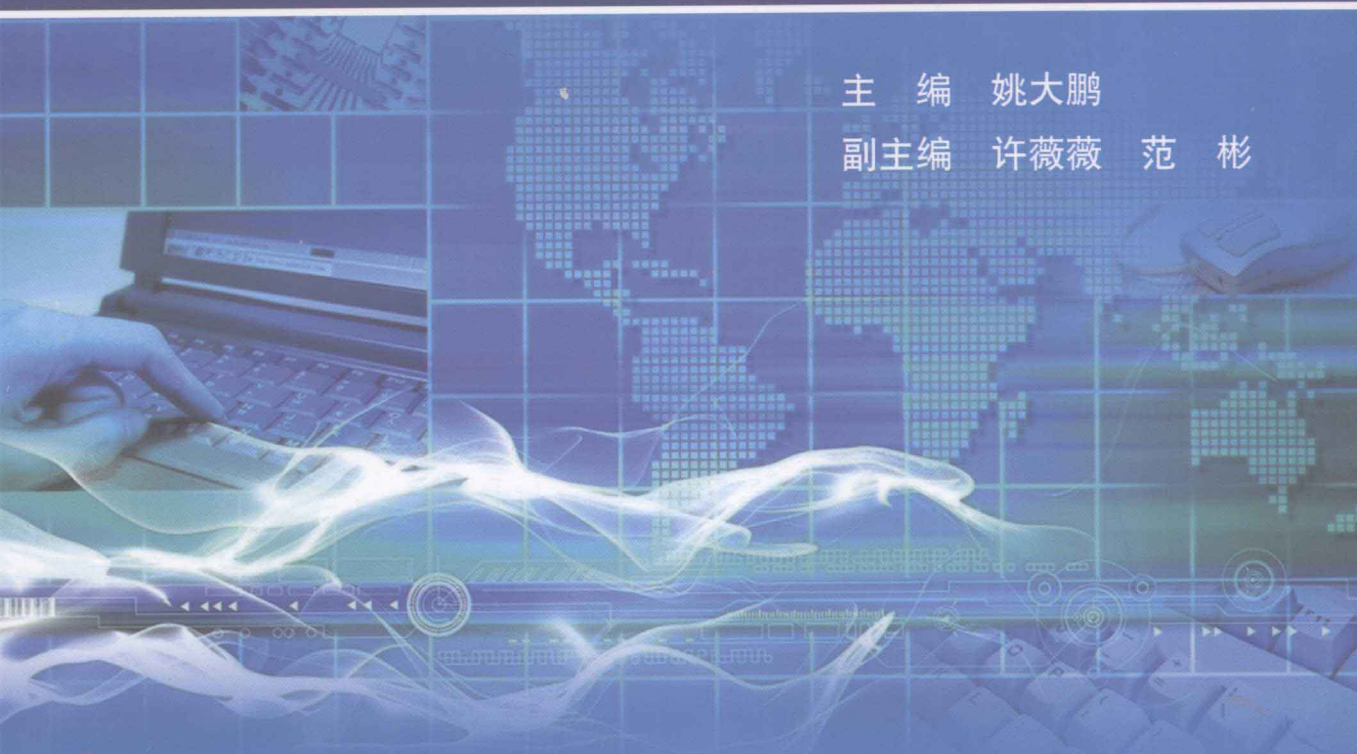
普通高等院校“十二五”规划教材

C语言实践教程

主　编　姚大鹏
副主编　许薇薇　范　彬



中国水利水电出版社
www.waterpub.com.cn

普通高等院校“十二五”规划教材

C语言实践教程

主　编　姚大鹏

副主编　许薇薇　范　彬

中国水利水电出版社
www.waterpub.com.cn

内 容 提 要

本书共分 8 章。第 1 章主要是复习性地阐述 C 语言的基本语法与重要的知识点；第 2 章介绍 C 语言的 3 种主要开发环境，并对它们的优劣进行了对比；第 3 章是从软件工程的角度介绍一些常用的设计理论与方法，目的就是要开阔大家的视野，为某些学生的深入发展打下一定的基础；第 4 章是从课程设计的角度比较深入地介绍 C 语言课程设计中经常使用的一些基本技术；第 5 章与第 6 章为大家列举了程度深浅不一的课程设计实例，希望起到抛砖引玉的作用；第 7 章详细列举了 Turbo C 常用函数库，以便大家在课程设计时可以从中实现绝大部分常用功能；第 8 章上机实验为基本语法模块提供了必要的实验内容。

