述	3
1.1 本章知识点	3
1.2 本章学习重点	3
1.3 本章主要难点	4
1.4 本章主要考点	4
第 2 章 基本数据类型、常量和变量	5
2.1 本章知识点	5
2.2 本章学习重点	5
2.3 本章主要难点	7
2.4 本章主要考点	7
第 3 章 数据运算	8
3.1 本章知识点	8
3.2 本章学习重点	8
3.3 本章主要难点	12
3.4 本章主要考点	13
第 4 章 流程控制	14
4.1 本章知识点	14
4.2 本章学习重点	14
4.3 本章主要难点	18
4.4 本章主要考点	21
第 5 章 数组和字符串	22
5.1 本章知识点	22
5.2 本章学习重点	22
5.3 本章主要难点	24
5.4 本章主要考点	26
第 6 章 指针	27
6.1 本章知识点	27
6.2 本章学习重点	27
6.3 本章主要难点	31
6.4 本章主要考点	34

第7章 函数	35
7.1 本章知识点	35
7.2 本章学习重点	35
7.3 本章主要难点	42
7.4 本章主要考点	46
第8章 结构、联合和枚举	47
8.1 本章知识点	47
8.2 本章学习重点	47
8.3 本章主要难点	50
8.4 本章主要考点	52
第9章 文件	53
9.1 本章知识点	53
9.2 本章学习重点	53
9.3 本章主要难点	55
9.4 本章主要考点	56
第10章 编译预处理和分割编译	57
10.1 本章知识点	57
10.2 本章学习重点	57
10.3 本章主要难点	58
10.4 本章主要考点	59
第二编 实习指导	60
第1章 简单的C程序设计	60
第2章 基本数据类型及其运算	64
第3章 选择结构程序设计	67
第4章 循环结构程序设计	70
第5章 数组和字符串	72
第6章 指针	74
第7章 函数	77
第8章 复合数据类型	83
第9章 文件	87
第10章 编译预处理	91
第三编 模拟试题	93
第1章 C语言概述	93
第2章 基本数据类型、常量和变量	95
第3章 数据运算	98
第4章 流程控制	103
第5章 数组和字符串	117
第6章 指针	129
第7章 函数	140

第 8 章	结构、联合和枚举·····	155
第 9 章	文件·····	168
第 10 章	编译预处理和分割编译·····	173

第二部分 数据结构

第一编	本课程基本要求·····	181
第 1 章	绪论·····	181
1.1	本章知识点·····	181
1.2	本章学习重点·····	182
1.3	本章主要难点·····	182
1.4	本章主要考点·····	182
第 2 章	线性表·····	183
2.1	本章知识点·····	183
2.2	本章学习重点·····	186
2.3	本章主要难点·····	186
2.4	本章主要考点·····	186
第 3 章	栈、队列和数组·····	187
3.1	本章知识点·····	187
3.2	本章学习重点·····	194
3.3	本章主要难点·····	194
3.4	本章主要考点·····	194
第 4 章	树·····	195
4.1	本章知识点·····	195
4.2	本章学习重点·····	200
4.3	本章主要难点·····	200
4.4	本章主要考点·····	200
第 5 章	图·····	201
5.1	本章知识点·····	201
5.2	本章学习重点·····	202
5.3	本章主要难点·····	202
5.4	本章主要考点·····	202
第二编	实习指导·····	203
第 1 章	线性表·····	203
第 2 章	栈、队列和数组·····	211
第 3 章	树·····	216
第 4 章	图·····	218
第三编	模拟试题·····	220
第 1 章	绪论·····	220

第 2 章	线性表·····	223
第 3 章	栈、队列和数组·····	229
第 4 章	树·····	237
第 5 章	图·····	243
附录一	C 语言程序设计上机实习	
	自测题参考答案·····	245
附录二	C 语言程序设计模拟试题	
	参考答案·····	252
附录三	数据结构模拟试题	
	参考答案·····	260

第一部分 C 语言程序设计

第一编 本课程基本要求

第 1 章 C 语言概述

1.1 本章知识点

C 程序的基本结构；C 语言关键字；C 语言标识符；C 语句（表达式语句、复合语句、空语句）的概念；赋值语句和格式输入/输出函数（printf()和 scanf()）的简单使用。

