

高等院校信息技术规划教材

C语言实践训练

马秀丽 刘志妩 虞闯 编著

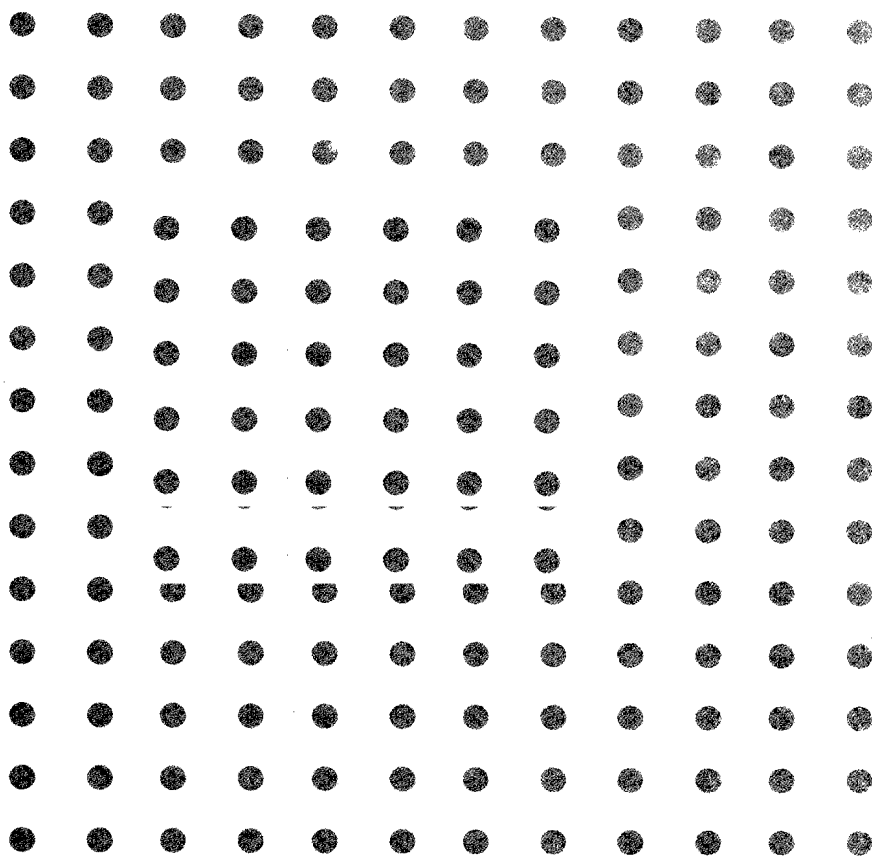


清华大学出版社

高等院校信息技术规划教材

C语言实践训练

马秀丽 刘志妩 虞闯 编著



清华大学出版社
北京

内 容 简 介

本书是结合本科生 C 语言程序设计课程教学大纲和全国计算机等级考试 C 语言上机考试的内容而编写的,旨在由浅入深、循序渐进地帮助读者提高 C 语言的实际编程能力。

本书共分 3 篇,第 1 篇是实验与上机指导,在总结 C 语言的语法规则基础上,帮助读者提高基础知识的实验和运用能力。本篇分 9 章,前 8 章均由浅入深地给出实验题目和要求,讲解编程分析过程,给出实验代码和运行结果,并辅以上机指导和常见错误分析,第 9 章介绍了复杂或大型程序的调试方法。第 2 篇是实践提高,在基础知识实验的基础之上,帮助读者进一步提高基础知识的综合运用能力和分析解决实际问题的能力。本篇采用项目训练方式,给出典型的实践题目,讲解题目分析和知识的综合运用过程,给出难点分析并辅以关键代码。第 3 篇是习题解答,包括清华大学出版社出版的《C 语言程序设计》(马秀丽等编著)一书中各章的全部习题,并给出了参考解答和程序的运行结果,其中代码均在 Turbo C 或 Visual C++ 6.0 环境下调试通过。