本书可作为理工科院校学生学习 C 语言，进行上机实验和课程设计的参考书。

本书配有免费电子教案，读者可以从中国水利水电出版社网站以及万水书苑下载，网址为：http://www.waterpub.com.cn/softdown/或 http://www.wsbookshow.com。

图书在版编目（CIP）数据

C语言实践教程 / 姚大鹏主编. -- 北京 : 中国水利水电出版社, 2011.2

普通高等院校“十二五”规划教材

ISBN 978-7-5084-8346-7

Ⅰ. ①C… Ⅱ. ①姚… 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(2011)第012688号

策划编辑：崔新勃　责任编辑：张玉玲　加工编辑：刘晶平　封面设计：李　佳

书　名	普通高等院校“十二五”规划教材 C 语言实践教程
作　者	主　编　姚大鹏　副主编　许薇薇　范　彬
出版发行	中国水利水电出版社 （北京市海淀区玉渊潭南路 1 号 D 座　100038） 网址：www.waterpub.com.cn E-mail：mchannel@263.net（万水） sales@waterpub.com.cn 电话：（010）68367658（营销中心）、82562819（万水）
经　售	全国各地新华书店和相关出版物销售网点
排　版	北京万水电子信息有限公司
印　刷	北京市天竺颖华印刷厂
规　格	184mm×260mm　16 开本　13.75 印张　346 千字
版　次	2011 年 2 月第 1 版　2011 年 2 月第 1 次印刷
印　数	0001—4000 册
定　价	24.00 元

凡购买我社图书，如有缺页、倒页、脱页的，本社营销中心负责调换

前　言

语言课程应该注重边学边练，并应该在不断的编程实践中进一步学习与巩固语法知识。但是由于课堂教学环境与时数的限制，很多有关编程的相关理论与技术都难以在语法教材中体现，为了弥补这方面的缺陷，也为了给同学们在课程设计时提供一些指导和帮助，我们编写了《C 语言实践教程》一书。

本书分为上机实验与课程设计两部分内容。其中上机实验部分主要是为了配合语法主教材的学习活动，指导学生对相关章节进行模块练习。而课程设计部分则突破以往相关教材的思路，既从软件工程的宏观角度介绍与软件开发相关的理论与方法，又从课程设计的微观角度介绍一些为了拓展课程设计选题面所必需的技术与方法。当然书中也合理地配置了初、高级两部分实例集合，便于大家模仿和借鉴。为了便于大家独立使用本教材，我们在第 1 章比较详尽地对 C 语言的语法知识进行了归纳与总结，有些问题总结的深度甚至超过一般的语法教材。

由于现在针对 C 语言的主流编程环境有 3 种，为此我们对这 3 种环境都给予一定的介绍，并对这 3 种环境的优缺点给予了点评。不过从我们的长期教学实践经验来看，大家还是应该先把 Turbo C 2.0 编译平台学深学透为好。

本书由姚大鹏任主编并负责第 1～5 章和第 7 章的编写工作，范彬、许薇薇任副主编并负责编写第 6 和第 8 章。

在本书编写过程中参考了很多教材，这些参考内容对我们的编写工作给予了极大的帮助，在此对这些书的作者们表示衷心的感谢。

编　者

2010 年 12 月

目　　录

第 1 章　C 语言概述

1.1　C 语言发展概述

C 语言是比较流行的高级程序设计语言之一。它不但具有一般高级语言的特点，又可以像汇编语言一样，对硬件内存的位、字节直接进行操作，其运行效率非常高。

C 语言的发展与操作系统 UNIX 密不可分，它是在 B 语言的基础上发展起来的，其根源可以追溯到 ALGOL60。

1960 年出现的 ALGOL60 是一种面向过程的高级语言，它离硬件比较远，不适合用来编写系统程序。1963 年剑桥大学推出了 CPL（Combined Programming Language）语言，CPL 语言在 ALGOL60 的基础上更接近硬件，但规模比较大。1967 年剑桥大学的 Matin Richards 对 CPL 语言做了简化，推出了 BCPL（Basic Combined Programming Language）语言。

