

C 语言基础教程

闫利平 马林芝 高 贞 编著

Publishing House of Electronics Industry

北京 · BEIJING

内 容 简 介

本书以流行的 Turbo C 为蓝本,由浅入深地介绍了 C 语言的基本概念、编程方法和技巧。全书共分 11 章:第 1 章为 C 语言概述;第 2 章是程序设计基础;第 3、4 章介绍条件分支及循环程序设计;第 5 章为函数;第 6 章为编译与预处理;第 7 章为数组;第 8 章为指针;第 9 章介绍结构体、联合体、枚举类型;第 10 章为文件;第 11 章是综合应用与实践。其中前七章是基础部分,后四章是实用的编程方法和技巧。

全书结构简洁明快,文字流畅易懂;始终以 C 语言编程为主线,并配有大量典型程序实例,突出通过程序学习程序的教学思想。每章末精选有基本题、练习题,作为该章内容的内涵与延伸。本书可作为 C 语言自学用书,亦可作为成人高校、高职教育及普通大专院校的教材和参考书。

未经许可,不得以任何方式复制或抄袭本书之部分或全部内容。

版权所有,翻版必究。

图书在版编目(CIP)数据

C 语言基础教程/阎利平等编著. —北京:电子工业出版社,2000. 11

ISBN 7-5053-6200-3

I. C... II. 阎... III. C 语言—程序设计—教材 IV. TP312

中国版本图书馆 CIP 数据核字(2000)第 70486 号

书 名: C 语言基础教程

编 著 者: 阎利平 马林艺 高 贞

责任编辑: 施玉新

排版制作: 电子工业出版社计算机排版室

印 刷 者:

装 订 者:

出版发行: 电子工业出版社 URL:<http://www.phei.com.cn>

北京市海淀区万寿路 173 信箱 邮编 100036

经 销: 各地新华书店

开 本: 787×1092 1/16 印张: 16.5 字数: 423 千字

版 次: 2000 年 11 月第 1 版 2000 年 11 月第 1 次印刷

书 号: ISBN 7-5053-6200-3
TP·3336

印 数: 5000 册 定价: 20.00 元

凡购买电子工业出版社的图书,如有缺页、倒页、脱页、所附磁盘或光盘有问题者,请向购买书店调换;

若书店售缺,请与本社发行部联系调换。电话 68279077

前 言

C 语言诞生于 20 世纪 70 年代,是伴随 UNIX 操作系统而诞生的。继而,以其崭新的面貌、坚实的步伐,走过了近三十个年头,走进了今天的信息时代。目前,C 语言以其丰富的表达能力,高效的目标代码,成为开发系统软件和应用软件的首选语言。广大非计算机专业人员,迫切希望学习它、掌握它。在这样的背景下,作者结合多年 C 语言教学实践,编成本教材,力求突出自己的特色。

第一 力求深入浅出,简洁明快。多年的教学实践证明,学习计算机语言是为了编程解决实际问题;而通过程序深入学习计算机语言又是一种最有效的学习方法。为此,在讲清基本概念的基础上,精选了约 130 个经过验证的典型程序,并对每个程序的结构、难点、变量设置及运行结果,进行了详细的分析说明。对那些概念容易模糊、程序容易产生错误的地方不仅从正面引导,有时也从反面引伸开去,正反两方面互相印证,加以强调。相信读者在认真思考的基础上,定能够举一反三,触类旁通。

第二 原理和应用相结合,这是作为一本语言教程在深度和广度两方面的体现。在原理指导下的应用,才能是高效的应用;而结合应用讲解原理,才更深入、更易理解。原理和应用的完美结合,展现了 C 语言的潜能,展示了 C 语言的强大生命力。

第三 每章后面均附有基本题和程序设计题以及上机实验,总计约 230 道习题,10 个实习作业。这些习题很有特点,它不是简单地复述前面的概念,也不是前面例子的翻版,而是比例子更具创造性思考和提高的余地。这也是一本好教材的内涵所在。读者如能认真完成这些练习与实习,必将大大加深和巩固所学知识,提高自己的编程能力,必将为自己成为一个专业程序员,打下一个坚实的基础。这些选题,我们以 Turbo C 为蓝本,一一作了调试。