本书内容全面,结构合理,实例丰富,实用性强,是学习 C 语言的理想实验教材,适合高等院校计算机及其他工科类相关专业的本科生使用,也适合爱好 C 语言的自学者和程序设计人员使用。

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

版权所有,侵权必究。侵权举报电话:010-62782989 13701121933

图书在版编目(CIP)数据

C 语言实践训练/马秀丽,刘志妩,虞闯编著. —北京:清华大学出版社,2010.3

(高等院校信息技术规划教材)

ISBN 978-7-302-22046-6

I. ①C… II. ①马… ②刘… ③虞… 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 025611 号

责任编辑:袁勤勇 张为民

责任校对:梁毅

责任印制:王秀菊

出版发行:清华大学出版社

地 址:北京清华大学学研大厦 A 座

<http://www.tup.com.cn>

邮 编:100084

社 总 机:010-62770175

邮 购:010-62786544

投稿与读者服务:010-62776969, c-service@tup.tsinghua.edu.cn

质 量 反 馈:010-62772015, zhiliang@tup.tsinghua.edu.cn

印 装 者:北京国马印刷厂

经 销:全国新华书店

开 本:185×260 印 张:19.75

字 数:464 千字

版 次:2010 年 3 月第 1 版

印 次:2010 年 3 月第 1 次印刷

印 数:1~3000

定 价:29.00 元

本书如存在文字不清、漏印、缺页、倒页、脱页等印装质量问题,请与清华大学出版社出版部联系调换。联系电话:010-62770177 转 3103 产品编号:036245-01

前言

foreword

C语言是从事计算机及其外围设备的软硬件开发的技术人员必备的语言工具。在学习C语言语法规则知识的基础上,基础知识的综合运用能力和程序设计能力的提高,需要在实践中反复训练才能达到。本书结合了本科生C语言程序设计课程教学大纲和全国计算机等级考试C语言上机考试的内容而编写,在编写过程中,力求通过实验和实践两个环节,分层次、由浅入深、循序渐进地帮助读者提高C语言的实际编程能力。

本书共分3篇,第1篇是实验与上机指导,分9章,前8章内容包括C语句及其程序设计、数组、函数、指针、结构体、共用体与枚举类型、文件,每章均由浅入深地给出实验题目和要求,讲解编程分析过程,给出实验代码和运行结果,并辅以上机指导和常见错误分析。第9章介绍了复杂和大型程序的调试方法。第2篇是实践提高,包括14个实践项目,采用项目训练的方式,给出了典型的实践题目,讲解题目分析和知识的综合运用过程,给出了难点分析,并辅以关键代码。第3篇是习题解答,包括清华大学出版社出版的《C语言程序设计》(马秀丽等编著)一书中各章的全部习题,并给出了参考解答和程序的运行结果,其中的代码均在Turbo C或Visual C++ 6.0环境下调试通过。

本书的特色体现在以下4个方面:

(1) 强调基础知识的实验。

本书较全面地总结了C语言的所有语法规则,并以语法规则为基础,通过实验环节强化基础知识的运用。每个实验内容均由浅入深地给出实验题目和要求,讲解编程分析过程,给出实验代码和程序运行结果,并辅以上机指导和常见错误分析。

(2) 强化基础知识的综合运用。

在实验环节基础之上,需要进一步提高基础知识的综合运用能力和分析解决实际问题的能力,这需要在实践中加以提高。实践部分采用项目训练方式,给出了典型的实践题目、讲解题目分析和知识的综合运用过程,给出难点分析,并辅以关键代码,循序渐进地提



高知识的综合运用能力和解决实际问题的能力。

(3) 强调反复练习。

知识的掌握需要大量反复的练习加以巩固才能获得,本书给出清华大学出版社出版的《C 语言程序设计》(马秀丽等编著)一书中的全部习题及参考解答,并给出了程序运行结果。其中代码均在 Turbo C 或 Visual C++ 6.0 环境下调试通过。

