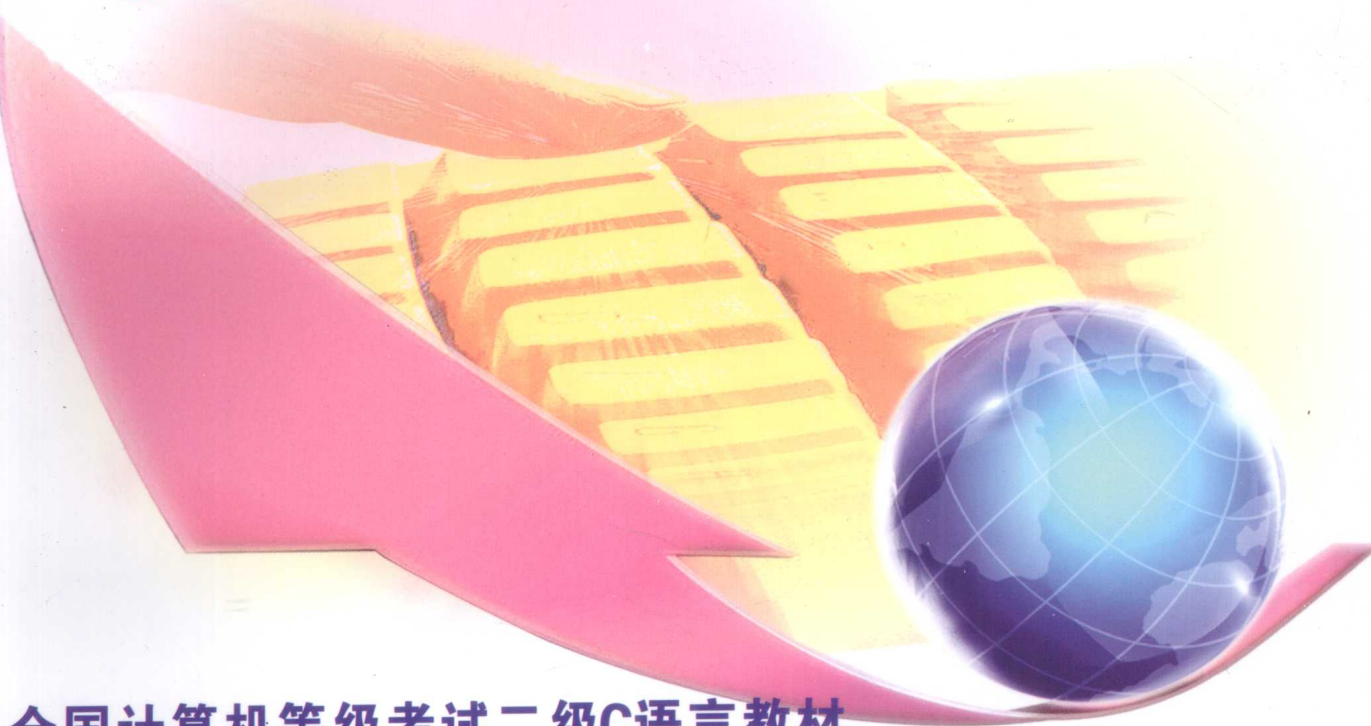




天行健教育

www.edu-exam.net



全国计算机等级考试二级C语言教材

C 语言程序设计

National Computer Rank Examination

天行健教研组 编

天行健教育专用教材



江西科学技术出版社

全国计算机等级考试二级 C 语言教材

C 语言程序设计

主 编：蒋金良

编 委：（按姓氏笔画排名）

丁宁虹 付中华 刘 华 李小明

张 侃 邱 霞 姚玉生 徐忠良

温润良 赖华平

 江西科学技术出版社

图书在版编目 (CIP) 数据

全国计算机等级考试二级 C 语言教材. C 语言程序设计 / 蒋金良编. —南昌:
江西科学技术出版社, 2010.8

ISBN 978-7-5390-3990-9

I. ①全… II. ①蒋… III. ①C 语言—程序设计—水平考试—教材 IV. ①TP312

中国版本图书馆 CIP 数据核字 (2010) 第 167974 号

国际互联网 (Internet) 地址:

<http://www.jxkjcs.com>

选题序号: ZK2010533

图书代码: B10023-101

全国计算机等级考试二级 C 语言教材 C 语言程序设计

天行健教研组 编

出版发行 江西科学技术出版社

社 址 南昌市蓼洲街 2 号附 1 号

邮编: 330009 电话: (0791) 6623491 6639342 (传真)