全书共分十一章,前七章是基础部分,后面三章是提高部分,第十一章是实际应用部分。

本书适合于那些希望通过自学较快、较深入地掌握 C 语言的读者,也可作为高职、高专及各类大专院校的教材和参考书,同时,也兼顾到计算机等级考试。

本书由闫利平、马林艺、高贞编写。在本书的编写过程中,得到了河北行政学院及学院信息工程系领导的鼓励和支持,以及孟少沛教授的悉心指导;邱新、靳兰、刘彦肖、王燕、高昆、陶燕红、贾永江等同志,在录入、校对等方面做了大量工作,苏民生、姚国清审校了全书,在此一并表示感谢。

由于作者水平有限,时间仓促,疏漏、不足、错误之处敬请广大读者朋友不吝赐教。我们的电子邮件地址是:lp@yan.com.cn

作 者

目 录

第 1 章 C 语言概述	(1)
1.1 C 语言概述	(1)
1.2 C 语言标识符	(2)
1.3 C 语言程序结构	(2)
1.4 C 语言程序的开发和运行	(3)
习题 1	(5)
上机实验 1	(6)
第 2 章 程序设计基础	(8)
2.1 数据类型	(8)
2.2 常量与变量	(9)
2.2.1 常量	(9)
2.2.2 变量	(11)
2.3 运算符及表达式	(12)
2.3.1 算术运算及算术表达式	(13)
2.3.2 关系运算及关系表达式	(14)
2.3.3 逻辑运算及逻辑表达式	(14)
2.3.4 赋值运算及赋值表达式	(15)
2.3.5 逗号运算及逗号表达式	(16)
2.3.6 条件运算及条件表达式	(17)
2.3.7 求字节数运算	(17)
2.3.8 位逻辑运算	(18)
2.3.9 位移位运算	(18)
2.3.10 位自反运算	(19)
2.4 C 语言的基本语句	(20)
2.5 程序的基本结构与算法	(21)
2.5.1 程序的基本结构	(21)
2.5.2 算法及其描述	(22)
2.6 输入与输出函数	(26)
2.6.1 格式化输入、输出函数 scanf() 和 printf()	(26)
2.6.2 字符输入、输出函数 getchar() 和 putchar()	(34)
2.7 程序综合示例	(37)
本章小结	(39)
习题 2	(39)
上机实验 2	(42)
第 3 章 条件语句	(44)

3.1	if 语句	(44)
3.2	if 语句的嵌套	(46)
3.3	多分支选择语句	(50)
3.4	程序综合示例	(53)
	本章小结	(57)
	习题 3	(58)
	上机实验 3	(62)
第 4 章	循环结构	(63)
4.1	for 语句	(63)
4.2	while 语句	(65)
4.3	do...while 语句	(67)
4.4	循环控制 break 语句和 continue 语句	(69)
4.5	循环的嵌套	(70)
4.6	转向语句和语句标号	(72)
4.7	程序综合示例	(73)
	本章小结	(75)
	习题 4	(76)
	上机实验 4	(81)
第 5 章	函数	(82)
5.1	函数的定义和调用	(82)
5.1.1	概述	(82)
5.1.2	函数的定义	(82)
5.1.3	函数调用	(83)
5.1.4	函数说明	(84)
5.2	变量的作用域和存储类型	(86)
5.3	函数间的数据传递	(89)
5.3.1	利用数据复制方式实现函数间的数据传递	(89)
5.3.2	利用函数返回实现函数间数据传递	(90)
5.4	函数嵌套调用和递归调用	(91)
5.5	内部函数和外部函数	(93)
5.5.1	C 语言源文件	(93)
5.5.2	内部函数	(93)
5.5.3	外部函数	(94)
5.5.4	多个源文件的连接	(94)
5.6	程序综合示例	(94)
	本章小结	(98)
	习题 5	(99)
	上机实验 5	(105)
第 6 章	编译预处理	(107)
6.1	宏定义	(107)