(4) 本书适合作为 C 语言实验教材和课程设计教材使用,也适合自学者使用。

本书第 1 篇的第 2、6、7、8、9 章和第 3 篇的第 1、4、6、10、11 章由马秀丽编写,第 1 篇的第 1、3、4、5 章和第 3 篇的第 2、3、5、7、8 章由刘志妩编写,第 2 篇由虞闯编写,第 3 篇的第 9、12 章由李筠编写。此外,王红霞、吴迪、王展红、杨秀杰、徐志勇、付立冬、苑擎颢参加了部分编写工作。另外,特别感谢张秉权教授审阅了全稿,并提出了许多宝贵建议。同时感谢本书所列参考文献的作者。

欢迎读者对本书内容提出任何建议,作者电子邮箱:xiulima@sina.com。

作 者

2010 年 1 月

目录

Contents

第1篇 实验与上机指导

第1章 C 语句及其程序设计	3
1.1 本章知识要点	3
1.1.1 基本数据类型	3
1.1.2 标识符、常量和变量	4
1.1.3 数据的输入与输出	6
1.1.4 C 运算符与表达式	8
1.1.5 顺序结构程序设计	13
1.1.6 选择结构程序设计	13
1.1.7 循环结构程序设计	16
1.2 上机实验与指导	17
1.2.1 C 语言基础知识实验	17
1.2.2 顺序结构程序设计实验	21
1.2.3 选择结构程序设计实验	22
1.2.4 循环结构程序设计实验	24
1.3 常见错误	26
第2章 数组	28
2.1 本章知识要点	28
2.1.1 数组的概念	28
2.1.2 一维数组	28
2.1.3 二维数组	29
2.1.4 字符数组	30
2.1.5 字符串处理函数	31
2.1.6 数组的应用	32
2.2 上机实验与指导	34
2.2.1 一维数组和二维数组的编程实验	34



2.2.2	字符数组和字符串函数的编程实验	37
2.2.3	上机练习	41
2.3	常见错误	42
第3章	函数	43
3.1	本章知识要点	43
3.1.1	函数的概念	43
3.1.2	函数的定义	43
3.1.3	函数的调用	44
3.1.4	声明函数原型	45
3.1.5	函数的参数传递	45
3.1.6	函数的嵌套调用与递归调用	45
3.1.7	变量的作用域	46
3.1.8	变量的存储属性	46
3.2	上机实验与指导	48
3.2.1	函数的定义及调用实验	48
3.2.2	变量的作用域与存储类别实验	52
3.3	常见错误	54
第4章	编译预处理命令	55
4.1	本章知识要点	55
4.1.1	预处理命令的概念	55
4.1.2	宏定义	55
4.1.3	文件包含	57
4.1.4	条件编译	57
4.2	上机实验与指导	59
4.3	常见错误	60
第5章	指针	62
5.1	本章知识要点	62
5.1.1	指针与指针变量	62
5.1.2	指向变量的指针变量	62
5.1.3	指针与函数	63
5.1.4	指针与数组	65
5.1.5	指针与字符串	67
5.1.6	指针数组	68
5.1.7	指向指针的指针	69

5.1.8	main 函数的参数	69
5.2	上机实验与指导	70
5.2.1	指针的应用实验	70
5.2.2	二级指针的应用实验	73
5.3	常见错误	76
第 6 章	结构体与链表操作	77
6.1	本章知识要点	77
6.1.1	结构体的概念和定义	77
6.1.2	结构体变量的定义和初始化	78
6.1.3	结构体变量的引用	78
6.1.4	结构体数组	79
6.1.5	结构体指针	80
6.1.6	链表的操作	81
6.2	上机实验与指导	83
6.2.1	结构体数据类型的编程实验	83
6.2.2	结构体链表的编程实验	86
6.2.3	上机练习	90
第 7 章	共用体与枚举类型	92
7.1	本章知识要点	92
7.1.1	共用体	92
7.1.2	枚举类型	94
7.1.3	用 typedef 定义类型	95
7.2	上机实验与指导	97
7.2.1	共用体数据类型的编程实验	97
7.2.2	枚举类型的编程实验	99
7.2.3	上机练习	102
第 8 章	文件	103
8.1	本章知识要点	103
8.1.1	文件的概念	103
8.1.2	文件的基本操作	104
8.1.3	文件的随机操作	106
8.1.4	文件的状态检测	107
8.2	上机实验与指导	108
8.2.1	文件的基本操作的编程实验	108