印 刷 江西和平印刷厂

经 销 各地新华书店

开 本 188mm×265mm 1/16

字 数 163 千字

印 张 14.25

版 次 2010 年 8 月第 1 版 2010 年 8 月第 1 次印刷

书 号 ISBN 978-7-5390-3990-9

定 价 128.00 元 (共五册)

赣版权登字-03-2010-283

版权所有, 侵权必究

本书凡属印装质量问题, 可向承印厂调换

前言

毫无疑问, 21 世纪是信息化的时代, 我们的生活正一天天被网络和计算机所包围和渗透。身处在这样的一个时代中, 是我们的幸运, 但也对我们提出了更高的要求。新时代的人才, 掌握一些计算机的操作和知识显然是必不可少的条件之一。

全国计算机等级考试 (National Computer Rank Examination, NCRE), 是用于考查应试人员计算机应用知识与技能的全国性计算机水平考试体系。它是由教育部考试中心主办, 是计算机类考试中规模最大、考试人数最多的考试。全国每年有超过 1000 万考生参加这个考试, 它也是排在高考和大学英语等级考试之后的全国第三大考试。通过 NCRE 考试, 不仅能够掌握丰富的计算机知识, 证明自身的能力与价值, 同时也可以为今后工作、晋升和深造打下良好的基础。例如, 在职称评定中, 计算机等级考试的证书可以直接作为计算机能力水平的认证。本书是针对全国计算机等级考试二级 C 语言科目所编写的辅导教材。

随着面向对象编程和可视化集成开发工具的发展, 很多曾经非常流行的编程语言的影响力正在下降, 甚至逐渐消失, 但有一种语言例外, 它就是 C 语言。C 语言是 1972 年由美国的 Dennis Ritchie 设计发明的, 并首次在 UNIX 操作系统的 DEC PDP-11 计算机上使用。C 语言发展非常迅速, 而且成为最受欢迎的语言之一, 主要因为它具有强大的功能。它的功能丰富, 表达能力强, 使用灵活方便, 应用面广, 目标程序效率高, 可移植性好, 既具有高级语言的优点, 又具有低级语言的许多特点。在全球最权威的编程语言评价组织 TOIBE 的排行榜中, C 语言一直以来都是最受欢迎、使用范围最广的两种编程语言之一。

到目前为止, 几乎所有的高等院校都将 C 语言列为重要基础科目之一, 不仅在计算机专业开设 C 语言课程, 而且在非计算机专业也开设了 C 语言课程。全国计算机等级考试、全国计算机应用技术证书考试、计算机技术与软件专业技术资格 (水平) 考试等专业或非专业的计算机水平考试都将 C 语言纳入了考试范围。由于 C 语言牵涉到的某些概念比较抽象, 规则繁多, 容易出错, 因此一本讲解透彻、条理清晰、易于理解的教材是学习过程中必不可少的。本书正是基于这样的目的而编写的, 我们组织中国科技大学、合肥工业大学、西安交通大学、贵州大学等重点高校的一线 C 语言教师, 按照全国计算机等级考试二级 C 语言考试大纲的要求, 耗费了近一年的时间打造而成。本书针对零基础初学者的特点和认识规律, 精选内容, 分散难点, 降低台阶, 以丰富的例题和通俗易懂的叙述阐明复杂的概念, 并力求做到内容新颖、概念清晰、通俗易懂。本书中的实例源代码均由编者在 VC++ 6.0 环境下调试通过, 读者可以自行上机实验, 以更好地掌握实际编程技巧。

本书在编写过程中得到了许多高校教师和等级考试辅导专家的大力支持, 为本书的编写提出了许多宝贵的意见, 在此表示衷心的感谢。由于编者水平有限, 书中难免有缺漏不当之处, 热切期望得到广大专家和读者的批评指正!