6.1.1	无参宏定义	(107)
6.1.2	有参宏定义	(108)
6.2	文件包含	(109)
6.3	条件编译	(111)
6.4	综合示例	(112)
	本章小结	(115)
	习题 6	(115)
	上机实验 6	(117)
第 7 章	数组	(118)
7.1	一维数组	(118)
7.1.1	一维数组的定义	(118)
7.1.2	一维数组元素的引用	(119)
7.1.3	一维数组的初始化	(119)
7.1.4	一维数组程序举例	(120)
7.2	多维数组	(121)
7.2.1	二维数组的定义	(121)
7.2.2	二维数组元素的引用	(121)
7.2.3	二维数组初始化	(122)
7.2.4	二维数组程序举例	(122)
7.3	字符数组	(123)
7.3.1	字符数组的定义	(123)
7.3.2	字符数组的初始化	(124)
7.3.3	字符数组的引用	(124)
7.3.4	字符数组程序举例	(124)
7.3.5	字符串处理函数	(126)
7.4	数组作为函数参数	(128)
7.5	程序综合示例	(129)
	本章小结	(133)
	习题 7	(134)
	上机实验 7	(139)
第 8 章	指针	(140)
8.1	指针与变量	(140)
8.1.1	指针变量的定义	(140)
8.1.2	指针变量的赋值	(141)
8.2	指针运算	(142)
8.2.1	指针的算术运算	(142)
8.2.2	指针的关系运算	(143)
8.2.3	指针变量的赋值	(143)
8.3	指针与数组	(144)
8.3.1	数组指针	(144)

8.3.2	字符指针	(148)
8.3.3	指针数组	(148)
8.4	指针与函数	(150)
8.4.1	函数指针	(150)
8.4.2	指针函数	(151)
8.4.3	命令行参数	(152)
8.5	程序综合示例	(152)
	本章小结	(161)
	习题 8	(162)
	上机实验 8	(167)
第 9 章	结构体、联合体与枚举类型	(168)
9.1	结构体类型与变量	(168)
9.1.1	结构体类型的定义	(168)
9.1.2	结构体变量的定义	(168)
9.1.3	结构体变量的引用	(169)
9.1.4	结构体初始化	(170)
9.2	结构体数组	(171)
9.2.1	结构体数组定义	(171)
9.2.2	结构体数组成员的引用	(172)
9.3	结构体指针	(173)
9.4	结构体与函数	(174)
9.5	利用结构体变量构成链表	(176)
9.5.1	链表概念	(176)
9.5.2	动态分配内存	(176)
9.5.3	建立链表和显示链表	(177)
9.5.4	链表的插入与删除	(178)
9.6	联合体	(180)
9.7	枚举类型	(181)
9.8	用户定义类型	(183)
9.9	程序综合示例	(184)
	本章小结	(186)
	习题 9	(187)
	上机实验 9	(190)
第 10 章	文件	(191)
10.1	文件的打开与关闭	(191)
10.2	文件的顺序读写	(192)
10.3	文件的随机读写	(197)
10.4	文件的错误检测	(199)
10.5	综合程序示例	(200)
	本章小结	(202)

习题 10	(203)
上机实验 10	(206)
第 11 章 综合应用与实践	(208)
11.1 确定功能	(208)
11.2 定义核心数据结构	(208)
11.3 程序的模块划分	(209)
11.3.1 初始化模块	(209)
11.3.2 游戏主模块	(209)
11.3.3 退出游戏模块	(209)
11.4 源程序	(210)
11.5 调试	(222)
附录 A Turbo C 集成环境 (IDE)	(227)
附录 B Turbo C2.0 常用库函数	(232)
附录 C ASCII 代码表	(241)
附录 D 错误信息中英文对照	(242)
附录 E C 语言程序设计考试要求	(246)
附录 F 部分习题参考答案	(248)
参考文献	(253)

第1章 C语言概述

1.1 C语言概述

计算机的程序设计语言经历了从机器语言、汇编语言到高级语言的发展过程。C语言属于高级语言，但是由于C语言同时具有低级语言的特性，所以许多人又把C语言称为中级语言。

在计算机刚诞生的时候，人们还没有发明一种工具用来比较方便地指挥计算机工作，计算机程序是直接计算机能识别的二进制指令来书写的。这种由二进制指令组成的程序设计语言就是机器语言。机器语言是直接对计算机硬件产生作用的，所以不同型号的计算机的机器语言不一样，使得机器语言很难被人们掌握和推广，一般只有少数计算机专家或专业技术人员才使用。

汇编语言实际上是一种符号化的机器语言。在汇编语言中，每一条机器指令对应一个符号的指令。比如，在机器语言中，二进制数10110110代表加法运算，可以用英文单词ADD代表，比一长串二进制数要简洁直观并且易于记忆。但是，用汇编语言编写的程序需要翻译成机器语言才能被计算机执行。

