

第一部分典型题解析

本部分各章首先概括《教程》相应章节的内容，然后对一些典型题进行解析。解析时，以国际标准 ISO/IEC 9899:1990(E) INTERNATIONAL STANDARD Programming Language C First edition 1990.12. 为基础，对其中为照顾 1990 年以前已有软件而保留的部分一律不予考虑。

第一章 引 论

一、内 容 提 要

程序是用程序设计语言描述的由计算机解题的算法或计算机解题任务。程序设计是将解题任务转变成程序的过程，一般包括分析问题，确定数据结构和算法（对复杂算法需画出流程图），用选定的程序设计语言编写源程序，最后上机调试、运行程序等主要步骤。

(一) C 语言简况

1. C 语言的发展过程

可粗略地分为三个阶段:1970年至1973年为诞生阶段,1973年至1988年为发展阶段,1988年以后为成熟阶段。

- C 语言的诞生 为编写 UNIX 系统,1972 年贝尔实验室的 Dennis M.Ritchie 提出了 C 语言,C 与 UNIX 是一对孪生兄弟。此时的 C 语言是附属于 UNIX 操作系统的。

- C 语言的发展 1977 年 C 的作者发表了不依赖于具体机器系统的 C 语言编译文本《可移植 C 语言编译程序》,从而推动了 UNIX 操作系统在各种机器上的实现以及 UNIX 操作系统的不断发展。

1978 年 Brian W.Kernighan 和 D.M.Ritchie 以 UNIX V7 中的 C 编译程序为基础写出了影响深远的名著《The C Programming Language》,使 C 语言迅速普及。

1978 年 C 的作者发表了《C 语言修订报告》。至此,C 已成为世界上使用最广泛的高级程序设计语言,且不依赖于 UNIX 操作系统而独立存在,《C 语言修订报告》所定义的文本习惯上称为传统 C。

- C 语言的成熟 1988 年美国国家标准协会制定了新的 C 语言文本标准 ANSI C。1990 年国际标准化委员会制定了 C 语言文本标准——标准 C。1990 年 B.W.Kernighan 和 D.M.Ritchie 按照标准重写了他们的经典著作,发表了《The C Programming Language, Second Edition》。

2. C 语言的主要特点

C 在语言方面的特点:(1)表达能力强;(2)流程控制结构化、程序结构模块化;(3)语言简练;(4)使用灵活。

C 在应用方面的特点:(1)C 程序代码质量高,几乎可以与汇

编语言媲美；(2) C 程序包括 C 编译程序本身的可移植性好。

由于 C 语言的上述语言特点和应用特点，使它成为一个实用的通用程序设计语言，特别适用于编写各种与硬件环境相关的系统软件，不愧为一种强有力的系统程序设计语言。

(二) C 程序的基本结构

一个 C 程序可由若干个函数构成，其中必须有且只能有一个以 main 命名的主函数。

(三) C 语言的基本语法单位

基本语法单位是指具有一定语法意义的最小语法成分（即词法分析单位，习惯上称为“单词”）。C 语言的单词分为六类：标识符、关键字、常量、字符串、运算符及分隔符。

常量、字符串及运算符在下章介绍，本节介绍标识符、关键字及分隔符。

1. 标识符

标识符是以字母开头的字母、数字串，有效长度为 31，下横线“_”被看成字母。C 语言中字母区分大小写。标识符可用作程序中的变量名、符号常量名、数据类型名和函数名等。

标识符不能与关键字同名。

2. 关键字

关键字是由编译程序预定义的、具有固定含义的、在组成结构上均由小写字母构成的标识符，关键字又称为保留字。C 语言中标识符的含义不包括关键字。

标准 C 定义了 32 个关键字，它们是：

auto	break	case	char	const	continue
default	do	double	else	enum	extern
float	for	goto	if	int	long

register return short signed sizeof static
struct switch typedef union unsigned void
volatile while