UNIX 系统的早期版本是用汇编语言编写的。但是汇编语言可读性和可移植性都比较差，效率也不高、编程比较困难，因此在 1970 年，UNIX 开发者——美国贝尔实验室的 Ken Thompson 以 BCPL 语言为基础，设计出更为简单的而且非常接近硬件的 B 语言，并用 B 语言编写了 UNIX 操作系统和绝大多数上层应用程序。

但是 B 语言过于依赖机器、也过于简单，功能很有限。为了克服 B 语言的局限性，1972 年，贝尔实验室的 D.M.Ritchie 在 B 语言的基础上又设计出了 C 语言。

C 语言既保持了 BCPL 和 B 语言的优点（精炼，接近硬件），又克服了它们的缺点（过于简单，数据无类型等）。C 语言最初只是为了描述和实现 UNIX 操作系统而设计的一种工作语言，1973 年，K.Thompson 和 D.M.Ritchie 两人合作把 UNIX 的 90%以上部分用 C 语言改写（即 UNIX 第 5 版）。后来，C 语言又多次做了改进，直到 1975 年 UNIX 第 6 版公布后，C 语言的突出优点才引起人们的普遍注意。到 1977 年出现了不依赖于具体机器的 C 语言编译文本《可移植 C 语言编译程序》，使 C 移植到其他机器时所需要的工作大大简化了。到了 20 世纪 80 年代，C 开始进入其他操作系统，并很快在各类大、中、小和微型计算机上得到了广泛的使用。从而成为当代最优秀的程序设计语言之一。

以 1978 年发表的 UNIX 第 7 版中的 C 编译程序为基础，BrianW.kernighan 和 Dennis M.Ritchie（合称 K&R）合著了影响深远的名著 *The C Programming Language*，该书介绍的 C 语言后来被称为标准 C，成为被广泛使用的 C 语言版本的基础。1983 年，美国国家标准化协会（ANSI）根据 C 语言问世以来各种版本对 C 的发展和扩充，制定了新的标准，称为 ANSI C。ANSI C 比原来的标准 C 有了很大的发展。1988 年 K&R 修改了他们的经典著作 *The C Programming Language*，按照 ANSI C 标准又重写了该书。1987 年，ANSI 再次公布了新的标准——87ANSI C。目前比较流行的 C 编译系统都是以它为基础的。

随着面向对象编程技术的出现，在进一步扩充和完善了 C 语言基础上，又出现了 C++。Java、C++等面向对象语言（第四代语言）是 C 语言的发展。但是，C 是 C++的基础，C++语言和 C 语言在很多方面是兼容的。在掌握了 C 语言后，再进一步学习 C++，就能以一种熟悉

的语法来学习面向对象的语言，从而达到事半功倍的目的。

1.2 C 语言的特点

相对于其他高级语言，C 语言有很多优点，概括起来，其主要特点如下：

（1）语言简洁紧凑，使用方便灵活，运算符丰富。

C 语言一共有 32 个关键字，9 种控制语句，它们构成了 C 语言的全部指令。C 程序比其他程序简练，源程序短，采用的表达方式简洁，书写形式自由，主要用小写字母表示，压缩了一切不必要的成分。

C 语言共有 34 种运算符。C 把括号、赋值、强制类型转换等都作为运算符处理。从而使 C 语言的运算类型极其丰富，表达式类型也多样化，通过灵活使用各种运算符可以实现在其他高级语言中难以实现的运算。

（2）表达能力强。

C 语言可以完成通常要由其指令来实现的算术及逻辑运算，也可以直接处理字符、数字、地址，还能够进行位操作，汇编语言的大部分功能都可以实现。

（3）数据结构丰富，具有现代化语言的各种数据结构。

C 语言具有丰富的数据结构。其数据类型除有整型、实型、字符型、数组类型、指针类型等基本数据结构类型外，还可以构造结构体类型、共用体类型等新的数据类型，也能用来实现各种复杂的数据结构（链表、树、栈等）运算。

（4）C 语言是一种结构化程序设计语言。

结构化程序结构清晰、可读性强，代码质量和运行效率都很高。C 语言具有功能很强的选择、循环等结构化控制语句（如 if-else 语句、while 语句、do-while 语句、for 语句）。函数是构成 C 语言的基本单位，C 语言是以函数形式提供给用户的，用函数作为程序模块以实现程序的模块化。因此，C 语言是结构化的理想语言，符合现代编程风格要求。