天行健教研组

2010 年 8 月

目 录

第一章 C语言的入门知识	1
1.1 简单C语言的格式	1
1.2 标识符、常量和变量	2
1.2.1 认识标识符	2
1.2.2 常量	3
1.2.3 变量	4
1.3 整型数据	4
1.3.1 整型常量的表示方法	4
1.3.2 整型变量	10
1.4 实型数据	11
1.4.1 实型常量	11
1.4.2 实型变量	11
1.5 算术表达式和赋值表达式	12
1.5.1 算术运算符和算术表达式	12
1.5.2 强制类型转换	13
1.5.3 赋值运算符和赋值表达式	13
1.5.4 复合赋值表达式	14
1.6 逗号运算符和自加、自减运算符	15
1.6.1 逗号运算符和逗号表达式	15
1.6.2 自加和自减运算符	15
1.7 位运算	17
课后习题	19
第二章 顺序结构	23
2.1 语句	23
2.2 数据输出	24
2.2.1 printf函数的调用形式	24
2.2.2 printf函数的格式说明	26
2.2.3 printf函数说明	28
2.3 数据输入	29
2.3.1 scanf函数的调用形式	29
2.3.2 scanf函数的格式说明	31
课后习题	32
第三章 选择结构	36
3.1 关系运算	36
3.1.1 C语言的逻辑值	36
3.1.2 关系运算符和关系表达式	36
3.2 逻辑运算	37

3.2.1	逻辑运算符	37
3.2.2	逻辑表达式	37
3.3	if语句	38
3.3.1	if语句结构	38
3.3.2	含else的if语句	40
3.3.3	if-else-if语句结构	41
3.3.4	嵌套的if条件语句	42
3.4	条件表达式、switch语句、goto语句	43
3.4.1	条件表达式	43
3.4.2	switch语句	44
3.4.3	语句标号和goto语句	46
	课后习题	47
第四章	循环结构	53
4.1	while语句	53
4.1.1	循环概述	53
4.1.2	while语句	53
4.2	do-while语句	56
4.3	for语句	58
4.4	循环结构的嵌套	60
4.5	break语句和continue语句	61
4.5.1	break语句	61
4.5.2	continue语句	62
	课后习题	63
第五章	字符	70
5.1	字符型常量	70
5.1.1	字符编码	70
5.1.2	字符常量	70
5.2	字符型变量	72
5.3	字符的输入与输出	72
5.3.1	使用printf输出字符	72
5.3.2	使用scanf输入字符	73
5.3.3	使用putchar输出字符	74
5.3.4	使用getchar输入字符	74
	课后习题	75
第六章	函数	79
6.1	函数的定义与返回值	79
6.1.1	函数的定义	79
6.1.2	函数的返回值	80
6.2	库函数	81
6.3	调用函数	81

6.3.1 函数的调用方式	81
6.3.2 函数调用时的语法要求	82
6.3.3 函数的递归调用	83
6.4 函数声明与数据传递	84
6.4.1 函数声明的形式	84
6.4.2 函数声明的位置	85
6.4.3 函数的参数传递与返回值	85
6.5 标识符的作用域与函数的存储分类	87
6.5.1 局部变量	87
6.5.2 全局变量	88
6.5.3 用extern说明函数	90
6.5.4 用static说明函数	90
课后习题	90
第七章 指针	97
7.1 变量地址与指针	97
7.1.1 变量的地址	97
7.1.2 指针变量	98
7.2 指针变量的操作	98
7.2.1 定义指针变量	98
7.2.2 给指针变量赋值	99
7.2.3 对指针变量操作	101
7.3 指针与函数	104
7.3.1 传值与传址	104
7.3.2 函数返回地址值	106
7.3.3 函数指针	107
课后习题	107
第八章 数组	113
8.1 一维数组	113
8.1.1 一维数组的定义	113
8.1.2 一维数组的初始化	114
8.1.3 一维数组的引用	114
8.1.4 程序举例	115
8.2 二维数组	116
8.2.1 二维数组的定义	116
8.2.2 二维数组的初始化	117
8.2.3 二维数组的引用	118
8.2.4 程序举例	119
课后习题	121
第九章 数组与指针、数组与函数	126
9.1 数组与指针	126