3. 分隔符

分隔符是一类字符，包括空格字符、水平制表和垂直制表符、行符、换页符及注释。分隔符又称为空白字符，空白字符在语法上仅起分隔单词的作用。

程序中相邻的标识符、关键字或常数之间必须有分隔符。

(四) 运行 C 程序的基本步骤

在任何一个编译系统中，运行一个 C 程序的过程包括编辑、编译、连结和运行等基本步骤。

二、典型题解析

例 1.1 什么是程序设计？程序设计包括哪几个主要步骤？

解 程序设计是将解题任务转变成程序的过程。一般包括分析问题，确定数据结构和算法（对复杂算法需画出流程图），用选定的程序设计语言编写源程序，最后上机调试、运行程序等主要步骤。其中最重要的是确定数据结构和算法。

例 1.2 什么是传统 C？什么是标准 C？

解 1978 年 C 的作者 D.M.Ritchie 发表的《C 语言修订报告》习惯上称为传统 C。标准 C 是指国际标准化组织 ISO 所规定的 C 语言（ISO/IEC 9899: INTERNATIONAL STANDARD Programming Languages C）。

B.W.Kernighan 和 D.M.Ritchie 在他们的经典著作《The C

Programming Language, Second Edition 》的附录 A 中列出了传统 C 和标准 C 之间的差异。

有些书将 B.W.Kernighan 和 D.M.Ritchie 在 1978 年著的《The C Programming Language》(第 1 版)或《C 语言修订报告》称为标准 C 是不对的,因为标准不是由哪个人或哪些人制定的,而是由相应的标准化组织制定的。我国的标准化管理条例也作了类似的规定。此规定不仅对计算机行业有效,所有行业都应遵守此规定。

例 1.3 C 语言的主要特点是什么?

解

(1) C 的语言特点:

- 表达能力强 具有丰富的数据类型和运算符;可以直接访问内存物理地址和硬件寄存器;可以表达直接由硬件实现的针对二进制位的运算。
- 流程控制结构化、程序结构模块化 具有各种流程结构的控制语句和多种转移语句;使用函数作为程序的基本单位以及变量的存储类属性,在某种程度上实现了数据的隐藏和模块化程序设计。
- 语言简练 在语言成分的表示方法上尽可能简洁;I/O 操作不是作为 C 的语法成分而是通过 C 库函数实现的 因而使 C 程序简洁,C 编译程序的体积也小。
- 使用灵活 C 语言是非强类型的语言,变量及其值在数据类型上不要求具有严格的对应关系;表达式可以作为语句使用;数组元素和结构成员可用多种形式引用。

(2) C 的应用特点:

- C 程序代码质量高 代码质量是指 C 程序经编译后生成的目标程序在运行速度上和存储空间上开销的大小,运行速度越高,占用的存储空间越少则代码质量越高。一般高级语言相对于

汇编语言而言其代码质量要低得多，但C语言在代码质量上几乎可以与汇编语言媲美。

○程序的可移植性好 可移植性是指将一个程序不作改动或稍加改动就能从一个机器系统上移到另一个机器系统上运行。由于C语言独立于具体机器系统以及C语言的标准化 使得用C语言编写的程序包括C编译程序本身可移植性好。

由于 C语言的上述语言特点和应用特点，使它成为一个实用的通用程序设计语言，既可用于编写系统软件又可编写应用软件，特别适用于编写各种与硬件环境相关的系统软件，不愧为一种强有力的系统程序设计语言。

例 1.4 对下列各小题，若给出的是标识符，请选择 A；若给出的是保留字，请选择 B；否则 请选择 C。将选择填入相应小题前面的（ ）中。

- | | |
|--------------------|---------------------|
| () (1) Book | () (2) I like C |
| () (3) WHILE | () (4) scanf |
| () (5) month_name | () (6) const |
| () (7) 5abc | () (8) student.num |
| () (9) for | () (10) break |
| () (11) main | () (12) 789 |

