

● 高等学校21世纪计算机教材

C语言 程序设计及应用 习题解析与上机指导

谢乐军 编著



冶金工业出版社

高等学校 21 世纪计算机教材

C 语言程序设计及应用习题解析 与上机指导

谢乐军 编著

北 京

冶金工业出版社

2004

内 容 简 介

程序设计是计算机及相关专业学生必备的基本技能之一，因此程序设计课程是计算机相关专业一门重要的必修课。本书共分三部分，第一部分回顾了程序设计的基础知识、C 语言的基本概念、顺序结构程序设计、分支结构程序设计、循环结构程序设计、函数、指针、数组、结构体和共用体以及枚举类型、编译预处理、位运算、文件以及 C 语言的图形功能等内容，同时还给出了较为丰富的例题和习题，以供学习者进行模仿和练习。第二部分是上机实验指导，包括了 11 个实验，涵盖了 C 语言的大部分内容，同时书末还附有习题与实验部分的参考答案。第三部分是 6 套模拟试题，并配有参考答案，以供读者参考使用。

本书结构清晰、习题丰富、语言通俗易懂，既可作为高等院校计算机相关专业的教材，又可作为计算机培训班学习 C 语言程序设计课程的复习资料和上机实验指导资料。

图书在版编目 (CIP) 数据

C 语言程序设计及应用习题解析与上机指导 / 谢乐军编著.

北京: 冶金工业出版社, 2004.5

ISBN 7-5024-3514-X

I. C... II. 谢... III. C 语言—程序设计—自学参考资料
IV. TP312

中国版本图书馆 CIP 数据核字 (2004) 第 026486 号

出版人 曹胜利 (北京沙滩嵩祝院北巷 39 号, 邮编 100009)

责任编辑 程志宏

佛山市新粤中印刷有限公司印刷; 冶金工业出版社发行; 各地新华书店经销

2004 年 5 月第 1 版, 2004 年 5 月第 1 次印刷

787mm×1092mm 1/16; 20 印张; 590 千字; 312 页; 1-2500 册

30.00 元

冶金工业出版社发行部 电话: (010) 64044283 传真: (010) 64027893

冶金书店 地址: 北京东四西大街 46 号 (100711) 电话: (010) 65289081

(本社图书如有印装质量问题, 本社发行部负责退换)

前 言

一、本书背景

软件开发离不开编程语言，而在众多的高级编程语言中，C 语言无疑占有很重要的地位。C 语言是一种在 UNIX 操作系统的早期就被广泛使用的通用编程语言。它最早是由贝尔实验室的 Dennis Ritchie 为了 UNIX 的辅助开发而写的，开始时 UNIX 是用汇编语言和一种叫 B 的语言编写的。从那时候起，C 语言就成为世界上使用最广泛的计算机语言。C 语言能在编程领域里得到如此广泛支持的原因有以下一些：

它是一种非常通用的语言，几乎所能想到的任何一种计算机上都有至少一种能用的 C 编译器，并且它的语法和函数库在不同的平台上都是统一的，这个特性对开发者来说很有吸引力。用 C 语言写的程序执行速度很快。C 语言是所有版本的 UNIX 上的系统语言。

二、本书结构

本书共分三部分，其具体结构安排如下：

第一部分：第 1~13 章。主要对各知识点进行回顾，同时还包括了典型例题分析和练习题，以供学习者进行模仿和练习。

第 1 章：程序设计基础。介绍了程序设计、结构化程序设计等概念和方法，还有使用流程图、N-S 图描述程序流程，最后介绍 C 语言的一些基本知识，如 C 语言的特点、结构等。

第 2 章：C 语言的基本概念。介绍了 C 语言的基本概念，包括常量、变量、简单数据类型、基本运算符与表达式和数据类型转换。这是以后深入学习 C 语言的重要基础。

第 3 章：顺序结构程序设计。介绍了程序的顺序结构和实现这种结构的语句，讲述了赋值语句、标准输入/输出函数的使用方法。

第 4 章：分支结构程序设计。讲述分支结构程序设计的思想、判断条件的关系表达式和逻辑表达式，还有实现分支结构的 if 语句和 switch 语句。

第 5 章：循环结构程序设计。介绍了循环结构程序设计以及实现循环的各种 C 语言语句。

第 6 章：函数。介绍了函数定义、函数调用、变量的作用域和类别、内外函数等知识。

第 7 章：指针。先介绍了 C 语言指针的概念，接着介绍 C 语言中指针变量的定义方式和对指针变量的引用方法。在建立了指针的概念后，本章还介绍了指针作为函数参数、函数间的地址传送、函数的指针等指针的一般用法。