由于汇编语言和机器语言都是面向机器的语言，而且在程序的书写形式上很难反映出程序设计者的思路，因此人们逐渐发明了与人类自然语言非常接近的高级程序设计语言。高级语言不仅易学、易用，而且写出的程序更加简练，便于在不同型号的机器上运行。但是用高级语言编写的程序不能被计算机直接理解和运行，而必须先由这种语言的编译或解释程序翻译成机器指令，然后再让计算机执行这些机器指令。

C语言是目前最流行、也是最优秀的一种高级程序设计语言。20世纪70年代初，美国贝尔实验室在研制UNIX操作系统时，设计出了C语言。从此，UNIX操作系统和C语言像一对孪生姐妹，以自己崭新的面貌引起人们广泛的关注。后经不断地改进与完善，而得以迅速成长、成熟，并显示了强大的生命力。它不仅具有高级语言的易学、易用、可移植性强等特点，而且具有低级语言的执行效率高、可对硬件直接操作等优点。这一重要特征使C语言可以大大提高运算效率，其在硬件、软件中的地位如图1-1所示。目前它已成为计算机程序设计语言的主流语种，也是学习和掌握更高层开发工具C++和VC++的基础。目前一些流行的游戏软件、工具软件包括Windows 95操作系统大部分都是由C语言编写的。

相对于新发展的C++等面向对象的程序设计语言，C语言属于面向过程的程序设计语言。

作为高层开发工具的C语言具有以下特点：

1. C语言是一种结构化程序设计语言，适合大型项目的编程和管理。
2. 语言简洁易读。C语言只有32个关键字，它们构成了C语言的全部指令。
3. C语言具有丰富的数据类型。可实现各种复杂的数据结构。
4. C语言具有丰富的运算能力，运算符有34种之多。

5. 可直接对硬件进行操作，具有低级语言的许多功能。
6. 移植性好。C语言程序本身不依存于机器硬件，可从一种环境搬到另一种环境中运行。

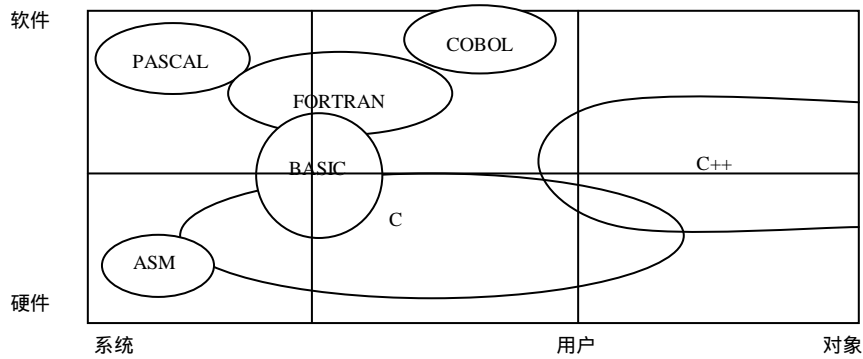


图1-1 C语言在硬件、软件中的地位

1.2 C语言标识符

标识符是一种特定的字符序列，用来标记变量名、常量名、数组名、函数名、结构名、标号和文件名等。

C语言标识符命名规则：

1. 只能由英文字母、数字和下划线组成。
2. 长度为1~32。
3. 必须以英文字母或下划线开头。
4. 严格区分大小写字母，例如：NAME和name是两个不同的标识符。
5. 关键字不能作为标识符。
6. 系统内部使用了一些用下划线开头的标识符，为防止与用户自己定义的标识符冲突，建议用户尽量避免使用下划线开头来定义标识符。

[例1.1] 找出下列符号中合法的标识符

```
“ abc ” so5 file.c Abc
10page printf yellow_red a&b
```

解：合法的标识符有 so5、Abc、printf、yellow_red。

1.3 C语言程序结构

先来看一个最简单的C语言程序：

[例1.2] 一个最简单的程序。

```
main()
{
    printf(" 咏柳\n");
}
```

```
printf("碧玉妆成一树高，\n");
printf("万条垂下绿丝绦。 \n");
printf("不知细叶谁裁出？\n");
printf("二月春风似剪刀。 \n");
}
```