分析 对于关键字，应该熟记。如果一个串仅由字母组成，只有当组成字符与关键字相同且全部为小写时才是关键字；否则为标识符。对于标识符，则须从标识符的定义及组成规则出发加以判断。一个仅由字母（包括下划线_）和数字组成的串，只要开头是一个字母（包括下划线_）则一定是标识符；否则不是标识符。

此外 主函数名 main 和标准函数名 例如 scanf, printf)都是标识符而不是关键字。

解 本题的答案如下：

- | | |
|--------------------|---------------------|
| (A) (1) Boo | (C) (2) I like C |
| (A) (3) WHILE | (A) (4) scanf |
| (A) (5) month_name | (B) (6) const |
| (C) (7) 5abc | (C) (8) student.num |
| (B) (9) for | (A) (10) break |
| (A) (11) main | (A) (12) _789 |

例 1.5 运行 C 程序的过程一般包括哪些步骤？各个步骤的作用是什么？

解 运行一个 C 程序的过程一般包括编辑、编译、连结和运行等步骤，如图 1.1 所示（图中方框表示数据，圆框表示处理，双线箭头表示数据的流向，单线箭头表示处理过程的重复）。

在适当的文本编辑环境下，例如 UNIX 环境下的 vi，DOS 环境下的 edit，或 C 编译程序提供的集成开发环境的编辑窗口，通过键盘将源程序输入计算机内并以 .c 为文件扩展名存入外存储设备 得到源程序文件。

用所选用的 C 编译程序将源程序文件翻译为二进制代码形式的目标文件，其扩展名在 UNIX 环境下为 .o，在 DOS 环境下为 .obj 如果发现错误，则回到编辑环境下修改源程序，然后再重新执行编译。重复此过程，直到没有错误出现为止。

将编译得到的目标文件与 C 库函数 或系统库函数 装配成为可执行的程序文件，在 DOS 环境下其扩展名为 .exe，在 UNIX 环境下为 .out 或指定的文件名。如果连接不成功，可能要重复编辑、编译和连接，直到连接成功。

以可执行文件名为命令名 或在 C 程序的集成开发环境中选择 Run 命令 执行该程序。如果程序不能正常执行或输出结果不正确则需要重复编辑、编译、连结和运行等步骤，直到正常执行并输出正确结果。其中编译时出现的错误是语法错（静态错）运行时出现的错误是逻辑错（动态错）

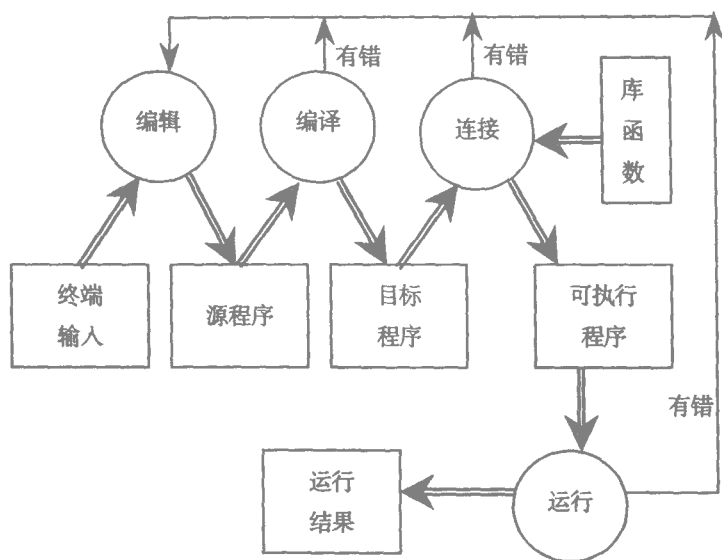


图 1.1 运行 C 程序的一般步骤

习 题 一

1. C 程序的基本结构如何？
2. C 语言的基本语法单位包括哪些元素？C 语言中标识符的含义是什么？对标识符的语法（组成规则、有效长度和大小写）有哪些规定？
3. 下列各小题，若给出的是标识符，请选择 A；若给出的是保留字，请选择 B；否则，请选择 C。将选择填入相应小题前面的（ ）中。