第 8 章：数组。主要讲述了一维数组、二维数组、字符数组以及它们和指针的关系，还介绍了如何在函数中传递数组，并给出了许多例子以帮助理解。

第 9 章：结构体、共用体和枚举类型。首先介绍了结构体的类型、定义以及一般的用法。接着把结构体和数组、函数、指针联系起来，讲解了结构体的高级应用。最后还介绍了链表、共用体、枚举类型和用户自定义类型。

第 10 章：编译预处理。介绍了常用的几种预处理功能，包括宏定义、文件包含、条件编译等。

第 11 章：位运算。介绍位运算，以及 6 种位运算符的使用以及位域。

第 12 章：文件。介绍了 C 语言中文件的概念，包括 ASCII 文件、二进制文件和流文件等。接着介绍文件的基本操作和与文件操作相关的函数的使用方法，包括文件打开、关闭、读、写、定位和出错检测等。

第 13 章：C 语言的图形功能简介。主要介绍了 C 语言中图形模式的概念，然后列举了基本的图

形函数，如画点、画线函数等。

第二部分：第14~24章。主要介绍了上机实验指导，提供了多道实验题供读者自己实践。每个实验题目都有详细的源代码。

第14章：实验一 —— 顺序结构与输入/输出方法。

第15章：实验二 —— 各种分支结构。

第16章：实验三 —— 函数与变量。

第17章：实验四 —— 指针及其运算。

第18章：实验五 —— 数组和字符串。

第19章：实验六 —— 指针与字符串。

第20章：实验七 —— 指针与数组。

第21章：实验八 —— 结构体与共用体。

第22章：实验九 —— 定义编译预处理。

第23章：实验十 —— 枚举与位运算。

第24章：实验十一 —— 文件系统。

第三部分：全真模拟试卷，共有6套。全部模拟题来源于计算机等级考试和国内重点院校计算机专业考试题目，其中部分试题稍作修改，且在试卷末附有参考答案，以供读者参考。

三、本书特点

本书结构清晰、语言力求简练、文字力求准确、方便阅读，例题习题选用既实用又丰富，并配有详细的分析过程，强调编程技能的培养。

四、本书适用对象

本书不仅可作为高等院校计算机相关专业的教材，还可作为计算机培训班学习C语言程序设计课程的复习资料和上机实验指导资料。

本书意在提供一本内容实用并具特色的C语言程序设计教材，但由于水平有限，难免有错误和不妥之处，希望广大读者批评指正。

虽然经过严格的审核、精细的编辑，本书在质量上有了一定的保障，但我们的目标是力求尽善尽美，欢迎广大读者和专家对我们的工作提出宝贵建议，联系方法如下：

电子邮件：service@cnbook.net

网址：www.cnbook.net

此外，本书所附源代码也可从该网站免费下载，该网站还有一些其他相关书籍的介绍，以便读者选购和参考。

编 者

2004年4月

目 录

第一部分 习题与解析

第1章 程序设计基础.....2

1.1 知识点回顾.....2

1.1.1 程序.....2

1.1.2 算法.....2

1.1.3 数据结构.....2

1.1.4 程序设计语言.....2

1.1.5 一般程序设计方法.....3

1.1.6 结构化程序设计.....3

1.1.7 程序设计风格.....4

1.1.8 C语言的特点.....4

1.1.9 C程序的结构.....4

1.1.10 C语句概述.....4

1.2 典型例题分析.....5

结束语.....6

综合练习一.....6

一、选择题.....6

二、填空题.....7

三、程序设计题.....7

第2章 C语言的基本概念.....8

2.1 知识点回顾.....8

2.1.1 常量.....8

2.1.2 变量.....8

2.1.3 简单数据类型.....8

2.1.4 运算符与表达式.....9

2.1.5 自动类型转换.....9

2.1.6 强制类型转换.....10

2.2 典型例题分析.....10

结束语.....11

综合练习二.....12

一、选择题.....12

二、填空题.....16

三、程序设计题.....17

第3章 顺序结构程序设计.....18

3.1 知识点回顾.....18

3.1.1 顺序结构的设计思想.....18

3.1.2 赋值语句.....18

3.1.3 标准的输入/输出.....18

3.2 典型例题分析.....20

结束语.....21

综合练习三.....21

一、选择题.....21

二、填空题.....25

三、程序设计题.....28

第4章 分支结构程序设计.....29

4.1 知识点回顾.....29

4.1.1 关系运算符与关系表达式.....29