9.1.1 一维数组与指针	126
9.1.2 二维数组与指针	128
9.2 数组与函数	131
9.2.1 一维数组与函数	131
9.2.2 二维数组与函数	133
9.2.3 指针数组与函数	134
课后习题	135
第十章 字符串	141
10.1 字符串的基本概念	141
10.1.1 字符串常量	141
10.1.2 用字符数组处理字符串常量	141
10.1.3 用指针处理字符串常量	143
10.2 字符串操作	144
10.2.1 字符串输入输出	144
10.2.2 字符串处理函数	146
10.3 字符串操作	149
10.3.1 二维字符数组处理字符串数组	149
10.3.2 指针数组处理字符串数组	149
10.3.3 对main函数的深入说明	151
课后习题	152
第十一章 结构体和共同体	161
11.1 用户自定义类型	161
11.2 结构体	162
11.2.1 结构体类型的说明	162
11.2.2 结构体类型的变量、数组和指针变量的定义	163
11.2.3 给结构体变量、数组赋初值	165
11.2.4 引用结构体变量中的成员	165
11.2.5 函数之间结构体变量的数据传递	168
11.3 共同体	170
11.3.1 共同体类型的说明和变量定义	170
11.3.2 共同体变量的引用	171
11.4 链表	173
课后习题	176
第十二章 编译预处理、文件	183
12.1 编译预处理	183
12.1.1 宏定义	183
12.1.2 文件包含	185
12.2 动态内存分配	186
12.2.1 malloc函数	186
12.2.2 calloc函数	187

12.2.3	free函数	187
12.3	文件	188
12.3.1	C文件概述	188
12.3.2	文件指针	188
12.3.3	打开文件（fopen函数）	188
12.3.4	关闭文件（fclose函数）	190
12.3.5	文件操作	190
	课后习题	195
附录一	关键字	203
附录二	C语言运算符优先级	204
附录三	常用字符与ASCII码对照表	206
附录四	C语言库函数	207
附录五	全国计算机等级考试二级C语言考试大纲	216

第一章 C 语言的入门知识

C 语言是国内外广泛使用的一种计算机语言。C 语言语法简洁、紧凑、功能丰富、使用方便灵活、目标程序效率高和可移植性好。用 C 语言编写的程序，称为 C 语言源程序。

通过本章，应该学习和掌握以下内容：

- C 程序的构成与格式
- 标识符的概念与命名规则
- 常量与变量
- 二进制、八进制、十进制、十六进制之间的转换
- 常用数据类型
- 运算符与表达式

1.1 C 语言的基本格式

下面通过介绍几个简单的 C 语言程序，来分析 C 语言程序的特点，进而使读者对 C 语言有一个初步的认识。

例 1.1 求两数之和

```
#include    "stdio.h"
main()
{
    int  a,b,c;          /*定义变量 a,b,c    */
    a = 1;               /* 给变量 a 赋初值 1  */
    b = 2;               /* 给变量 b 赋初值 2  */
    c = a+b;
    printf("%d",c);      /* 输出变量 a,b 的和 */
}
```

这个程序的目的是把两个数加起来，然后输出和值。/*...*/表示对语句的注释。第一行通常称为命令行，命令行必须用“#”号开头，后面不能加分号，因为它不是程序中的语句，一对双引号中的 `stdio.h` 是系统提供的文件名，该文件中包含着有关输入输出函数的信息，调用不同的标准库函数，应当包含不同的文件，以便能包含有关库函数的信息。第二行是程序的开始，程序中有且仅有一个 `main` 函数，并且程序的执行始终从这个 `main` 函数开始的。第三行和第九行是一对大括号“{ }”，这对大括号括起来的是整个函数体的内容。第四行是定义整型变量。`int` 表示变量类型是整型，`a,b,c` 是变量名，后面加一个分号表示 C 语言程序的第一句结束。第五行和第六行都是整型变量的初始化，即给整型变量赋初值。注意“=”表示赋值，等于用两个等号表示“==”。第七行是将整型变量 `a,b` 相加

赋值给整型变量 `c`。最后将 `c` 的内容通过 `printf` 语句输出。注意 `printf` 语句的书写格式，我们在第二章将着重讲解这个函数的相关内容。