() (1) <code>main</code>	() (2) <code>volatile</code>
() (3) <code>3th</code>	() (4) <code>.exe</code>
() (5) <code>typedef</code>	() (6) <code>CONST</code>

() (7) `_char`

() (9) `auto`

() (11) `case`

() (8) `#include`

() (10) `printf`

() (12) `good!`

第二章基本数据类型和运算

一、内 容 提 要

学习本章的目的是掌握在程序中如何表示数据和对数据的运算。本章内容是 C 语言的重要基础，包括基本数据类型、常量和变量、运算符、表达式以及数据类型转换等。

(一) 数据类型

C 有丰富的数据类型。C 的数据类型分为基本类型和导出类型。基本类型包括整型和浮点型(实数类型)整数和浮点数都是算术量，所以基本类型又称算术类型。整型又包括各种整型、字符型和枚举型；浮点型又分为不同精度的浮点型。从数据的结构上分，基本类型和指针类型是简单类型，其变量只含一个分量(单成员数据)；除指针以外的导出类型是构造类型，其变量含有多个分量(多成员数据)。

(二) 常量、变量和枚举类型

1. 常数的表示

C 的常数包括：整数常量、浮点常量、字符常量和字符串常量。

(1) 整数 [前缀][整数部分][后缀]*

前缀表示数的进制，它可以是 0、0x 或 0X。当前缀省缺时，整数部分是十进制数字 0~9 组成的整数；当有前缀 0 时，整数部分是八进制数字 0~7 组成的整数；当有前缀 0x 或 0X 时，整数部分是十六进制数字（0~9，A~F 或 a~f）组成的整数。

后缀表示类型，它可以是 u 或 U，l 或 L，ul 或 UL。当后缀省缺时，其类型为 int；当有后缀 u 或 U，l 或 L，ul 或 UL 时，整数类型分别是 unsigned，long 或 unsigned long。

(2) 浮点数 [±][整数部分]·[小数部分][e±n][后缀]
组成规则：

- 一个浮点数可以无整数部分或小数部分，但不能二者均无；
- 一个浮点数可以无小数点或指数部分，但不能二者均无。

后缀表示类型，它可以是 f 或 F，l 或 L。当后缀省缺时，浮点数类型为 double；当有后缀 f 或 F，l 或 L 时，浮点数类型分别是 float 或 long double。

(3) 字符常数 字符常数一般是指用一对单引号(单撇号)括起来的一个字符，形式为：'字符'。

括在单引号中的字符可以是图形符号，也可以是转义序列（参阅《教程》中的转义序列列表）。只有转义序列的位型可用于表示字符集中的任何字符，图形符号只能表示某些字符。

字符常数的值是字符的字符码，类型为 char。

(4) 字符串 " 字符序列 "

字符串常数（简称字符串或串）是用一对双引号括起来的一个

* 本书文法中用 [] 括起来的部分，如未作特殊说明，表示可以省缺。

字符序列。

字符串在机内存储时，系统在其末尾添加了一个空字符‘\0’，存储单元比字符串的长度多 1 个字节。

字符串本身是一个字符数组（见第六章），它在程序中表示的值是一个字符型指针（见第七章）。

2. 符号常量和枚举类型

用 `#define` 定义符号常量：`#define 标识符 常量表达式`；

用 `const` 定义符号常量：`const 类型区分符 标识符 = 常量表达式`；

用枚举类型定义符号常量（枚举常量）：

```
enum[ 枚举名 ] { 标识符 [= 常量表达式 ],
                标识符 [= 常量表达式 ] ... } ;
```