4.1.2 逻辑运算符与逻辑表达式.....29

4.1.3 if语句.....30

4.1.4 switch语句.....31

4.2 典型例题分析.....31

结束语.....35

综合练习四.....35

一、选择题.....35

二、填空题.....39

三、程序设计题.....41

第5章 循环结构程序设计.....42

5.1 知识点回顾.....42

5.1.1 循环结构的设计思想.....42

5.1.2 while语句.....42

5.1.3 for语句.....42

5.1.4 do-while语句.....43

5.1.5 多重循环.....43

5.1.6 break语句.....43

5.1.7 continue语句.....44

5.1.8 goto语句.....44

5.2 典型例题分析.....44

结束语.....49

综合练习五.....49

一、选择题.....	49	8.1 知识点回顾.....	126
二、填空题.....	62	8.1.1 一维数组的定义.....	126
三、程序设计题.....	74	8.1.2 一维数组元素的引用.....	126
第 6 章 函数.....	75	8.1.3 一维数组的初始化.....	126
6.1 知识点回顾.....	75	8.1.4 二维数组的定义.....	126
6.1.1 函数定义.....	75	8.1.5 二维数组元素的引用.....	127
6.1.2 函数的参数.....	75	8.1.6 二维数组的初始化.....	127
6.1.3 函数的返回值.....	76	8.1.7 字符数组的定义和使用.....	127
6.1.4 函数声明.....	76	8.1.8 字符串和字符数组.....	127
6.1.5 函数调用.....	76	8.1.9 常用字符串处理函数.....	128
6.1.6 函数调用中的值传递方式.....	76	8.1.10 指向一维数组的指针变量.....	128
6.1.7 函数嵌套调用.....	76	8.1.11 通过指针引用数组元素.....	128
6.1.8 函数递归调用.....	77	8.1.12 二维数组元素的地址.....	128
6.1.9 变量的作用域.....	77	8.1.13 指向二维数组元素的指针变量.....	129
6.1.10 变量的存储类别.....	77	8.1.14 指向字符串常量的指针变量.....	129
6.1.11 外部函数.....	78	8.1.15 字符串常量指针变量与字符 数组的区别.....	129
6.1.12 内部函数.....	78	8.1.16 值传递方式与地址传递方式.....	130
6.2 典型例题分析.....	78	8.1.17 数组名做函数参数.....	130
结束语.....	85	8.1.18 指针数组的定义和作用.....	130
综合练习六.....	85	8.1.19 main 函数的参数.....	130
一、选择题.....	85	8.2 典型例题分析.....	130
二、填空题.....	91	结束语.....	136
三、程序设计题.....	110	综合练习八.....	136
第 7 章 指针.....	113	一、选择题.....	136
7.1 知识点回顾.....	113	二、填空题.....	145
7.1.1 指针变量的定义.....	113	三、程序设计题.....	167
7.1.2 指针变量的引用.....	113	第 9 章 结构体、共用体和枚举类型.....	169
7.1.3 指针作函数参数.....	113	9.1 知识点回顾.....	169
7.1.4 返回指针值的函数.....	114	9.1.1 结构体类型的定义.....	169
7.1.5 函数的指针与指向函数的 指针变量.....	114	9.1.2 结构体变量的定义.....	169
7.2 典型例题分析.....	114	9.1.3 结构体变量的初始化.....	169
结束语.....	120	9.1.4 结构体数组的定义.....	169
综合练习七.....	120	9.1.5 结构体数组的初始化.....	170
一、选择题.....	120	9.1.6 向函数传递结构体数据.....	170
二、填空题.....	123	9.1.7 指向结构变量的指针.....	170
三、程序设计题.....	125	9.1.8 指向结构数组的指针.....	170
第 8 章 数组.....	126	9.1.9 结构指针变量作函数参数.....	170
		9.1.10 动态存储结构.....	170