运行结果：

咏柳 碧玉妆成一树高， 万条垂下绿丝绦。 不知细叶谁裁出？ 二月春风似剪刀。
--

由此我们看到C语言程序具有以下主要结构特点：

1. 所有程序都是由一个或多个函数组成。函数是组成C语言程序的基本单位。
2. C语言程序总是从主函数main()开始执行，不论main()函数放在程序的什么位置。
3. C语言程序中的语句用分号“；”终止。一般一个语句占用一行位置。允许一行中多个语句，也可以将一条语句分成几行书写。
4. 用大括号对{}表示程序的结构层次范围，一个完整的程序模块用一对大括号表示其范围。
5. C语言严格区分大小写英文字母。
6. C语言程序可以使用注释。注释部分格式如下：

```
/*注释内容*/
```

可对程序的任何部分作注释。注释可有若干行，但不允许嵌套。

7. C语言程序由函数组成，函数亦可称之为模块，通常每个模块实现一个单一的功能，称之为模块化程序设计。模块化程序设计的思想是，按照自顶向下的原则，把问题逐层分解，直到每个小任务都能用程序的基本结构（顺序、选择、循环）表示为止。在分解过程中，应保证模块的单入口、单出口、完整性和独立性，这种方法称之为结构化程序设计。

1.4 C语言程序的开发和运行

C语言有多种不同的编译器，TURBO C是其中比较有代表性的一种。本书以TURBO C2.0版本为“集成开发”环境，介绍C语言程序的开发和运行。

1. 启动TURBO C

TURBO C一般安装在C:\TC子目录下，在C:\下装有启动UCDOS中文系统的批命令文件UP.BAT，则启动TURBO C的操作步骤如下：

(1) 调入UCDOS中文系统(若程序不涉及汉字的输入输出，可以跳过本步骤)，键入命令：

```
C:\>UP
```

(2) 进入TC子目录，键入命令：

```
C:\>CD TC
```

(3) 运行TC集成环境，键入命令：

C:\TC>TC

此时出现如图1-2所示的屏幕格式。



图1-2 TURBO C菜单示意图

2. 编辑源程序：在出现图1-2后，按F10键，在主菜单项中选择File菜单的新文件New子功能，如图1-3所示，并按回车键，在该编辑器屏幕上出现“ ”光标，此时可以输入并编辑C语言源程序，系统自动给出文件扩展名为“ .C ”。

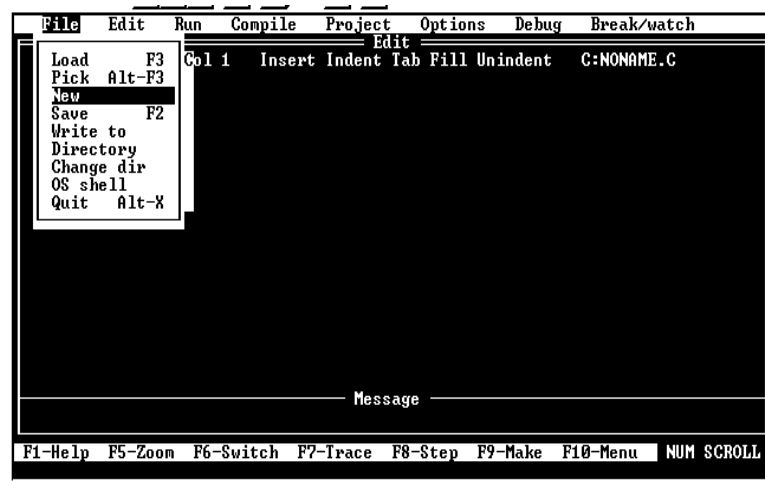


图1-3 File菜单示意图

3. 将源程序编译生成目标程序：对源程序进行语法检查和翻译，一次性地将源程序翻译成机器代码，可以在“ Compile ”菜单中选择“ Compile to obj ”子功能，生成与源程序同名的“ .obj ”目标文件。在编译过程中，若出现错误，则需要察看错误信息后按F6键重新编辑，再编译直到正确为止。

4. 链接目标程序生成可执行程序：在“ Compile ”菜单中选择“ Make EXE File ”子功能，链接各目标程序文件，生成同名的“ .exe ”的可执行文件。

5. 执行程序：在“ Run ”菜单中选择“ Run ”子功能，即可在TURBO C环境中执行程序。也可以脱离C语言编译系统，在DOS环境下键入链接生成的可执行程序文件名或在Windows环境下点击该文件名直接运行。