（5）可对硬件直接进行操作。

C 语言可以直接访问物理地址，能进行位（bit）操作，能实现汇编语言的大部分功能，可以直接对硬件进行操作。

（6）生成目标代码质量高，程序执行效率高。

相对汇编语言而言，许多高级语言的代码效率要低得多，C 语言则不然。据统计实验表明，针对同一问题，C 语言的代码效率只比汇编语言低 10%~20%。

（7）可移植性好（与汇编语言相比）。

移植是指程序从一个环境不加改动或稍加改动就可以在另一个环境中运行。C 语言标准化程度高，其编译系统已在多种类型的计算机上实现，因此 C 程序移植起来非常容易。基本上不做修改就能用于各种型号的计算机。

1.3 C 语言基本语法概述

1.3.1 C 语言的语法特点

（1）C 语言程序是由函数构成的，其中必须有且只有一个主函数 main()。

（2）函数体是由左右花括号{ }括起来的。

（3）一个 C 程序总是从 main 函数开始执行的。

（4）C 语言中的每个基本语句都以“;”结束。

（5）C 语言书写格式自由，一行可以写一个语句，也可以写多个语句。

（6）C 语言本身没有输入、输出语句。输入和输出操作都是由库函数 scanf 和 printf 等函数来完成的。

（7）用/*...*/可以对 C 程序中的任何部分作注释。

1.3.2　标识符、常量和变量

（1）标识符。标识符由英文字母、数字、下划线组成，且第一个字符必须是字母或下划线，一般不超过 8 个字符。另外，大小写字母的含义不同；还有不能够使用 C 语言中的关键字做标识符；再有用户取名时，应当尽量遵循“简洁明了”和“见名知意”的原则。

（2）常量。常量是在程序运行过程中其值不能被改变的量。

注意：符号常量就是用一个标识符代表一个常量。但在程序中必须用 define 进行说明。例如，“#define　N　30”，这里 N 就是一个字符常量，编程时直接用 N 来表示字符串 30。

（3）变量。在程序的执行过程中其值可以被改变的量。

注意：使用变量前，一定要先定义后使用。因为每一个变量属于一种类型，在编译时为其分配一定的存储单元，并依据此类型检查该变量所进行的运算是否合法。

（4）变量的初始化。就是在定义变量的同时给变量赋予初值。

1.3.3　数据类型

1. 整型数据

（1）整型常量。

1）十进制整数：以数码直接开头的常量是十进制数，如 1234、-234。

2）八进制整数：以 0 开头的常量是八进制数，如 011。

3）十六进制整数：以 0x 开头的常量是十六进制数，如 0x123。

（2）整型变量如表 1-1 所示。

表 1-1　整型变量说明表

整型数据类型	关键字	所占位数	所占字节	数的表示范围
基本型	int	16	2	-215~215-1
短整型	short int	16	2	-215~215-1
长整型	long int	32	4	-231~231-1
无符号整型	unsigned int	16	2	0~216-1
无符号短整型	unsigned short	16	2	0~216-1
无符号长整型	unsigned long	32	4	0~232-1

2. 浮点型数据

（1）浮点型常量。

1）十进制数形式：它是由数字和小数组成。

2）指数形式：如0.00123用指数法可表示为1.23e-3。

注意：字母e或E之前（即尾数部分）必须有数字；同时e或E后面的指数部分必须是整数。

（2）实型变量。

1）单精度（float）：占4个字节，有效位为7位，数值范围为10^{-38}~10^{38}。

2）双精度（double）：占8个字节，有效位为15~16位，数值范围为10^{-308}~10^{308}。

3. 字符型数据

（1）字符型常量。由一对单引号括起来的单个字符。一个字符型常量的值就是该字符集中对应的ASCII码值。

（2）转义字符。它是特殊形式的字符常量，它以“\”开头，如表1-2所示。

表1-2 转义字符

换码序列	意义	换码序列	意义
\n	回车换行	\r	回车
\b	左退一格	\t	横向跳格字符
\f	换页	\0	空值（NULL）
\'	单引号	\"	双引号
\v	竖向跳格	\000	1～3位八进制所代表的字符
\\	反斜线	\xhh	1～2位十六进制所代表的字符

（3）字符型变量。用来存放一个字符的ASCII码值，它在内存占一个字节。字符型变量分为两种类型：一般字符型（char）和无符号字符型（unsigned char）。