9.1.11 链表概念	170	结束语	200
9.1.12 链表操作	171	综合练习十一	200
9.1.13 共用体类型的定义	171	一、选择题	200
9.1.14 共用体变量使用	171	二、填空题	201
9.1.15 枚举类型和枚举变量的定义	171	三、程序设计题	202
9.1.16 枚举类型变量使用	171	第 12 章 文件	203
9.1.17 用户自定义类型	171	12.1 知识点回顾	203
9.2 典型例题分析	172	12.1.1 文件的概念	203
结束语	178	12.1.2 C 文件操作的基本方法	203
综合练习九	178	12.1.3 文件指针	204
一、选择题	178	12.1.4 文件打开	204
二、填空题	180	12.1.5 文件关闭	204
三、程序设计题	181	12.1.6 读、写文件	204
第 10 章 编译预处理	182	12.1.7 文件定位	204
10.1 知识点回顾	182	12.1.8 出错检测	204
10.1.1 不带参数的宏定义	182	12.2 典型例题分析	205
10.1.2 带参数的宏定义	182	结束语	210
10.1.3 终止宏定义	183	综合练习十二	210
10.1.4 文件包含	183	一、选择题	210
10.1.5 条件编译	183	二、填空题	211
10.2 典型例题分析	184	三、程序设计题	212
结束语	186	第 13 章 C 语言的图形功能简介	213
综合练习十	186	13.1 知识点回顾	213
一、选择题	186	13.1.1 图形模式	213
二、填空题	191	13.1.2 基本图形函数	213
三、程序设计题	196	13.2 典型例题分析	215
第 11 章 位运算	197	结束语	216
11.1 知识点回顾	197	综合练习十三	216
11.1.1 字节和位	197	一、选择题	216
11.1.2 原码、反码、补码	197	二、填空题	217
11.1.3 位运算符	197	三、程序设计题	217
11.1.4 按位与运算符	197	第二部分 上机实验指导	
11.1.5 按位或运算符	197	第 14 章 实验一——顺序结构与	
11.1.6 按位异或运算符	198	输入/输出方法	219
11.1.7 取反运算符	198	14.1 实验目的	219
11.1.8 左移运算符<<	198	14.2 实验要点	219
11.1.9 右移运算符>>	198	14.3 实验要求	219
11.1.10 位域	198		
11.2 典型例题分析	198		

14.4 实验内容	219	21.2 实验要点	240
第 15 章 实验二——各种分支结构	221	21.3 实验要求	240
15.1 实验目的	221	21.4 实验内容	240
15.2 实验要点	221	第 22 章 实验九——定义编译预处理	243
15.2.1 if-else 语句	221	22.1 实验目的	243
15.2.2 switch-case 语句	221	22.2 实验要点	243
15.3 实验要求	221	22.3 实验要求	243
15.4 实验内容	221	22.4 实验内容	243
第 16 章 实验三——函数与变量	225	第 23 章 实验十——枚举与位运算	245
16.1 实验目的	225	23.1 实验目的	245
16.2 实验要点	225	23.2 实验要点	245
16.3 实验要求	225	23.3 实验要求	245
16.4 实验内容	225	23.4 实验内容	245
第 17 章 实验四——指针及其运算	228	第 24 章 实验十一——文件系统	247
17.1 实验目的	228	24.1 实验目的	247
17.2 实验要点	228	24.2 实验要点	247
17.3 实验要求	228	24.3 实验要求	247
17.4 实验内容	228	24.4 实验内容	247
第 18 章 实验五——数组和字符串	231	第三部分 全真模拟试卷	
18.1 实验目的	231	全真模拟试卷	251
18.2 实验要点	231	全真模拟试卷之一	251
18.3 实验要求	231	全真模拟试卷之二	254
18.4 实验内容	231	全真模拟试卷之三	257
第 19 章 实验六——指针与字符串	235	全真模拟试卷之四	260
19.1 实验目的	235	全真模拟试卷之五	264
19.2 实验要点	235	全真模拟试卷之六	267
19.3 实验要求	235	模拟试卷参考答案	272
19.4 实验内容	235	全真模拟试卷之一	272
第 20 章 实验七——指针与数组	238	全真模拟试卷之二	274
20.1 实验目的	238	全真模拟试卷之三	275
20.2 实验要点	238	全真模拟试卷之四	277
20.3 实验要求	238	全真模拟试卷之五	278
20.4 实验内容	238	全真模拟试卷之六	280
第 21 章 实验八——结构体与共用体	240	参考答案	282
21.1 实验目的	240		