由例 1.1 可以看出：

- C 语言是由函数组成的。一个 C 源程序有且仅有一个 `main` 函数，C 语言总是从 `main` 开始执行。
- 一个 C 语言程序由两部分组成：函数的首部（如 `main()`）和函数体（由大括号括起来的部分）。函数体是程序的主要精华部分，一般情况下，函数体分为定义部分和执行部分。
- 任何变量都必须先定义，后使用。例 1.1 中的 `int a,b,c`，为定义变量 `a,b,c`，之后才可以使用它们。绝对不能在未定义的情况下就直接使用，后果会造成编译错误。
- 每个语句后面必须加一个分号。分号“`;`”是 C 程序语句的结束标志。
- C 程序的书写格式很自由，一行可以写几个语句。这一点就是 C 语言的优越性所在，可以在一行写多条语句，只要语句合法即可。
- 可以用 `/*.....*/` 对 C 语言的程序语句进行注释。这个注释可以放在 C 语言函数体的任何一个合法的地方。在程序执行的时候，不会执行这个注释里面的语句。（可以注释多行语句）
- `#include "stdio.h" 或 <stdio.h> 通常称为命令行，它不是 C 语言的程序语句。命令行必须使用 # 开头，末尾不可以加分号，因为它不是 C 程序语句，stdio.h 是系统提供的头文件，包含着输入输出函数的说明信息。在程序中，调用不同的标准库函数，应该包含相应的文件，以使程序含有标准库函数的说明信息。至于调用哪个文件，后面的章节会有所介绍。`

1.2 标识符、常量和变量

1.2.1 认识标识符

在 C 语言中，有很多符号的命名，如变量命名，数组命名，指针命名等等。这些标识的命名都必须遵守一定的规则。按照这个规则命名的符号，在 C 语言中称之为标识符。C 语言中标识符的命名规则是这样的：

第一条：标识符必须由数字、字母和下划线组成。

第二条：标识符的首字符必须是字母或者下划线。

注意标识符的大小写是有区别的。以下的标识符都是正确的：

```
area PI _initial a nve _to3 file_5 _2 B3
```

以下的标识符都是错误的：

```
128c no? r.r q&p
```

你知道它们错在哪里吗？`128c` 的首字符是数字，这个是不允许的；`no?` 的？违反了规则的第一条；`r.r` 和 `no?` 一样也是违反了第一条规则，出现了“`.`”这个符号；`q&p` 同上，出现了“`&`”符号。正确的标识符都必须完全的遵守规则的两条，违反了任何一条，这个符号就是错误的。

另外，对于标识符，不宜冗长，定义时应尽量做到简单明了，让人一看就能明白。

C 语言的标识符分类如下：

1. 关键字

关键字是预先规定的一批具有特定含义的、专门用来说明 C 语言成分的标识符。所有的关键字都是由小写字符组成的。比如变量类型标识符 `int` (整型), `float` (实型, 又称浮点型), `char` (字符型), `double` (双精度浮点型) 以及 `if` 语句中的 `if` 和 `else` 标识符等等, 都已经有了专门的用途了, 不能作为变量名和函数名。C 语言中还有很多的关键字, 具体内容见附录一。

2. 预定义标识符

所谓预定义标识符是指在 C 语言中预先定义并使用了的标识符, 具有特定含义。比如输出 `printf`、输入 `scanf`, 正弦 `sin`、编译预处理命令 `define` 和文件包含 `include` 等等。

3. 用户标识符

用于根据自己需要定义的标识符称为用户标识符。简单的说, 就是用于自己取的名字, 一般用来给变量, 函数和数组命名。命名时, 除了做到按照规则以外, 还要求不能使用关键字, 并且名称要做到见名知意。相对于用户来说, 用户标识符是比较重要的, 因为这个标识符是用户自己设立并且应用的, 所以读者要更加注意这个方面的学习。

注意: 预定义标识符可以作为用户标识符, 只是一旦作为用户标识符了, 原来系统规定的意义就没有了。比如库函数中 `sin` 表示求正弦值, 但是作为用户标识符之后, `sin` 就不再含有这个意义了。

1.2.2 常量

所谓常量就是在程序的运行过程中, 其值不能改变的量。比如一个语句: `a=1`; 那么这个 1 就是常量。在 C 语言中, 常量分为整型常量、实型常量、字符常量和字符串常量。其中, 整型常量和实型常量称为数值型常量。它们的区别是整型常量要求数字不带小数点, 比如 10, 0, -28 等; 而实型常量必须带小数点, 如 24.9, 0.0, -3.14 等。'D'和'd'是字符型常量, 而一连串的字符构成的常量称为字符串常量, 如单词"hello"就属于字符串常量(在后面的章节中会学到)。那么除此之外, 还有一个非常重要的常量叫做符号常量。

我们通过下面一个例子来把握一下这个符号常量的意义:

例 1.2 求圆的面积

```
#define PI 3.14159
#include "stdio.h"
main()
{
    float r, area;
    r=4.8;
    area = PI*r*r;
    printf("area=%f\n",area);
}
```