（4）字符串常量。由一对双引号括起来的字符序列。C语言中规定以“\0”作为字符串结束标志，字符“\0”由系统自动加入到每个字符串的结尾。

1.3.4 运算符与表达式

C语言的运算符根据运算对象的个数不同可以分为单目运算符、双目运算符和三目运算符。

运算符的优先级是指不同的运算符计算时的先后顺序。

运算符的结合性是指当一个运算对象两侧的运算符的优先级相同时，进行运算处理的结合方向，其结合方向分为自左向右和自右向左。

1.3.4.1 算术运算符及其表达式

算术运算符包括加、减、乘、除及取模5种，分别用+、-、*、/和%表示。

C语言规定：

（1）模运算符%，仅用于整型变量或整型常量。

（2）优先级：乘、除、模优先级高于加、减的优先级。

（3）结合方向：算术运算符的结合方向是从左至右。

算术表达式是由算术运算符、括号及操作对象组成的符合C语言语法规则的表达式。

1.3.4.2 赋值运算符及其表达式

赋值运算符“=”的作用是将一个数据赋给一个变量。

赋值运算符“=”之前加上其他运算符就可构成复合运算符，如+=等。

赋值表达式：<变量> <赋值运算符> <表达式>。

1.3.4.3　增 1、减 1 运算符及其表达式

i++（或 i--）表示在使用该表达式值之后将 i 值加 1（或减 1）。

++i（或--i）表示在使用该表达式值之前将 i 值加 1（或减 1）。

注意：++和--仅适用于变量，不能用于常量或表达式。

1.3.4.4　关系运算符及其表达式

关系运算符均为两目运算符，共有 6 种：>、<、>=、<=、==、!=。前 4 个运算符的优先级高于后 2 个。结合方向是自左向右。

关系运算符要求两个操作数是同一种数据类型。

关系表达式是由关系运算符将两个表达式连接起来的有意义的式子。关系表达式的值是一个逻辑值，即“真”或“假”。用 1 表示“真”，用 0 表示“假”。

1.3.4.5　逻辑运算符及其表达式

逻辑运算符有“&&”（逻辑与）、“||”（逻辑或）和“!”（逻辑非）3 种。其中“&&”和“||”为二目运算符，并为自左向右结合方向；“!”为单目运算符，仅对其右边的对象进行逻辑求反运算。逻辑运算符的操作对象应为零或非零整数值。

逻辑表达式是由逻辑运算符和其操作对象组成的表达式。

1.3.4.6　位运算符

位运算符包括&、|、~、<<、>>、^。在位运算符中，~的优先级最高，其次是<< 和 >>，然后依次是&、^、|。

注意：这些运算符除~之外，均为二目运算符，即要求两侧各有一个运算量；运算量只能是整型或字符型的数据，不能是实型数据。

1.3.4.7　逗号运算符及其表达式

逗号运算符为“,”。逗号表达式是用逗号运算符把两个表达式连接起来。

其一般形式为：<表达式 1>,<表达式 2>

说明：

（1）逗号表达式的执行过程是：先求“表达式 1”的值，再求“表达式 2”的值，“表达式 2”的值就是整个逗号表达式的值。

（2）逗号运算符是所有运算符中级别最低的。

（3）逗号表达式的一般形式可以扩展为：<表达式 1>,<表达式 2>,<表达式 3>,……,<表达式 N>，“表达式 N”的值是整个表达式的值。

1.3.4.8　条件运算符及其表达式

条件运算符是一个三目运算符，它把 3 个表达式组合成一个表达式。

其一般形式为：<表达式 1>?<表达式 2>:<表达式 3>

注意：其执行过程为：先计算<表达式 1>的值，如果该值为真（非 0 值），则计算<表达式 2>的值，并且把该值作为条件表达式的值。若<表达式 1>的值为假（0），则计算<表达式 3>的值，并将该值作为条件表达式的值。

1.3.5 数据的输入与输出

1.3.5.1 字符数据的输入与输出

1. putchar 函数

格式：putchar(c);

功能：向终端输出一个字符。

注意：括号内的 c 可以是单个字符常量或单个字符变量，也可以是整型变量。

2. getchar 函数

格式：getchar();

功能：从键盘上接收输入的一个字符。

1.3.5.2 数据按格式输入与输出

1. printf 函数

格式：printf("格式控制",输出表列);