1.2 本章学习重点

1. C 程序的构成

1) C 程序的基本结构是函数，函数是完成某个整体功能的最小单位。一个 C 程序由一个或多个 C 函数组成，其中必须有且只能有一个用“main”命名的主函数，其他函数由用户自行命名。

2) C 函数由函数名、形式参数和函数体三部分组成，其一般格式为

函数名([形式参数])

{

函数体

}

函数名代表该函数在内存中的首地址。因此，每个函数都有自己的函数名。

形式参数(简称形参)用于进行函数间的数据传递。当不需要进行数据传递时，形参可以省略。

函数体是函数的主体，从左花括号开始，到与之匹配的右花括号结束。函数体主要有两大部分，第一部分是本函数内部用到的局部变量定义，第二部分是完成本函数功能的语句序列。

3) main()函数可以出现在程序的任何位置上，但 C 程序执行时，总是从 main()函数开始。

2. C 关键字、标识符和 C 语句

1) 关键字是 C 语言中有特定意义和用途、不得另作他用的字符序列。ANSI C 标准规定 C 语言关键字共 32 个，用作数据类型、存储类型、类型修饰、流程控制等。所有的 C 关键字都必须小写，只用作 C 命令，不能作其他用途。

2) 标识符是 C 语言中用来表示变量名、数组名、函数名、指针名、结构体名、联合体名、

枚举常数名、用户定义的数据类型名及语句标号等用途的字符序列。标识符以字母或下划线开头、后跟字母、数字或下划线，长度不超过 32 个字符。标识符不能与 C 关键字相同，并区分大小写。

3) C 语句是组成 C 程序的基本单位，具有独立的程序功能，包括简单语句、复合语句和空语句。所有的 C 语句都以分号结尾。

4) 赋值语句用来进行表达式的计算和存储，是 C 程序中使用频率最高的语句，其一般格式为：

`v=e;`

在使用赋值语句时要注意：赋值号左边必须是变量或数组元素，赋值号右边可以是常数、变量或表达式（包括函数）。

5) 注释不是 C 语句。注释以 “/*” 开始，而以 “*/” 结束，在 “*” 和 “/” 之间不能有空格。注释既不被编译也不被执行，使用注释主要是为了增加程序的可读性。

1.3 本章主要难点

1. C 程序的开发过程

开发一个 C 程序要经过编辑、编译、连接和运行四个阶段。编辑阶段生成源程序（扩展名为 .c），编译阶段生成目标程序（扩展名为 .obj），连接阶段生成可执行文件（扩展名为 .exe），运行可执行文件即可得到程序的执行结果。

2. Turbo C 环境的使用

Turbo C 是一个集程序编辑、编译、连接、运行功能于一体的集成工作环境。常用的组合功能键为 Alt+N（编辑新程序）、Alt+F+L（编辑已有程序）、Alt+F+S（保存源程序）、Alt+R（运行程序，包括编译、连接和运行）。在 C 程序调试中，可能出现的错误包括语法错误、逻辑错误和运行错误。编译程序可以发现语法错误，而逻辑错误和运行错误只能由用户自己发现并纠正。用简单的赋值语句和格式输入输出函数编制简单的程序，并能上机调试运行。

1.4 本章主要考点

1. 了解 C 语言的特点，掌握 C 源程序的基本结构和书写格式。

2. 能正确识别 C 语言的关键字，能正确定义标识符。

3. 能使用赋值语句和格式输入输出函数编写简单的 C 程序，熟练使用 Turbo C 环境进行 C 程序的编辑、调试和运行。

第2章 基本数据类型、常量和变量

2.1 本章知识点

五种基本数据类型；不同类型常量的表示方法；变量的定义和初始化；符号常数(宏定义及 const 修饰)。

2.2 本章学习重点

1. 数据类型

- 1) C语言提供了五种基本数据类型，即char、int、float、double和void。
- 2) 数据类型决定了内存的分配模式，而内存分配模式不是C语言本身的规定，而是由宿主主机硬件决定的。
- 3) 不同的数据类型允许的数值范围也是不同的，超出了允许的数值范围，就会得到错误的结果，C编译系统没有检查此类错误的能力。
- 4) C语言对整型和字符型不加区分，可以混合使用。