在这个程序中, `#define PI 3.14159` 就预定义了一个符号常量 `PI`。在 `main` 函数中

只要有 PI 的地方，它的值会自动转化为 3.14159。也就是说在函数体中，你看到了 PI 就相当于看到了 3.14159 了。它们意思等同，是替换与被替换的关系，这种替代关系的定义在 C 语言中称为宏定义。

一般符号常量也必须满足标识符的规则，还有做到见名知意，便于用户来阅读。

1.2.3 变量

和常量相对的就是变量了。所谓变量，指的是在程序运行过程中，其值可以随用户的要求而实时改变的量。注意程序中用到的所有变量都必须有一个名字作为标识，变量的名字由用户任意选取，如例 1.2 中的 `r`, `area`，我们可以给变量 `r` 任意赋值，使其半径的长度不同。

C 语言规定所有的变量都必须先定义，后使用。对变量的定义通常放在函数体内的前部，但也可以放在外部或者复合语句的开头，放在主函数外部的变量我们称之为外部变量。这一点在后面的章节会有详细的讲述。

和常量的分类一样，变量也有整型变量 `int`、实型变量 `float`、字符型变量 `char` 和变量等不同类型的。一般，在定义变量的时候就要根据用户需要来定义它们的类型。

1.3 整型数据

1.3.1 整型常量的表示方法

在 C 语言中，整型常量分为八进制整型常量、十进制整型常量和十六进制整型常量三种形式表示。注意，在 C 语言中没有直接表示二进制的整型常量。

- 十进制整数。用一连串连续的 0~9 的数字表示，例如 234，-57，0 等。这一点和我们在数学中应用的十进制数据是一样的。
- 八进制整数。它也是用一连串连续的数字 0~7 表示，但是它前面必须加一个数字“0”以示区别。例如：017、011、0123 等。八进制和十进制的转换关系是这样的，如 $011 = 1 \times 8^1 + 1 \times 8^0 = 9$ ，即八进制的 011 转换成十进制就是 9。
- 十六进制整数。以 0x(数字 0 和字母 x)或 0X 开头的数，其中的数字可以是 0~9 以及 a~f 或者 A~F，例如 0x56，0xff，0x123。十六进制和十进制的转换关系是这样的，如 $0x123 = 1 \times 16^2 + 2 \times 16^1 + 3 \times 16^0 = 291$ ，即十六进制的 0x123 转换成十进制就是 291。

表 1-1 十进制、二进制、八进制和十六进制对照表

十进制	0	1	2	3	4	5	6	7
二进制	0	1	10	11	100	101	110	111
八进制	0	1	2	3	4	5	6	7
十六进制	0	1	2	3	4	5	6	7
十进制	8	9	10	11	12	13	14	15
二进制	1000	1001	1010	1011	1100	1101	1110	1111
八进制	10	11	12	13	14	15	16	17
十六进制	8	9	A	B	C	D	E	F

下面将详细说明一下这三种进制数据之间的转换方法以及它们在 C 语言中的具体含义和应用:

1. 为什么需要八进制和十六进制?

编程中, 我们常用的还是十进制。

比如:

```
int    a = 100, b = 99;
```

不过, 由于数据在计算机中的表示, 最终以二进制的形式存在, 所以有时候使用二进制, 可以更直观地解决问题。但是二进制数太长了。比如 int 类型占用 4 个字节, 32 位。比如 100, 用 int 类型的二进制数表达将是:

```
0000 0000 0000 0000 0000 0000 0110 0100
```

面对这么长的数进行思考或操作, 没有人会喜欢。因此, C 程序没有提供二进制数的表示方法。

用十六进制或八进制可以解决这个问题。因为, 进制越大, 数的表达长度也就越短。不过, 为什么偏偏是十六或八进制, 而不其它的, 诸如九或二十进制呢?