习 题 1

一、单项选择题

1. 以下不正确的C语言标识符是()。
A. ABC B. Abc C. a_bc D. ab.c
2. 以下正确的C语言标识符是()。
A. %x B. a+b C. a123 D. test !
3. 一个C语言程序的执行是从()。
A. main()函数开始，直到main()函数结束
B. 第一个函数开始，直到最后一个函数结束
C. 第一个语句开始，直到最后一个语句结束
D. main()函数开始，直到最后一个函数结束
4. 在C语言程序中，main()的位置()。
A. 必须作为第一个函数
B. 必须作为最后一个函数
C. 可以放在任意位置
D. 必须放在它所调用的函数之后
5. 一个C语言程序是由()。
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
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. 编辑程序的功能是()。
A. 建立并修改源程序
B. 将C源程序编译成目标程序
C. 调试程序
D. 命令计算机执行指定的程序
8. C编译程序是()。
A. C语言程序的机器语言版本
B. 一种文本编辑程序
C. 将C源程序编译成目标程序的程序
D. 由制造厂家提供的一套启动软件

二、填空题

1. C语言中的标识符只能由三种字符组成，它们是_____、_____、_____。

2. 一个C语言程序有且仅有一个_____函数。
3. 一个C语言程序有_____个main()函数和_____个其他函数。
4. 结构化设计中程序的三种基本结构是_____。
5. C语言程序的执行是从_____开始执行的。
6. 在DOS或Windows环境下,通常C语言源程序文件扩展名是_____,经编译后生成文件的扩展名是_____,经链接后生成文件的扩展名是_____。
7. C语言源程序的语句分隔符是_____。
8. C语言程序开发的四个步骤是_____,_____,_____,_____。

三、编程题

1. 结合Turbo C2.0编译系统,说明C语言程序的开发过程。
2. 从与自然语言比较的角度出发,说明计算机语言(如C)规定的语法非常严格的原因。
3. 编写一个C语言程序,输出以下信息:

```

*   *   *   *   *
*   Very   good!   *
*   *   *   *   *

```

上机实验1

一、题目在Turbo C环境下运行

二、目的要求

1. 初步掌握在Turbo C系统下如何编辑、编译、链接和运行一个。
2. 通过运行简单的,初步了解C语言源程序的特点。

三、实验内容及操作步骤

实验内容:

1. 编辑、链接、运行例1.2程序。
2. 编写习题1中,编程题3题程序,上机验证。

操作步骤:

1. 进入Turbo C系统,编辑C语言源程序。

 进入Turbo C系统

硬盘启动,进入TC子目录,键入TC,回车。

(i)C:\>CD\TC↵

(ii)C:\TC>TC↵或C:\TC>TC filename↵

 编辑C语言源程序

(i) 光条移动到Edit,回车。进入编辑状态。

(ii) 录入“ 例1.2 ”,或自编程序。

(iii) 按F10激活主菜单。File/write to 存盘,起文件名为example1_2.c,或xte1_3.c。

 退出TC环境,查看磁盘目录

(i)File/quit或按Alt_X 退回到操作系统C:\TC>状态下。

(ii)C:\tc>dir *.c ↵

2. 编译源文件

调入源程序

(i) 进入TC。按F10激活菜单。

(ii) File/Load

提示键入文件名：example1_2.c或xte1_3.c

程序example1_2.c或xte1_3.c被调入TC环境。

编译源程序

(i) 按Alt_C 出现Complie菜单，选择Compile to obj或按C，开始编译。

(ii) 结果生成目标文件example1_2.obj或xte1_3.obj

退出TC环境，查看目标文件

```
C:\TC> dir *.obj
```

3. 链接目标文件

进入TC。

按Alt_C 出现Compile菜单，选择Link ExE file或按L。

结果生成可执行文件：example1_2.exe或xte1_3.exe

4. 运行程序

退出TC环境。File/Quit.

```
C:\TC>example1_2 ✓或xte1_3✓
```

运行后查看结果。

说明：

也可以在源程序编辑完成后，按Ctrl_F9，这时将一次完成编译，链接到运行的全过程。按Alt_F5，查看结果。看完后，按任意键返回TC状态。如果源程序有错误，按F6修改源程序后，再重复以上过程。