2. 常数

1) C语言中的常数有四种类型：字符型常数、整数、双精度型数和字符串常数。常数的数据类型完全由书写形式确定。

2) 字符型常数是使用单引号括起来的单个字符，不可显示字符用转义字符表示。转义字符以“\”开头，后跟一个特定的字母、一个1~3位的八进制整数或一个1~2位的十六进制整数组成。用八进制表示时，整数前面可以加、也可以不加数字0；用十六进制表示时，整数前面必须且只能加x。例如，'\n'、'\012'、'\12'或'\xa'都表示换行符。

3) 整数在计算机中是准确表示的，分为有符号数、无符号数和长整数，它们都可以用十进制、八进制或十六进制表示。用八进制表示时，第一位数字前必须加0，用十六进制表示时，第一位数字前必须加0x，无符号整数末尾必须加u或U，长整数的末尾必须加l或L。例如，123为有符号十进制整数，123u为无符号十进制整数，123L为有符号十进制长整数，123uL为无符号十进制长整数；0123为有符号八进制整数，0123u为无符号八进制整数，0123L为有符号八进制长整数，0123uL为无符号八进制长整数；0x123为有符号十六进制整数，0x123u为无符号十六进制整数，0x123L为有符号十六进制长整数，0x123uL为无符号十六进制长整数。

4) C语言将所有的实常数都视为双精度型。双精度数在计算机中是近似表示的，只能用十进制表示，包括小数形式和指数形式。用小数形式表示时，小数点不能缺少；用指数形式表示时，E前面必须有数字（带不带小数点均可），E后面必须是整数，不能带小数点。

5) 字符串常数是使用双引号括起来的0个或多个字符。字符串常数中允许出现转义字符序列。字符串常数在存储时，每个字符占一个字节，系统会自动在其末尾加上字符'\0'（ASCII

码为0的空操作字符)。

3. 变量

1) 变量代表内存中的一个位置。C 程序中的所有变量都必须先定义后使用。在函数外部定义的变量称为外部变量或全局变量，在函数内部定义的变量称为内部变量或局部变量。

2) 变量的数据类型表示该变量值的含义，即某种类型的变量只能存放同一类型的数据。变量的数据类型有char、int、float、double和void五种。其中，void只用于指针变量。变量的存储类型有auto、register、static和extern四种，它影响变量的生命期和作用域，详见第7章。

3) 在定义变量时，数据类型不能缺省，存储类型可以缺省。全局变量的缺省存储类型是extern，局部变量的缺省存储类型是auto。

4) 在定义变量的同时赋予变量初值称为变量的初始化。static和extern型变量只能用常数或常数表达式进行初始化，auto和register型变量除了可以用常数或常数表达式外，还可以用已经初始化的变量进行初始化。auto和register型变量若未初始化，其值是不定的；static和extern型变量若未初始化，其值为0。

4. 符号常数

符号常数是使用标识符表示的常量，它具有变量的外表和常量的内涵。符号常数有两种表示方法：

(1) 用#define定义。例如

```
#define X 123
```

定义了一个符号常数X，它代表123这一串字符。X称为宏名，123称为宏体。用#define定义的符号常数X没有值和类型的含义。C编译系统在对源程序进行编译前，将“123”视为一串字符替换程序中所有的X（称为宏展开），在编译中再根据上下文确定其值和类型。

(2) 用const修饰。例如

```
const float a=12.3;
```

定义了一个浮点型符号常数a，其值为12.3。用const定义的符号常数既有类型也有值的含义。

5. 格式输出函数 printf()

调用 printf()函数时，应注意以下几点：

1) 格式转换说明符与输出项必须在顺序和数据类型上一一对应和匹配。

2) 当格式转换说明符个数少于输出项个数时，多余的输出项不予输出；当格式转换说明符个数多于输出项个数时，Turbo C 规定缺少的输出项输出不定值。

3) 根据实际需要，可在格式转换说明的“%”与格式字符之间增加“m.n”进行修饰。如果输出数据的实际宽度超过 m 位，则 m 无效。如果在“m.n”前加一个“-”，表示输出时采取左对齐方式。