2、8、16, 分别是 2 的 1 次方, 3 次方, 4 次方。这一点使得三种进制之间可以非常直接地互相转换。八进制或十六进制缩短了二进制数, 但保持了二进制数的表达特点。在下面的关于进制转换的课程中, 你可以发现这一点。

2. 二进制转换为十进制

二进制数第 0 位的权值是 2 的 0 次方, 第 1 位的权值是 2 的 1 次方……所以, 设有一个二进制数: 0110 0100, 转换为十进制为:

$$0 * 2^0 + 0 * 2^1 + 1 * 2^2 + 0 * 2^3 + 0 * 2^4 + 1 * 2^5 + 1 * 2^6 + 0 * 2^7 = 100$$

0 乘以多少都是 0, 所以我们可以直接跳过值为 0 的位:

$$1 * 2^2 + 1 * 2^5 + 1 * 2^6 = 100$$

结果是, 二进制数 0110 0100 转换成十进制数为 100

3. 八进制转换为十进制

八进制就是逢 8 进 1。

八进制数采用 0~7 这八数来表达一个数。

八进制数第 0 位的权值为 8 的 0 次方, 第 1 位权值为 8 的 1 次方, 第 2 位权值为 8 的 2 次方……

所以, 设有一个八进制数: 1507, 转换为十进制为:

同样, 我们也可以直接计算:

$$7 * 8^0 + 0 * 8^1 + 5 * 8^2 + 1 * 8^3 = 839$$

结果是, 八进制数 1507 转换成十进制数为 839

4. 八进制数的表示方法

C 语言中, 如何表达一个八进制数呢? 如果这个数是 876, 我们可以断定它不是八进制数, 因为八进制数中由 0~7 组成, 不可能出 7 以上的阿拉伯数字。但如果这个数是 123、是 567, 或 12345670, 那么它是八进制数还是十进制数, 都有可能。

所以,C 语言规定,一个数如果要指明它采用八进制,必须在它前面加上一个 0,如:123 是十进制,但 0123 则表示八进制。这就是八进制数在 C、C++中的表达方法。

由于 C 语言都没有提供二进制数的表达方法,所以,这里所学的八进制是我们学习的 C 语言的数值表达的第二种进制法。

现在,对于同样一个数,比如是 100,我们在代码中可以用平常的十进制表达,例如在变量初始化时:

```
int a = 100;
```

我们也可以这样写:

```
int a = 0144; /*0144 是十进制的 100; 一个十进制数如何转成八进制,我们后面会学到*/
```

千万记住,用八进制表达时,你不能少了最前的那个 0。否则计算机会通通当成十进制。不过,有一个地方使用八进制数时,却不能使用加 0,那就是我们用于表达字符的“转义符”表达法。

5. 十六进制数转换为十进制数

二进制,用两个阿拉伯数字: 0、1;

八进制,用八个阿拉伯数字: 0、1、2、3、4、5、6、7;

十进制,用十个阿拉伯数字: 0 到 9;

十六进制,用十六个阿拉伯数字……等等,阿拉伯人或说是印度人,只发明了 10 个数字啊!

十六进制就是逢 16 进 1,但我们只有 0~9 这十个数字,所以我们用 A, B, C, D, E, F 或 a, b, c, d, e, f 这五个字母来分别表示 10, 11, 12, 13, 14, 15。

十六进制数的第 1 位的权值为 16 的 0 次方,第 2 位的权值为 16 的 1 次方,第 3 位的权值为 16 的 2 次方……

所以,在第 N (N 从 0 开始)位上,如果是数 X (X 大于等于 0,并且 X 小于等于 15,即: F) 表示的大小为 $X * 16$ 的 N 次方。

假设有一个十六进数 2AF5,那么如何换算成十进制呢?

直接计算就是:

$$5 * 16^0 + F * 16^1 + A * 16^2 + 2 * 16^3 = 10997$$

(别忘了,在上面的计算中, A 表示 10,而 F 表示 15)

现在可以看出,所有进制换算成十进制,关键在于各自的权值不同。

假设有人问你,十进数 1234 为什么是一千二百三十四?你尽可以给他这么一个算式:

$$1234 = 1 * 10^3 + 2 * 10^2 + 3 * 10^1 + 4 * 10^0$$

6. 十六进制数的表示方法

如果不使用特殊的书写形式,十六进制数也会和十进制相混。随便一个数: 9876,就看不出它是十六进制或十进制。

C 规定, 16 进制数必须以 0x 开头。比如 0x1 表示一个十六进制数。而 1 则表示一个十进制。另外如: 0xff, 0xFF, 0X102A, 等等都是十六进制数。其中的 x 也不区分大小写。(注意: 0x 中的 0 是数字 0, 而不是字母 o)。

以下是一些用法示例:

```
int a = 0x100F;
```

```
int b = 0x70 a;
```

至此, 我们学完了所有进制: 十进制, 八进制, 十六进制数的表达方式。最后一点很重要, C++ 中, 十进制数有正负之分, 比如 12 表示正 12, 而 -12 表示负 12,; 但八进制和十六进制只能用无符号的正整数, 如果你在代码中里: -078, 或者写: -0xF2, C 并不把它当成一个负数。

7. 十进制数转换为二进制数

给你一个十进制, 比如: 6, 如果将它转换成二进制数呢?