枚举符表（括在 ‘{’ 和 ‘}’ 中的部分）中的标识符即被定义的符号常量，称为枚举常量。如果枚举符表中所有的标识符都没有指定值，则规定每个标识符所代表的值从左至右分别为整数 0, 1, 2, ...。枚举标识符后面可以有等号 “=” 和常量表达式，该表达式必须为整数，它用来指定等号左边标识符所代表的值。其后未指定值的枚举标识符的值为前驱枚举常量加 1。

```
enum[ 枚举名 ] { 标识符 [= 常量表达式 ] ,
                标识符 [= 常量表达式 ] ... } ;
```

和

```
enum 枚举名
```

称为枚举区分符（或枚举类型名）。

3. 变量说明

程序中任何标识符，包括变量名，都必须先定义然后才能使用。

变量说明的形式：类型区分符 标识符 [=常量表达式], ...;

(三) 运算符和表达式

运算符执行对运算对象 或称为操作数 的各种操作。C 语言中的运算符见 教程 中运算符的优先级和结合性表。

表达式是由运算符和操作数组成的符合 C 的语法的算式。从本质上说，表达式是对运算规则的描述并按规则执行运算，运算的结果是一个值，称为表达式的值，结果的类型称为表达式（值）的类型。

1. 运算规则

- 算术运算和关系运算：算术运算和关系运算的规则与数学中的运算规则相同，其中关系运算结果的真和假分别用非0和0表示。

- 逻辑运算：逻辑运算规则如表2.1所示。表中1和0分别表示一个非0和0的操作数或运算结果。

表 2.1 逻辑运算规则表

I	J	!!	i & j	i j
0	0	1	0	0
0	1	1	0	1
1	0	0	0	1
1	1	0	1	1

位运算：位运算对操作数的每一个二进位按表2.2所示的规则进行逐位运算。

表 2.2 按位运算规则

I	J	i & j	i j	i ^ j
0	0	0	0	0
0	1	0	1	1
1	0	0	1	1
1	1	1	1	0

- 自增和自减运算 前缀式： $++n$ 或 $-n$ 。先将操作数增 或 减 1，然后取操作数的新值作为表达式的结果。

后缀式： $n++$ 或 $n--$ 。将操作数增（或减）1 之前的值作为表达式的结果 然后操作数增（或减）1。后缀式 $++$ 或 $--$ 的延迟运算是在到达序列点时完成的。C 语言规定，下列运算符或操作可作为序列点：

逻辑与运算符 $\&\&$ ），逻辑或运算符 $\|\$ ）条件运算符 $(? :)$ ，顺序求值运算符 $(,)$ 及一个完整表达式之后。

- 赋值运算：简单赋值：操作数 1 = 操作数 2。将操作数 2 拷贝到操作数 1 中去。

复合赋值：操作数 1 $op=$ 操作数 2。可以理解为展开形式：操作数 1 = 操作数 1 op 操作数 2。其中 op 代表： $+ - * / \% \& | ^ << >>$ 中的一个。

赋值运算的左操作数必须是一个可更改内容的左值表达式。

- 条件运算 条件运算 操作数 1 操作数 2 操作数 3。如果操作数 1 为非 0 值，则操作数 2 被求值，并且将它作为表达式的结果 否则 操作数 3 被求值，并且将它作为表达式的结果。

- 顺序求值运算：顺序求值运算：操作数 1 操作数 2。逗号运算符指出两个操作数按从左至右顺序独立求值。运算结果与操作数 2 相同。

- 强制类型转换：强制类型转换：（类型名）表达式。将表达式的结果强制性地转换为指定的类型。

2. 运算过程中的类型转换

计算表达式时常要引起数据的类型转换。当两个操作数类型不相同时，或当操作数类型为字符型或短整型（包括单操作数）

时 编译程序要对操作数执行类型转换 称之为一般算术转换(简称算术转换)。算术转换的基本规则为值域较窄的那个类型向值域较宽的那个类型转换。算术转换所遵循的转换方向如下(其中的箭头仅表示类型的值域由窄到宽顺序的方向):

char/short→int→unsigned→long→unsigned long→float→double→long double

以下是算术转换中的两个特殊情况(与传统 C 不同):