四、实验报告

见下表：

实验题目	时间	地点	指导教师
目的要求			
实验步骤			
出现问题			
解决方法			
运行结果			
源程序			

班级_____ 学生姓名_____

第2章 程序设计基础

2.1 数据类型

C 语言的数据结构是通过对数据的存储类型、数据类型的定义实现的。存储类型说明数据在内存中的存储位置，将在第 5 章函数部分作详细介绍。数据类型说明数据占内存单元的字节数，即占内存的大小，并规定程序对于该数据的处理方式。本节详细讨论数据类型。

C 语言提供了丰富的数据类型，如图 2-1 所示。

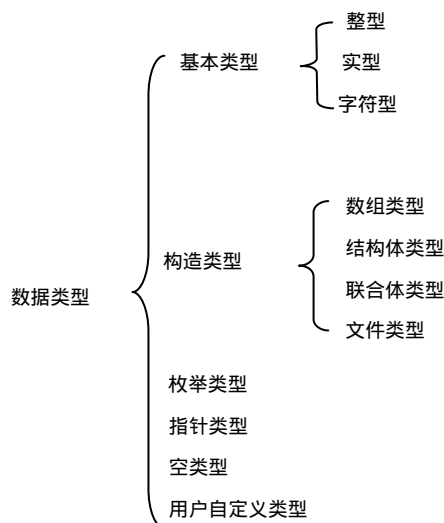


图2-1 C语言数据类型

其中基本类型是基础，由它可以构造其他复杂的数据类型，例如：数组类型就是由若干个具有相同数据类型的数据成员构成的。有关构造类型的内容将在以后章节结合应用陆续介绍，本章重点介绍基本类型。

基本类型有整型、实型、字符型。

整型有基本整型、短整型、长整型、无符号整型、无符号短整型、无符号长整型；实型有单精度实型和双精度实型。对每一种类型由系统提供相应的关键字来标识，用户不能随意修改。基本类型标识符及各基本类型数据在 Turbo C 2.0 环境中通常占用字节数，如表 2-1 所列。

表2-1 各种基本类型关键字及所占字节数

类型关键字	所占字节数	说明
int	2	基本整型(简称整型)
short	2	短整型
long	4	长整型
unsigned	2	无符号整型

类型关键字	所占字节数	说 明
unsigned short	2	无符号短整型
unsigned long	4	无符号长整型
float	4	单精度实型
double	8	双精度实型
char	1	字符型

2.2 常量与变量

在程序运行过程中，其值不能被改变的量称为常量，其值可以改变的量称为变量。

2.2.1 常量

C 语言常用的常量有三类：整型常量、实型常量和字符型常量。另外还有用标识符形式出现的符号常量。

一、整型常量

整型常量即整型数，在 C 语言中有三种不同的表示形式：十进制整数、八进制整数和十六进制整数。

1. 十进制整数：用 10 个不同的数字(0~9)来表示，逢十进位。如 123、-789、0。

2. 八进制整数：用 8 个不同的数字(0~7)来表示，逢八进位，在写法上要加前缀“ 0 ”(零)。如 0123 表示八进制数 123，即 $(123)_8=1 \times 8^2+2 \times 8^1+3 \times 8^0=83$ ，等于十进制数 83。

3. 十六进制整数：用 16 个不同的数字(0~9, a~f)来表示的数，逢十六进位，在写法上要加前缀“ 0x ”。如 0x123，代表十六进制数 123，即 $(123)_{16}=1 \times 16^2+2 \times 16^1+3 \times 16^0=291$ ，等于十进制数 291。

需要注意的是：10，010，0x10 是数值不同的整数，它们的十进制数分别是 10，8 和 16。

整型数可分为长整型数(long int)、短整型数(short int)和无符号整型数(unsigned int)等若干种。长整型数在写法上要加一个后缀“ L ”，如 123L，0123L，0x123abL 等，无符号整型在写法上要加一个后缀“ U ”，如 123U，0123U，0x123abU 等。

整型数又可以是整数和负数，分别在数值的前面加正号或负号。正号一般可以省略。下面是不同进位制数的正数和负数：

123，-123，0123，-0123，0x789，-0x789

虽然数有不同的进位制表示法，但同值的数在计算机中的内部表示是一样的。即 16、020、0x10 在计算机中的内部表示都相同。

二、实型常量

实型常量即实数，在 C 语言中又称浮点数。实数有两种表现形式：

1. 十进制实数：由数字和小数点组成(注意必须加小数点)。如 0.123，.123、123.0，123.，0.0 都是十进制形式。