8.2.2 文件的随机操作的编程实验	111
8.2.3 上机练习	114
第9章 如何调试C程序	115

第2篇 实践提高

实践1 编写一个猜数字游戏	121
实践2 学生成绩的有序管理	126
实践3 保安值班安排系统	133
实践4 流星雨的实现	138
实践5 编写一个万年历	143
实践6 约瑟夫生死者游戏	149
实践7 字符串回文个数统计	154
实践8 求 n 个叠数之和	158
实践9 找出两个字符串中最长的相同单词	162
实践10 销售管理与统计	166
实践11 八皇后问题	175
实践12 文本文件的检索	181
实践13 科研成果管理系统	187
实践14 求两个一元多项式的和	197

第3篇 习题解答

第1章 C语言概述习题解答	205
第2章 算法习题解答	206
第3章 基本数据类型及数据的输入输出习题解答	209
第4章 运算符与表达式习题解答	213
第5章 C语句及其程序设计习题解答	217
第6章 数组习题解答	230

第 7 章 函数习题解答	243
第 8 章 编译预处理命令习题解答	255
第 9 章 指针习题解答	261
第 10 章 结构体类型与链表操作习题解答	273
第 11 章 共用体与枚举类型习题解答	281
第 12 章 文件习题解答	286
附录 A C 语言报错信息	295
参考文献	300

第 1 篇

实验与上机指导

C 语句及其程序设计

1.1 本章知识要点

1.1.1 基本数据类型

数据类型是按被定义变量的性质、表示形式和占据存储空间的多少来划分的。在 C 语言中,数据类型可划分为基本数据类型、构造数据类型、指针类型和空(void)类型四大类,如图 1.1 所示。

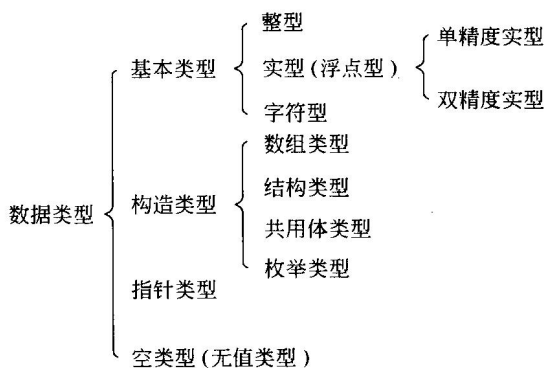


图 1.1 C 语言的数据类型

不同类型的数据在内存中所占的存储空间的长度不同,数据的取值范围也不同,如表 1.1 所示。

表 1.1 基本数据类型的特点

类型标识符	类型说明	长度	取值范围
int	有符号整型	2 字节	-32768~32767
long	有符号长整型	4 字节	-2147483648~2147483647
unsigned int	无符号整型	2 字节	0~65535
unsigned long	无符号长整型	4 字节	0~4294967295



续表

类型标识符	类型说明	长度	取值范围
float	单精度实型	4 字节	$-3.4 \times 10^{-38} \sim 3.4 \times 10^{38}$
double	双精度实型	8 字节	$-1.7 \times 10^{-308} \sim 1.7 \times 10^{308}$
char	有符号字符型	1 字节	-128~127
unsigned char	无符号字符型	1 字节	0~255

注意：在字符型数据的取值范围内，字符型数据和整型数据可以通用。

1.1.2 标识符、常量和变量

1. 标识符

标识符就是一个名字，用来标识常量、变量、数组以及函数等的名称。

在 Turbo C 中有以下三类标识符：

(1) 关键字：指已被 C 系统保留的有特定含义的标识符，Turbo C 2.0 的关键字有 32 个。

(2) 编译预处理的命令单词，如 #include、#define 等。

(3) 用户标识符：指用户自己定义的变量名、常量名、函数名和数组名等。

用户自定义标识符的命名规则如下：

- 标识符由字母、数字、下划线组成，且不能以数字开头。例如，a2、_b1 是正确的，而 2a 就不是合法的变量名。
- 标识符区分大小写字母。例如，Ab 与 ab 代表不同的标识符。
- 用户标识符不要使用 C 系统的关键字和编译预处理命令单词。