说明："格式控制"是用双引号括起来的字符串。包括格式说明、普通字符和转义字符。其中格式说明符是由"%"和格式符组成，其作用是将要输出的数据转换为指定格式后输出。常用的格式符如表 1-3 所示。

表 1-3 格式控制符

格式字符	功能
d	按十进制形式输出带符号的整数（正数前无+号）
o	按八进制形式无符号输出（无前导 0）
x	按十六进制形式无符号输出（无前导 0x）
u	按十进制无符号形式输出
c	按字符形式输出一个字符
f	按十进制形式输出单、双精度浮点数（默认 6 位小数）
e	按指数形式输出单、双精度浮点数
s	输出以"\0"结尾的字符串
ld	长整型输出
lo	长八进制整型输出
lx	长十六进制整型输出
lu	按无符号长整型输出
m 格式字符	按宽度 m 输出，右对齐
-m 格式字符	按宽度 m 输出，左对齐
m,n 格式字符	按宽度 m，n 位小数，或截字符串前 n 个字符输出，右对齐
-m,n 格式字符	按宽度 m，n 位小数，或截字符串前 n 个字符输出，左对齐

2. scanf 函数

格式：scanf("格式控制",地址表列);

功能：用来输入任何类型的数据，可以同时输入多个相同类型或不同类型的数据。

说明：

（1）"格式控制"的含义同 printf 函数。

（2）“地址表列”是由“&+变量名”组成，多个地址之间用“,”分隔。

（3）当输入多个整型或浮点型数据时，例如：当输入 3 个整数时，其格式控制部分可以为“%d%d%d”，此时输入的数据之间用空格（一个或多个）、回车键 Enter 或跳格键 Tab 分隔都是合法的；若格式控制部分为“%d，%d，%d”时，必须用“,”作为分隔，即要求输入数据的格式必须与“格式控制”的情况完全一样。

1.3.6　分支语句

1.3.6.1　C 语句概述

C 语句分为 5 类：

（1）控制语句：完成一定的控制功能。

1）if()-else（条件语句）。

2）for()（循环语句）。

3）while()（循环语句）。

4）do- while()（循环语句）。

5）continue（结束本次循环语句）。

6）break（中止执行 switch 或循环语句）。

7）switch（多分支选择语句）。

8）goto（转向语句）。

9）return（从函数返回语句）。

（2）函数调用语句：由一次函数调用加一个分号构成一个语句。例如：

```
printf("This is a C statement.");
```

（3）表达式语句：由一个表达式加一个分号构成一个表达式语句。例如：

```
a>b;
```

（4）空语句：由单独一个分号组成。

（5）复合语句：用{ }把一些语句括起来成为复合语句，又称为分程序。

1.3.6.2　3 种基本结构

（1）顺序结构：是最简单的 C 程序语句，执行时按从上到下的顺序依次执行。

（2）选择结构：是通过对一个特定条件的判断来选择一个分支执行，常见的语句类型为 if-else 语句和 switch 语句。

（3）循环结构：是在给定的条件下，重复执行某段程序，直到不满足条件为止。它包括三种类型的语句：while 语句、do-while 语句和 for 语句。

1.3.6.3　条件语句

条件语句的 3 种形式：

（1）if 语句的第一种形式：

格式：if(表达式) <语句>

功能：首先计算表达式的值，若表达式的值为“真”（为非 0），则执行语句；若表达式的值为“假”（为 0），则不执行语句。

（2）if 语句的第二种形式：

格式：if(表达式) <语句 1>

　　else　　<语句 2>

功能：首先计算表达式的值，若表达式的值为“真”（为非0），则执行语句1；若表达式的值为“假”（为0），则执行语句2。

（3）if语句的第三种形式：

格式：if(表达式1) <语句1>

else if(表达式2) <语句2>

else if(表达式3) <语句3>

⋮

else if(表达式n) <语句n>

else <语句n+1>

功能：首先计算表达式的值，若第n个表达式的值为“真”（为非0），则执行语句n，若所有的表达式的值都为“假”（为0），则执行语句n+1。

注意：

- if语句中的表达式可以是C语言的合法表达式。
- 第二种、第三种格式的if语句中，在每个else前面有一个分号，整个语句结束处也有一个分号。这是由于分号是C语句中不可缺少的部分，这个分号是if语句中的内嵌语句所要求的。
- 在if和else后面可以只含有一个内嵌的操作语句，也可以含有多个操作语句，此时应用花括号“{ }”将几个语句括起来，构成一个复合语句。
- 条件语句中，if的个数一定不小于else的个数，因为每个else的前面必须有一个if与之相对应，但if后不一定有else。
- else总是与它上面最近的且未曾使用过的if相配对，与书写格式无关。