4) 当 printf()函数中出现多个表达式输出项，Turbo C 规定，printf 按照从右到左的顺序计算各表达式的值，然后再输出结果。

5) 用%ld、%lo、%lx、%lu 可以输出相应的十进制、八进制、十六进制和无符号长整数，用%lf、%le 可以输出小数形式和指数形式的 double 或 long double 型数。

6) printf()函数的返回值是输出的字符个数。

6. 格式输入函数

调用 scanf()函数时，应注意以下几点：

- 1) 格式转换说明符与输入项必须在顺序和数据类型上一一对应和匹配。
- 2) scanf()函数中至少必须有一个输入项, 输入项必须用地址表示。
- 3) 当一个 scanf()函数中包含若干个输入项或几个 scanf()函数连续出现时, 可以同时输入几个数据, 各数据之间只能用空格、制表符或回车键分隔, 其他符号都不能作为分隔符。
- 4) 在调用 scanf()时, 格式控制字符串中出现的格式转换说明符以外的字符, 必须在对应位置上原样输入。
- 5) 当用%d 控制 int 型变量时, 键盘输入的数据必须是整数; 当用%f 或%e 控制 float 型或 double 型变量时, 可从键盘输入整数、以小数形式或指数形式(如 2e-3、3.6e4 等)表示的实数, 函数会自动进行转换; 当用%c 控制 char 型变量时, 输入的字符不必加单引号。连续几个 char 型数据同时输入时, 不要使用分隔符。
- 6) 在百分号和控制字符之间插入一个整数可以限制从输入数据中接收的字符个数。例如,

```
int a; float b;  
scanf("%4d%5f",&a,&b);
```

- 7) 不能用%m.nf 来规定小数的位数。
- 8) 可以在“%”与格式字符之间插入一个“*”来抑制对应类型数据的读入。例如

```
int x,y;  
scanf("%d%*c%d",&x,&y);
```

- 9) scanf()函数的返回值是输入的字符个数。

2.3 本章主要难点

1. 不同类型数据所占存储空间的大小及所能表示的数据范围。
2. C 语言中有字符型常数和字符型变量, 有字符串常数, 但没有字符串变量。字符型常数或字符型变量代表单个字符, 以 ASCII 码存放; 字符串常数可含有一个或多个字符, 各个字符的 ASCII 码依次存放, 并以字符'\0'结尾。
3. C 语言中用来存放实数的变量可以是浮点型和双精度型的, 而实常数一概视为双精度型。
4. C 程序中所有的变量必须先定义后使用, 先赋值后引用。局部变量省略存储类型时默认为 auto 型, 这类变量可以用常数或已初始化的变量进行初始化; 直接引用未经初始化及赋值的 auto 型变量会得到错误结果。

2.4 本章主要考点

1. 掌握基本数据类型及类型关键字。
2. 掌握十进制整数、小数形式和指数形式的实数、字符常量及字符串常量的表示和应用。
3. 掌握变量的定义、初始化和应用。
4. 掌握符号常量的宏定义方法。
5. 能正确进行基本类型常量和变量的输入/输出。

第3章 数据运算

3.1 本章知识点

运算符及其优先级和结合性的概念；算术表达式及其求值；赋值表达式；关系表达式和逻辑表达式；逗号表达式；条件表达式；位运算；测试数据长度运算符。

3.2 本章学习重点

1. 运算符的优先级和结合性

1) C语言共有42个运算符（连同括号共43个），按其连接的运算量个数可分为一元运算符、二元运算符和三元运算符，它们实现数据访问、算术运算、关系运算、逻辑运算、位操作、条件运算、赋值运算和逗号运算等功能。

2) 运算符的优先级指的是不同的运算符同时出现在同一个表达式中时，进行运算的优先次序；结合性指的是同一优先级的运算符同时出现时运算的优先次序。运算符的优先级和结合性对书写正确的表达式以及对表达式进行正确的运算十分重要。