2. 常量

常量是指程序运行过程中其值不变的量。

(1) 整型常量：指不带小数点的整数。C 语言可以处理不同进制的常整数，有十进制、八进制和十六进制。

十进制整数：由 0~9 这 10 个数码及正、负号组成，如 -200、50 等。注意不能以 0 开头。

八进制整数：由 0~7 这 8 个数码组成，且必须以 0 开头。如 012、071 等。

十六进制整数：由 0~9、a~f(或 A~F) 这 16 个数码组成，且必须以 0x 开头。如 0x20、0x1a、0xff 等。

长整型常量：在整型常量后边加 L(或 l) 后缀来表示。如 10L、012L、0x2aL 等。

(2) 实型常量：指数学中表示的实数，可以表示带小数点的数。实型常量有小数形式和指数形式两种。

十进制小数形式：例如，3.14、0.035 等。

指数形式：例如， $3.45\text{e}-2$ ，表示 3.45×10^{-2} 。

(3) 字符常量：是用单引号括起来的一个字符。如：'A'、'b'等都是常见的字符常量。字符常量是与其对应的 ASCII 码值存放在存储器中的。例如，'A'的 ASCII 码值为 65，'0'的 ASCII 码值为 48。

除了以上形式的字符常量外，C 语言还允许特殊形式的字符常量，即以“\”开头的字符常量，又被称为转义字符或控制字符。控制字符可以控制输出光标的位置。如 '\n' 表示控制输出换行，'\b' 表示控制输出回退一格，'\r' 表示回车，即控制输出到本行的开头。

(4) 字符串常量：是指用双引号括起来的一个或多个字符。例如，“abcd”，“a”等都表示字符串常量。字符串的存放形式是按每个字符在串中的顺序，依次存放各自对应的 ASCII 码值，并且以 '\0' 结尾。'\0' 的 ASCII 码值是 0，在 C 语言中作为字符串的结束标志。

注意：“c”和 'c' 是不同的。前者是字符串，后者是字符，二者的存放形式是不同的。字符串 “c” 的存放形式为 2 个字节，前一个字节存放其 ASCII 码值 99，后 1 个字节存放字符串结束标志 '\0'。而字符常量 'c' 的存放只占 1 个字节，只存放其 ASCII 码值 99。所以不能将字符串常量赋给字符型变量。即 `char ch='c';` 是正确的，而 `char ch="c";` 则是错误的。

(5) 符号常量：C 语言程序中允许用一个标识符来表示一个常量，称之为符号常量。符号常量的定义格式如下：

#define 常量名 常量表达式

例如：

```
#define PI 3.14159
```

表示在程序中出现的 PI 符号常量都将替换为数值 3.14159 参与运算。

3. 变量

变量是在程序运行期间其值可以改变的量。每个变量都有名字、类型和值，并且在内存中占据一定的存储空间，用来存放数据。不同类型的变量占用的存储单元的长度不同，其中存储单元的首地址称为变量的地址。

1) 变量的定义

C 语言规定变量要先定义，后使用。所谓定义变量就是声明变量的名称和变量所属的数据类型。变量一经定义后，C 编译系统就会为该变量分配相应类型的存储单元。

变量的定义格式如下：

数据类型 变量名列表；

例如：

```
int a;                /* 定义 a 为整型变量，为其分配 2 字节的存储单元 */
float c, d;           /* 定义 c 和 d 为单精度实型变量，各分配 4 字节的存储单元 */
```