第一部分 习题与解析

本部分回顾了程序设计的基础知识、C 语言的基本概念、顺序结构程序设计、分支结构程序设计、循环结构程序设计、函数、指针、数组、结构体和共用体以及枚举类型、编译预处理、位运算、文件和图形处理等内容。本书给出了较为丰富的例题和习题，以供学习者进行模仿和练习。

- 第 1 章 程序设计基础
- 第 2 章 C 语言的基本概念
- 第 3 章 顺序结构程序设计
- 第 4 章 分支结构程序设计
- 第 5 章 循环结构程序设计
- 第 6 章 函数
- 第 7 章 指针
- 第 8 章 数组
- 第 9 章 结构体、共用体和枚举类型
- 第 10 章 编译预处理
- 第 11 章 位运算
- 第 12 章 文件
- 第 13 章 C 语言的图形功能简介

第1章 程序设计基础

掌握了一门计算机语言并不等于就懂得了程序设计,要编写好程序,还必须掌握程序设计的方法。本章围绕这个问题回顾了程序设计的基本概念,程序设计的一般方法和C语言的概况等知识点,另外还给出了例题分析和综合练习题。

1.1 知识点回顾

1.1.1 程序

人和计算机打交道,必须要解决一个语言沟通问题,因为计算机不能理解和执行人们使用的自然语言,而只能接受和执行二进制的指令。这些指令的集合就叫做“程序”(program)。

换言之,一个程序是完成某一特定任务的一组指令序列,或者说,是为实现某一算法的指令序列。

1.1.2 算法

为解决一个问题而采取的方法和步骤,称为“算法”(algorithm)。或者说,算法是解题方法的精确描述。解决一个问题的过程就是实现一个算法的过程。对同一个问题,往往有不同的解题方法,每种方法都有利有弊。为了有效地进行计算,就应当选择合适的算法。选择的首要标准是算法的正确性和可靠性,简单性和易理解性,其次是算法所需的存储空间少和执行速度快等。

一个算法应具有以下特点:

- (1) 有穷性,一个算法应该包含有限个步骤,而不能是无限的。
- (2) 确定性,算法的每一步都应当是明确无误的,不能含义模糊。
- (3) 有零个或多个输入,所谓输入是指执行指定的算法时,需要外界提供的信息。
- (4) 有一个或多个输出,没有输出的算法是没有意义的。
- (5) 有效性,算法中的每一步都应该能有效地执行。

1.1.3 数据结构

计算机算法的处理对象是描述客观事物属性的数据。由于客观事物的多样性,数据有不同的形式,如整数、实数,字符、图像、声音等,它们是计算机程序加工的“原料”。数据结构是指数据对象及其之间的相互关系和构造方法。一个数据结构S可以形式地用一个二元组表示: $S=(D, R)$

其中,D是数据结构中的数据(又称为“结点”)的非空有限集合,R是定义在D上的关系的非空有限集合。

1.1.4 程序设计语言

程序语言可以分为低级语言和高级语言两大类。

低级语言又叫面向机器的语言,它是特定的计算机系统所固有的语言。它又可分为机器语言和符号语言(汇编语言)两类。

机器语言就是计算机能够直接识别和执行的指令的集合。由于计算机只能识别“0”和“1”两种状态(即电气的正负状态),所以机器语言指令都是二进制指令。例如某种型号的计算机以10000000表示“进行一次加法”。很明显,这种由0和1组成的指令难学、难记、难阅读、难修改,给用户带来

很大的不便,而且机器语言随不同机器而不同,因此移植性很差。但其优点是执行速度最快。

为了解决程序员的困难,高级语言发展起来了。50年代出现了FORTRAN语言,60年代又相继出现了COBOL,ALGOL 60,BASIC等语言,后来又陆续出现了PASCAL、C、C++等。这些语言的特点是:用一种接近自然语言和数学语言的专用语言来表示算法,而且与具体的计算机无关,即用它所写的程序可以在任一种计算机上运行。这些高级语言的出现大大提高了程序设计的效率,使人们能更方便地使用计算机。

1.1.5 一般程序设计方法

对于一般的问题,设计一个程序大概要经过以下几个步骤:

- (1) 建立数学模型;
- (2) 选定算法,并用适当工具描述;
- (3) 编程;
- (4) 测试及调试。

1. 建立数学模型

建立数学模型是程序设计中最复杂、最困难的一步,好的数学模型本身就是一个定律,它要经过大量观察、分析、推理、验证等工作后才可以得到。这是数学范畴的工作,计算机工作人员一般不用去完成这部分的研究工作,但也应对其基本知识有一定了解。在进行程序设计时往往都是利用已有的基本数学模型去构造出问题的模型,如微积分、运筹学、图论、高等数学等都是不同的基本数学模型。