1.3.6.4 开关（switch）语句

switch语句是多分支选择语句，其一般形式如下：

```
switch(表达式)
  { case <常量表达式1>:<语句1>
    case <常量表达式2>:<语句2>
                ⋮
    case <常量表达式n>:<语句n>
    default:<语句n+1>
  }
```

switch语句的执行过程是：首先计算switch后面圆括号内表达式的值，若此值等于某个case后面的常量表达式的值，则转向该case后面的语句去执行；若表达式的值不等于任何case后面的常量表达式的值，则转向default后面的语句去执行；如果没有default部分，则将不执行switch语句中的任何语句，而直接转到switch语句后面的语句去执行。

注意：

- switch后面圆括号内的表达式，必须是整型量表达式，既可以是整型或字符型，也可以是枚举型。
- case后面必须有空格，紧接着是常量表达式。
- 同一个switch语句中的所有case后面的常量表达式的值都必须互不相同。

- switch 语句中的 case 和 default 的出现次序是任意的，也就是说 default 也可以位于 case 的前面，且 case 的次序也不要求按常量表达式的大小顺序排列。
- 由于 switch 语句中的“case <常量表达式>”部分只起标号的作用，而不进行条件判断，所以，在执行完某个 case 后的语句后，将自动转到该语句后面的语句去执行，直至遇到 switch 语句的右花括号或“break”语句为止，而不再进行条件判断。所以在执行完一个 case 分支后，一般应跳出 switch 语句，转到下一条语句执行，这样可在一个 case 的结束后，下一个 case 开始前，插入一个 break 语句，一旦执行到 break 语句，将立即跳出 switch 语句。
- 每个 case 的后面既可以是一个语句，也可以是多个语句，当是多个语句的时候，也不需要用花括号括起来。
- 多个 case 的后面可以共用一组执行语句。

1.3.7 循环语句

1.3.7.1　用 goto 语句和 if 语句构成循环

一般形式：

```
语句标号：语句;
        …
        if (条件表达式) goto　语句标号;
```

说明：语句标号必须用标识符表示，goto 语句与 if 语句一起构成循环结构；当条件表达式成立时，重复执行语句标号到 if 语句之前的内容。

注意：goto 语句的用法不符合结构化原则，一般不宜采用。

1.3.7.2　while 语句

格式：

```
while (表达式)
    <语句>
```

功能：当表达式的值为非 0 时，执行 while 语句中的循环体；当表达式的值为 0 时，结束循环，继续执行循环体下面的语句。

注意：循环体如果包含一个以上语句，应该用花括号括起来，以复合语句的形式出现；否则 while 语句范围只到 while 后面第一个分号处。

在循环体中应有使循环趋向于结束的语句，即设置修改循环条件的语句。

1.3.7.3　do-while 语句

格式：

```
do
    <语句>
    while(表达式)
```

功能：先执行一次指定的语句，然后判断表达式的值，当表达式的值为非零（“真”）时，返回重新执行该语句，如此反复，直到表达式的值等于 0 为止，此时循环结束。

注意：

- 循环体部分如果有多个语句，则必须用左右花括号括起来，使其形成复合语句。
- 用 while 语句和用 do-while 语句处理同一问题时，若二者的循环体部分一样，其结果也一样。但在 while 后面的表达式一开始就为假（0 值）时，两种循环的结果是不同的。

1.3.7.4 for语句

格式：for (表达式1;表达式2;表达式3)

<循环体语句>

执行过程：先计算“表达式1”的值；然后计算“表达式2”的值，若结果为真（非0），则执行后面的循环体中的各语句；若为假，则结束循环；进行“表达式3”的计算，至此完成一次循环；再次计算“表达式2”的值，开始再次循环，直到计算“表达式2”的值为0，中止循环。