2) 变量的赋值和初始化

定义变量，只是为变量名分配了存储单元，并没有赋值。即这些存储单元中仍存放



着其原始数据,这些数据是随机的且对于程序是无意义的,因此在使用变量前,要先为其赋予初值。

可以用赋值语句为变量赋值。如:

```
int a; float b;           /* 定义变量 */
a=10;                     /* 为变量赋初值 */
b=3.45;                   /* 为变量赋初值 */
```

也可以在定义变量的同时赋予初值,称之为变量的初始化。如:

```
int a=10, float b=3.45;
```

注意: 不允许在定义变量时,连续为多个变量赋值。如:

```
int a=b=c=0;              /* 这种初始化方法是错误的 */
```

1.1.3 数据的输入与输出

在 C 语言中,输入或输出操作是通过调用输入输出库函数来实现的。C 语言函数库中有一批标准输入输出函数,包括 printf()、scanf()、putchar() 和 getchar()等。

在使用标准输入输出库函数时,应包含的头文件是“stdio.h”,即在程序的开头应该加入以下命令:

```
#include <stdio.h>
```

或者

```
#include "stdio.h"
```

1. 字符输出函数 putchar()

函数的调用格式:

```
putchar(x);
```

函数的功能:显示单个字符。将变量 x 中的内容以一个字符形式输出到屏幕上。其中 x 可以是变量,也可以是常量。

2. 字符输入函数 getchar()

函数的调用格式:

```
c=getchar();
```

函数的功能:接收从标准输入设备(键盘)输入的一个字符,并将该字符作为函数的值赋给变量 c。

该函数的执行是等待从键盘输入一个字符,当用户输入一个字符后,函数的值就是所输入的字符常量。所以常用赋值语句的形式,将键盘输入的字符,赋给一个变量。

例如:

```
c=getchar();           /* 输入一个字符 */
```

使用 `getchar` 函数应注意以下几点:

- (1) `getchar` 函数只能接收单个字符,输入的数字也按字符处理。
- (2) 使用 `getchar` 函数时,Enter 键也会作为输入字符的一部分。尤其在连续使用 `getchar` 函数时要注意 Enter 键将会作为换行符被下一个 `getchar()` 所接收。

3. 格式输出函数 `printf()`

函数的调用格式:

```
printf("格式控制",输出项列表);
```

函数的功能: 按指定格式,向输出设备输出若干个指定类型的数据。

`printf` 函数可以用于各种类型数据的输出。该函数的执行过程是先按输出项列表中自右向左的顺序,依次计算各输出项的值,再按“格式控制”指定的格式,将输出项的值输出到显示器上。

`printf` 函数的参数包括格式控制和输出项列表两部分。格式控制部分要用一对双撇号括起来,用于说明输出项所采用的格式;输出项列表是所要输出的内容,可以是变量、表达式或常量。

使用说明:

- (1) 格式控制部分,用于指定输出格式,由普通字符和格式说明两部分组成。
- (2) 普通字符(包括转义控制字符),将原样输出。
- (3) 格式说明部分,以 % 符号开头,以格式说明符结尾(格式说明符的含义,详见教科书),用于控制输出数据的类型和形式。
- (4) 一般格式说明符与输出项在个数、次序和数据类型方面要一一对应。若格式说明符的个数少于输出项数,多余的输出项将不会输出;反之若格式说明符的个数多于输出项数,则多余的格式说明将输出不定值。

4. 格式输入函数 `scanf()`

函数的调用格式:

```
scanf("格式控制说明",地址列表);
```

函数的功能: 接收键盘输入的信息,按指定格式转换后,存放到地址列表所对应的变量中。

使用说明:

- (1) 地址列表由若干个地址组成,它们可以是变量的地址也可以是字符串的首地址。变量的地址可以通过取地址运算符 & 得到。例如,变量 `a` 的地址表示为 `&a`。
- (2) 格式控制说明字符串中,可以包含普通字符和格式控制说明符。格式控制符的作用与 `printf()` 的相似,用于控制输入数据的类型;普通字符是用户自加的,可以用来分隔数据。这种情况下,用户输入数据时,必须输入自加的普通字符。