2. 选定算法

算法是解决问题的方法与步骤。设计算法也是一件非常困难的工作,经常采用的算法设计技术主要有迭代法、穷举搜索法、递推法、贪婪法、回溯法、分治法、动态规划法等。另外,为了以更简洁的形式设计和描述算法,在设计算法时又常常采用递归技术,用递归描述算法。

3. 编程

编程就是将选定的算法从非计算机语言的描述形式转换为计算机语言的语句形式描述出来。这个过程同特定的高级语言有关,同一个算法可以用不同的高级语言来实现。因此,在编程前首先要选择开发的语言。

4. 测试及调试

编程完成以后,首先应该静态审查程序,即由人工“代替”或“模拟”计算机,对程序进行仔细检查,然后将高级语言源程序输入计算机,经过编译、连接,然后再运行。在编译、连接及运行时如果在某一步发现错误,就必须找到错误并改正,然后再从重新编译运行,直到得到正确结果为止。

1.1.6 结构化程序设计

要设计出结构化程序,应当采用以下的方法:

- (1) 自顶向下。
- (2) 逐步细化。
- (3) 模块化。

所谓“自顶向下,逐步细化”,是指一种先整体后局部的设计方法。对于一个较复杂的问题,一般不能立即写出详细的算法或程序,但可以很容易写出一级算法,即求解问题的轮廓,然后对这个算法逐步求精,把它的某些步骤扩展成更详细的步骤。在细化过程中,一方面加入详细算法,一方面明确数据,直到根据这个算法可以写出程序为止。自顶向下,逐步求精的方法符合人类解决复杂问题的思维方式,用先全局后局部,先整体后细节,先抽象后具体的过程开发出的程序,层次结构清晰,容易阅读、理解和测试。

所谓模块化,是将一个大任务分成若干个较小的部分,每一部分承担起一定的功能,称为“功能

模块”。各个模块都可以分别由不同的人编写程序和调试，便于组织人力完成较复杂的任务。

自顶向下、逐步细化和模块化技术三者是紧密结合的，是结构化程序设计方法不可分割的三个要点。

1.1.7 程序设计风格

由于程序设计的依据是算法设计的结果，因此程序的质量主要取决于设计。但是，程序设计的风格也在很大程度上影响着程序的质量。一个优秀的程序员应该养成良好的编程习惯。下面就列举一些有关编程风格的问题。

- (1) 选择合适的标识符名字。
- (2) 添加良好的注解。
- (3) 注意程序的视觉效果。
- (4) 在程序中都有数据说明。
- (5) 每条语句应该尽可能简单明了，能直截了当地反映程序员的意图。
- (6) 对所有的输入数据都进行校验，以确保输入数据的有效性。

1.1.8 C 语言的特点

C 语言是一种成功的系统描述语言，用 C 语言开发的 UNIX 操作系统就是一个成功的范例；同时 C 语言又是一种通用的程序设计语言，在国际上广泛流行。世界上很多著名的计算机公司都成功地开发了不同版本的 C 语言，很多优秀的应用程序也都是用 C 语言开发的，它是一种很有发展前途的高级程序设计语言。主要有以下一些特点：

- (1) 语言表达能力强。
- (2) 语言简洁、紧凑，使用灵活，易于学习和使用。
- (3) 数据类型丰富，具有很强的结构化控制性。
- (4) 语言生成的代码质量高。
- (5) 语法限制不严格，程序设计自由度大。
- (6) 可移植性好。

1.1.9 C 程序的结构

归纳起来 C 语言的程序结构，如图 1-1 所示。

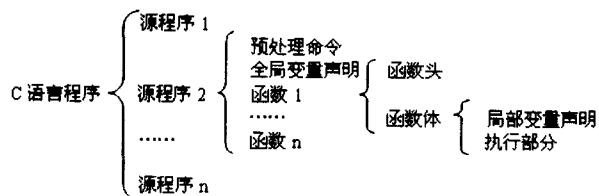


图 1-1 C 语言结构

1.1.10 C 语句概述

C 程序的执行部分是由语句组成的，程序的功能也是由执行语句实现的。C 语句可分为以下五类：

1. 表达式语句

表达式语句由表达式加上分号“;”组成。

其一般形式为：

表达式；

执行表达式语句就是计算表达式的值。例如：

```
x=y+z;    赋值语句
y+z;      加法运算语句,但计算结果不保留,无实际意义
i++;      自增1语句,i值增1
```