注意：

- for语句中条件测试总是在循环开始时进行。如果循环体部分是由多个语句组成的，则必须用左、右花括号括起来，使其成为一个复合语句。for语句中的表达式1和表达式3既可以是一个简单的表达式，也可以是逗号连接的多个表达式，此时的逗号作为运算符使用。在逗号表达式内按自左至右的顺序求解，整个逗号表达式的值为其中最右边的表达式的值。for语句的一般形式中的“表达式1”可以省略，此时应在for语句之前给循环变量赋初值。
- 省略表达式1时，其后的分号不能省略。如果表达式2省略，即不判断循环条件，循环无终止地进行下去。也就是认为表达式2始终为真。表达式3也可以省略，但此时程序设计应另外设法保证循环能正常结束。可以省略表达式1和表达式3，只有表达式2，即只给循环条件。3个表达式都可以省略，如for(; ;) 语句相当于while(1) 语句。即不设初值，不判断条件（认为表达式2为真值），循环变量不增值，无终止地执行循环体。
- 表达式1可以是设置循环变量初值的赋值表达式，也可以是与循环变量无关的其他表达式。表达式3也可以是与循环控制无关的任意表达式。
- 表达式2一般是关系表达式（如i<=100）或逻辑表达式（如a<b&&x<y），但也可以是数值表达式或字符表达式，只要其值为非0，就执行循环体。

1.3.7.5 循环的嵌套

在一个循环内又完整地包含另一个循环，称为循环的嵌套。循环的嵌套分为双重循环和多重循环。

1.3.7.6 break语句和continue语句

（1）break语句。

格式：break;

功能：break语句可以用于switch语句或循环语句中。在switch语句中，其作用是跳出switch语句，转入switch外的下一个语句；在循环语句中，其作用是跳出该层循环，转到下一个语句。

注意：break语句不能跳出多层循环，如果需要跳出多重循环，可以用goto语句实现。几种循环中，主要是在循环次数不能预先确定的情况下使用break语句，在循环体中增加一个分支结构。当某个条件成立时，由break语句退出循环体，从而结束循环过程。

（2）continue语句。

格式：continue;

功能：跳过循环体中位于continue语句后面的尚未执行的语句，转去判断是否继续进行下一次循环。

注意：continue语句只结束本次循环，不是终止整个循环的执行。而break语句则是结束循环，不再进行判断。如果continue语句是循环体的最后一句，则不起任何作用。

1.3.8　数组

1.3.8.1　一维数组的定义、引用和初始化

1. 一维数组的定义

格式：类型说明符　数组名[常量表达式];

功能：定义一个一维数组，常量表达式的值就是数组元素的个数。

注意：

- 数组名的规定和常量名相同，遵守标识符命名规则。
- 数组名后面是用方括号括起来的常量表达式，不能有圆括号。
- 常量表达式表示元素个数，即数组的长度。
- 数组元素的下标是从 0 开始的。下标最大值为常量表达式值减 1。
- 常量表达式中可以包括常量和符号常量，不能包含变量。
- 数组必须先定义后使用。

2. 一维数组元素的引用

数组元素表示形式为：数组名[下标];

3. 一维数组的初始化

（1）给数组 a 各元素赋予初值。int a[10]={0,1,2,3,4,5,6,7,8,9}。

（2）可以只给一部分元素赋初值，后几个元素值为 0。

（3）如果想使一个数组中全部元素值为 0，可以写成：

```
int a[10]={0,0,0,0,0,0,0,0,0,0}
```

或

```
int a[10];
```

（4）在全部数组元素赋初值时，可以不指定数组长度。 例如：

```
int a[5]={1,2,3,4,5};
```

或

```
int a[ ]={1,2,3,4,5};
```

1.3.8.2　二维数组的定义、引用和初始化

1. 二维数组的定义

格式：类型说明符　数组名[常量表达式 1][常量表达式 2];

功能：定义一个二维数组。“常量表达式 1”是数组元素的行数，“常量表达式 2”是数组元素的列数。

在 C 语言中，二维数组元素排列的顺序为按行存放，即在内存中先顺序存放第一行元素，再存放第二行元素。

2. 二维数组的引用

二维数组元素的表示形式为：数组名[下标][下标];

注意：

- 在引用二维数组时，必须是逐个元素引用，不能引用整个数组。
- 下标可以是整型常量、整型表达式。
- 数组元素可以出现在表达式中，也可以被赋值。
- 在使用数组元素时，应该注意下标值应在已定义的数组的范围内。