(1) 表达式中的有符号或无符号的字符和短整数无论什么情况下均被转换为整数, 如果 int 能表示原来类型的值, 则被转换成 int, 否则转换成 unsigned。这种处理称为“整数提升”。

(2) 当一个操作数为 long 而另一个为 unsigned 时, 如果 long 能表示 unsigned 操作数的全部值则将 unsigned 操作数转换成 long, 否则, 将两个操作数都转换成 unsigned long。

各类运算对操作数执行的类型转换如表 2.3 所示。

表 2.3 对操作数执行的类型转换

运算类	运算符	操作数类型	类型转换	结果类型
算术运算	单目: + -	基本类型	算术转换	算术转换后的类型
	双目: + - * /			
	%	整数		
关系运算	< <= == != > >=	基本类型	算术转换	int
逻辑运算	&& !	基本类型	不转换	int
自增和自减运算	++ --	基本类型	不转换	与操作数同

续 表

运算类	运算符	操作数类型	类型转换	结果类型
位运算	~	整数	算术转换 (>> 和 << 只对左操作数)	算术转换后的类型
	& ^			
	>> <<			
条件运算	? :	基本类型		与转换后的左操作数同
顺序求值运算	,	基本类型	两操作数之间不转换	与右操作数同
强制类型转换	(类型名)	基本类型	强制转换	类型强制符指定的类型

注：① 整数提升在运算前进行；② 其余转换在运算时进行。

二、典型题解析

例 2.1 对于下面的表示，请在相应小题前面的空白处填入 A, B, C, D 或 E。A, B, C, D, E 的含义如下：

(A) 字符常数 (B) 整常数 (C) 浮点常数
(D) 字符串 (E) 非法表示

- | | |
|-----------------|---------------|
| () (1) 5.0L | () (2) '\0' |
| () (3) '\108' | () (4) 0UL |
| () (5) '\t' | () (6) 0526 |
| () (7) 1.23e+3 | () (8) 100e2 |
| () (9) 0xfdbu | () (10) "" |

- | | |
|-----------------------|------------------|
| () (11) 3.1415927 | () (12) 0XAFBDE |
| () (13) +32767 | () (14) "" |
| () (15) 0X00FFH | () (16) 34F |
| () (17) "30'40"\n" | () (18) .414 |
| () (19) "My " "god!" | () (20) 'a' |
| () (21) 27. | () (22) 2000L |
| () (23) '\x18' | () (24) 99.F |
| () (25) " | () (26) "Error" |
| () (27) 3.14e ~ 0.5 | () (28) e2 |
| () (29) '\x999' | () (30) '\60' |

分析 从常数表示的语法规则出发加以判断,需考虑每种常数的组成规则、前缀、后缀以及对一些表示的特殊规定。此例中,第(3), (10), (15), (16), (17), (25), (27), 和(29)小题均是非法表示,第28)小题不是常数也不是字符串。第3)小题从表面上看是字符的位型表示 无前缀 x,应为1~3 位八进制数字 而数字 8 不是八进制数字;第10)小题,单引号字符不能用其图形符号表示,而必须用转义序列表示("\");第(15)小题,有 0x 前缀,其后应是十六进制数字,而 H 既不是十六进制数字也不是整数后缀;第16)小题,34 是整数, F 只能作浮点数后缀表示单精度浮点数,不能作整数后缀;第17)小题,本意是想用'表示分 用'表示秒 但字符串中的双引号字符必须使用转义序列表示;第25)小题 单引号间必须有字符;第(27)小题,阶码不能是小数;第28)小题 同时缺小数和整数部分,e2 不是浮点数而是一个标识符;第29)小题,字符的位型表示 (\xhh) 只能是 1~2 位十六进制数,而'\x999'超过了两位。