2. 函数调用语句

函数调用语句由函数名、实际参数加上分号“;”组成。

其一般形式为：

函数名(实际参数表);

执行函数语句就是调用函数体,并把实际参数赋予函数中定义的形式参数,然后执行被调函数体中的语句,求取函数值(在后面函数章节中再详细介绍)。例如：

```
printf("C Program");    调用库函数,输出字符串
```

3. 控制语句

控制语句用于控制程序的流程,以实现程序的各种结构方式,它们是由特定的语句定义符组成的。

C语言有九种控制语句。可分成以下三类：

条件判断语句：if 语句、switch 语句。

循环执行语句：do while 语句、while 语句、for 语句。

转移语句：break 语句、goto 语句、continue 语句、return 语句。

4. 复合语句

把多个语句用大括号{}括起来组成的一个语句称复合语句。在程序中应把复合语句看成是单条语句,而不是多条语句。例如：

```
{
    x=y+z;
    a=b+c;
    printf("%d%d\n",x,a);
}
```

就是一条复合语句。复合语句内的各条语句都必须以分号“;”结尾,在括号“}”外不能再加分号。

5. 空语句

只有分号“;”组成的语句称为空语句。空语句是什么也不执行的语句。在程序中空语句可用来作空循环体。例如：

```
while (getchar()!='\n');
```

本语句的功能是：只要从键盘输入的字符不是回车符就重新输入。这里的循环体为空语句。

1.2 典型例题分析

【例 1-1】用高级语言编写的源程序必须通过_____程序翻译成二进制程序才能执行,这个二进制程序称为_____程序。

【分析】用高级语言编写的源程序有两种执行方式：一是利用“解释程序”,翻译一条语句,执行一条语句,这种方式不会产生可以执行的二进制程序,例如 BASIC 语言;二是利用“编译程序”一次翻译形成可以执行的二进制程序,例如 C 语言。凡是编译后形成的可执行二进制程序都叫做“目标程序”。

【答案】编译 目标

【例 1-2】关于 C 语言,下列说法中错误的是()。

- A. 数据类型丰富,程序设计简单
- B. 运算符丰富,算法描述方便
- C. 是一种结构化的程序设计语言
- D. 语法结构复杂,难学

【分析】C 语言只有 8 类语句，语法简单易学。

【答案】D

【例 1-3】下列说法中，正确的是（ ）。

- A. 主函数名是由程序设计人员按照“标识符”的规则选取的
- B. 分号和回车符都可以作为一个语句的结束符号
- C. 在程序清单的任何地方都可以插入一个或多个空格符
- D. 程序的执行，总是从源程序清单的第 1 行开始

【分析】主函数名规定为“main”，不是程序设计人员自由选取的；分号是表示语句结束的惟一符号；程序的执行总是由主函数的第 1 条语句开始执行。

【答案】C

【例 1-4】在 Turbo C 的主屏幕中，将当前编辑的源程序以原名存盘，可以选用_____菜单项，也可以直接用热键_____。

【分析】如果选用“File / Save”菜单项，或者使用热键（F12），当前编辑的源程序将以原来的文件名存盘；如果选用“File / Write to”，当前编辑的源程序将以新的文件名存盘。

【答案】File / Save F12

【例 1-5】一个 C 程序是由（ ）。

- A. 一个主程序和若干个子程序组成
- B. 一个或多个函数组成。
- C. 若干过程组成
- D. 若干子程序组成

【分析】C 语言程序可由一个或多个函数组成。C 语言程序无论由多少个函数组成，都有且仅有一个 main 函数，即主函数。不管 main 函数放在前面或后面，程序总是从 main 函数开始执行。

【答案】B

结束语

本章介绍了程序设计的基本概念，讲述了程序设计的一般方法、结构化程序设计思想和程序设计风格，还简单介绍了 C 语言的概况，包括其特点和程序结构等。后面附有例题和练习题帮助读者掌握这些概念。