十进制数转换成二进制数, 这里采用短除法连续除 2 的过程:

把要转换的数, 除以 2, 得到商和余数, 将商继续除以 2, 直到商为 0。最后将所有余数倒序排列, 得到数就是转换结果。

听起来有些糊涂? 我们结合例子来说明。比如要转换 6 为二进制数。

“把要转换的数, 除以 2, 得到商和余数”。

那么:

要转换的数是 6, $6 \div 2$, 得到商是 3, 余数是 0。(不要告诉我你不会计算 $6 \div 3$!)

“将商继续除以 2, 直到商为 0……”

现在商是 3, 还不是 0, 所以继续除以 2。

那就: $3 \div 2$, 得到商是 1, 余数是 1。

“将商继续除以 2, 直到商为 0……”

现在商是 1, 还不是 0, 所以继续除以 2。

那就: $1 \div 2$, 得到商是 0, 余数是 1

“将商继续除以 2, 直到商为 0……最后将所有余数倒序排列”

好极了! 现在商已经是 0。

我们三次计算依次得到余数分别是: 0、1、1, 将所有余数倒序排列, 那就是: 110 了!

6 转换成二进制, 结果是 110。

用短除法表示为, (在计算机中, $\div$ 用 / 来表示)

		余数	
2	6	 0	↑
2	3	 1	
2	1	 1	
	0		

从下往上取余数为 110

请大家对照图、表及文字说明, 并且自己拿笔计算一遍如何将 6 转换为二进制数。

说了半天, 我们的转换结果对吗? 二进制数 110 是 6 吗? 你已经学会如何将二进制数转换成十进制数了, 所以请现在就计算一下 110 换成十进制是否就是 6。

8. 十进制数转换为八、十六进制数

十进制数转换成八进制的方法，和转换为二进制的方法类似，惟一变化：除数由 2 变成 8。

来看一个例子，如何将十进制数 120 转换成八进制数。

用短除法表示：

		余数	
8	120		0
8	15		7
8	1		1
	0		

120 转换为八进制，结果为：170。

十进制数转换成十六进制的方法，和转换为二进制的方法类似，惟一变化：除数由 2 变成 16。

同样是 120，转换成十六进制则为：

		余数	
16	120		8
16	7		7
	0		

120 转换为十六进制，结果为：78。

9. 二、十六进制数相互转换

二进制和十六进制的互相转换比较重要。不过这二者的转换却不用计算，每个 C、C++ 程序员都能做到看见二进制数，直接就能转换为十六进制数，反之亦然。

我们也一样，只要学完这一小节，就能做到。

首先我们来看一个二进制数：1111，它是多少呢？

你可能还要这样计算： $1 * 2^0 + 1 * 2^1 + 1 * 2^2 + 1 * 2^3 = 1 * 1 + 1 * 2 + 1 * 4 + 1 * 8 = 15$ 。

然而，由于 1111 才 4 位，所以我们必须直接记住它每一位表示的数值，并且是从高位往低位记：8、4、2、1。即，最高位的权值为 $2^3 = 8$ ，然后依次是 $2^2 = 4$ ， $2^1 = 2$ ， $2^0 = 1$ 。

记住 8421，对于任意一个 4 位的二进制数，我们都可以很快算出它对应的十进制值。

下面列出四位二进制数 xxxx 所有可能的值（中间略过部分）

仅 4 位的 2 进制数	快速计算方法	十进制值	十六进值
1111	$= 8 + 4 + 2 + 1$	$= 15$	F
1110	$= 8 + 4 + 2 + 0$	$= 14$	E
1101	$= 8 + 4 + 0 + 1$	$= 13$	D
1100	$= 8 + 4 + 0 + 0$	$= 12$	C
1011	$= 8 + 4 + 0 + 1$	$= 11$	B
1010	$= 8 + 0 + 2 + 0$	$= 10$	A
1001	$= 8 + 0 + 0 + 1$	$= 9$	9