3) 运算符优先级共15级，从高到低大体可分为以下四块：

(1) 访问数据运算符和括号运算符，即[]、.、->和()共四种。

(2) 所有的一元运算符。

(3) 算术运算符、关系运算符和逻辑运算符（其中“逻辑与”优先于“逻辑或”，“逻辑非”是一元运算符。

(4) 条件运算符、赋值运算符和逗号运算符。

4) 运算符的结合性遵循以下规则：

一元运算符、三元运算符和赋值运算符均为自右到左。

二元运算符（赋值运算符除外）均为自左到右。

2. 算术表达式

1) 算术表达式是用+、-、*、/、%、++、--等算术运算符连接数值型的运算量，完成数值计算功能的式子。其中，%只能用于int型或char型数据的求余数，余数的符号与被除数相同。

2) 算术表达式允许不同类型数据的混合运算，计算机自动将低精度类型转换成高精度类型后再计算，表达式值的类型取决于其中最高精度数据的类型。求值规则为：

规则1 如果表达式中含char型，则全部转换成int型。

规则2 如果表达式中含float型，则全部转换成double型。

规则3 同类型的运算量运算后，结果仍为同类型；不同类型的运算量运算前，先将低精度类型转换成高精度类型，运算结果为高精度类型。

根据求值规则，算术表达式的类型不可能是char或float型。

3) 自动类型转换总是将低精度转换为高精度，其规则为：

(1) 精度由低到高的顺序，一是看它的长度，二是看它所能表示的数据范围，依次为 char、short int、int、unsigned int、long int、unsigned long int、float、double、long double。

(2) 有符号整数转换为无符号整数时，将符号位视为数值位。负数转换为无符号数，则其值会改变。

(3) 短字节整数转换为长字节整数时，将短字节数放入长字节的低端，高端进行符号扩展（即符号位为 0 时，剩余的高端位全补 0；符号位为 1 时，剩余的高端位全补 1），不会造成数值的改变。

(4) 低精度数转换为高精度数通常不会造成数值的改变，但负整数转换为无符号整数将造成值的改变。

4) 强制类型转换

(1) 当自动类型转换达不到要求时，可以用强制类型转换将一个算术表达式或其中的某些运算量的值转换成所需要的类型。

(2) 使用强制类型转换时，类型关键字必须用括号括起来，例如(float)(a+b)。

(3) 强制类型转换只能将表达式的值转换成所需要的类型，并不能改变算术表达式本身的类型。例如，设已定义

```
int a=3;
```

```
float b=4.5;
```

则(int)(a+b)可得到整数 7，而表达式 a+b 本身仍然是 double 型的，其值为 7.5。

5) 强制类型转换既可以将低精度转换为高精度，也可以将高精度转换为低精度。高精度数转换为低精度数的规则为：

(1) 无符号整数转换为有符号整数时，将无符号数的最高位视为符号位。

(2) 长字节整数转换为短字节整数时，将高端截去。

(3) 高精度数转换为低精度数时常会造成值的改变，例如，无符号整数转换为有符号整数、绝对值大于 32767 的浮点数转换为整数、实数转换为整数都可能改变其值。

3. 赋值表达式

1) 在计算数值型的赋值表达式时，要将赋值号右边的表达式转换成与左边变量或数组元素相同的类型。这种转换既可能将高精度转换成低精度，也可能将低精度转换成高精度。例如

```
int a; char b; float c;
```

```
c=2.5;
```

```
b=c;
```

```
a=b;
```

在赋值表达式 c=2.5 中，首先要将双精度型常数 2.5 转换成 float 型，再赋给变量 c；在赋值表达式 b=c 中，先要将 c 中的 float 型数转换成 char 型，然后赋给变量 b；在赋值表达式 a=b 中，先要将 b 中的 char 型值转换成 int 型，然后赋给变量 a。前两个赋值要将高精度转换成低精度，第三个赋值则要将低精度转换为高精度。最后结果为 a=2,b=2,c=2.5。

2) 在赋值过程中，要注意不同类型变量或数组元素所能存放的数值范围，C 语言并不检查数据是否越界。例如