综合练习一

一、选择题

1. 一个 C 程序的执行是从（ ）。
 - A. 本程序的 main 函数开始，到 main 函数结束
 - B. 本程序文件的第一个函数开始，到本程序文件的最后一个函数结束
 - C. 本程序的 main 函数开始，到本程序文件的最后一个函数结束
 - D. 本程序文件的第一个函数开始，到本程序 main 函数结束
2. 以下叙述正确的是（ ）。
 - A. 在 C 程序中，main 函数必须位于程序的最前面
 - B. C 程序的每行只能写一条语句
 - C. C 语言本身没有输入输出语句
 - D. 在对一个 C 程序进行编译的过程中，可发现注释中的拼写错误
3. 以下叙述不正确的是（ ）。

- A. 一个C源程序可由一个或多个函数组成
 - B. 一个C源程序必须包含一个main函数
 - C. C程序的基本组成单位是函数
 - D. 在C程序中, 注释说明只能位于一条语句的后面
4. C语言规定: 在一个源程序中, main函数的位置().
- A. 必须在最开始
 - B. 必须在系统调用的库函数的后面
 - C. 可以任意
 - D. 必须在最后
5. 一个C语言程序是由().
- A. 一个主程序和若干子程序组成
 - B. 函数组成
 - C. 若干过程组成
 - D. 若干子程序组成
6. C语言源程序的基本单位是().
- A. 过程
 - B. 函数
 - C. 子程序
 - D. 标识符
7. 以下不是C语言的特点的是().
- A. 语言简洁、紧凑
 - B. 能够编制出功能复杂的程序
 - C. 可以直接对硬件进行操作
 - D. 移植性好

二、填空题

- 1. 为解决一个问题而采取的方法和步骤, 称为“_____”。
- 2. 结构化程序设计中的三种基本结构是_____。
- 3. 循环结构分_____和_____两种。
- 4. 计算机能够直接识别和执行的指令集合称为“_____”。
- 5. 程序语言可以分为_____和_____两大类。
- 6. C程序的执行是从_____开始执行的。
- 7. C语言源程序的语句分隔符是_____。

三、程序设计题

- 1. 编写程序, 输出“hello, world!”。
- 2. 编写一个加法运算程序。
- 3. 编写程序, 输入一个数字, 输出其余弦值。
- 4. 编写一个减法运算程序。

第2章 C语言的基本概念

本章回顾了C语言的基本概念,包括常量、变量、标识符、简单数据类型、运算符与表达式和类型转换,这些都是进行C语言编程的重要基础,必须熟练掌握。本章后面还给出了典型例题分析和练习题。

2.1 知识点回顾

2.1.1 常量

在程序执行过程中,其值不发生改变的量称为常量。常量可以分为直接常量和符号常量两种。直接常量,又叫字面常量,其数据类型可以从字面形式直接判断出来。根据不同的数据类型分为:

- (1) 整型常量:如12、0、-3等。
- (2) 实型常量:如4.6、-1.23等。
- (3) 字符常量:如'a'、'b'等。
- (4) 字符串常量:如"How are you?"等。

在C语言中,还可以用一个标识符来表示一个常量,称之为符号常量。

标识符是用来标识变量名、符号常量名、函数名、数组名、类型名、文件名等的有效字符序列。

2.1.2 变量

其值可以改变的量称为变量。一个变量应该有一个名字,在内存中占据一定的存储单元。变量定义必须放在变量使用之前,一般放在函数体的开头部分。要区分变量名和变量值这两个不同的概念。

C语言规定,所有的变量必须先定义其类型,然后才能使用。规定变量必须先定义后使用的目的是:

(1) 可以保证程序中变量名使用的正确性。如已定义了一个变量 count,但在程序中却写成了 conut,在编译时先检查变量名的合法性,发现 conut 未被定义过,按出错处理,因而可以帮助人们查找拼写错误。

(2) 系统根据变量的类型为变量在内存中开辟存储单元。

(3) 变量类型确定后,也就确定了变量的取值范围和可以对其进行的运算。

2.1.3 简单数据类型

数据类型是按被定义变量的性质、表示形式、占据存储空间的大小、构造特点来划分的。

在C语言中,数据类型可分为:基本数据类型,构造数据类型,指针类型,空类型四大类,如图2-1所示。

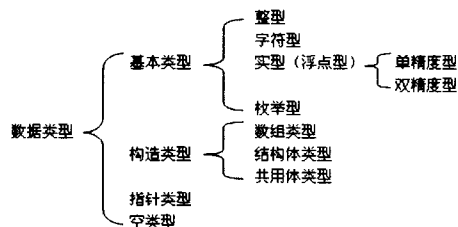


图 2-1 数据类型汇总