第六章　二进制、八进制、十六进制

这是一节“前不着村后不着店”的课。不同进制之间的转换纯粹是数学上的计算。不过，你不必担心会有么复杂，无非是乘或除的计算。

生活中其实很多地方的计数方法都多少有点不同进制的影子。

比如我们最常用的 10 进制，其实起源于人有 10 个指头。如果我们的祖先始终没有摆脱手脚不分的境况，我想我们现在一定是在使用 20 进制。

至于二进制……没有袜子称为 0 只袜子，有一只袜子称为 1 只袜子，但若有两袜子，则我们常说的是：1 双袜子。

生活中还有：七进制，比如星期。十六进制，比如小时或“一打”，六十进制，比如分钟或角度……

6.1 为什么需要八进制和十六进制?

编程中，我们常用的还是 10 进制……必竟 C/C++是高级语言。

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，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 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,C++ 没有提供在代码直接写二进制数的方法。

用 16 进制或 8 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 16 或 8 进制，而不其它的，诸如 9 或 20 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。8 进制或 16 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。

6.2 二、八、十六进制数转换到十进制数

6.2.1 二进制数转换为十进制数

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 10 进制为：

下面是竖式：

0110 0100 换算成 十进制

第 0 位 0 * 20 = 0

第 1 位 0 * 21 = 0

第 2 位 1 * 22 = 4

第 3 位 0 * 23 = 0

第 4 位 0 * 24 = 0

第 5 位 1 * 25 = 32

第 6 位 1 * 26 = 64

第 7 位 0 * 27 = 0 ＋

100

用横式计算为：

0 * 20 + 0 * 21 + 1 * 22 + 1 * 23 + 0 * 24 + 1 * 25 + 1 * 26 + 0 * 27 = 100

0 乘以多少都是 0，所以我们也可以直接跳过值为 0 的位：

1 * 22 + 1 * 23 + 1 * 25 + 1 * 26 = 100

6.2.2 八进制数转换为十进制数

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：

用竖式表示：

1507 换算成十进制。

第 0 位 7 * 80 = 7

第 1 位 0 * 81 = 0

第 2 位 5 * 82 = 320

第 3 位 1 * 83 = 512 ＋

839

同样，我们也可以用横式直接计算：

7 * 80 + 0 * 81 + 5 * 82 + 1 * 83 = 839

结果是，八进制数 1507 转换成十进制数为 839

6.2.3 八进制数的表达方法

C,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 10 进制数，都有可能。

所以,C,C++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 则表示采用八进制。这就是八进制数在 C、C++中的表达方法。

由于 C 和 C++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，CtC++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 10 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 是八进制的 100；一个 10 进制数如何转成 8 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 10 进制。不过，有一个地方使用八进制数时，却不能使用加 0，那就是我们前面学的用于表达字符的“转义符”表达法。

6.2.4 八进制数在转义符中的使用

我们学过用一个转义符'\'加上一个特殊字母来表示某个字符的方法，如：'\n'表示换行(line)，而'\t'表示 Tab 字符，'\"'则表示单引号。今天我们又学习了一种使用转义符的方法：转义符'\'后面接一个八进制数，用于表示 ASCII 码等于该值的字符。

比如，查一下第 5 章中的 ASCII 码表，我们找到问号字符（?)的 ASCII 值是 63，那么我们可以把它转换为八进值：77，然后用 '\77'来表示'?'。由于是八进制，所以本应写成 '\077'，但因为 C,C++规定不允许使用斜杠加 10 进制数来表示字符，所以这里的 0 可以不写。

事实上我们很少在实际编程中非要用转义符加八进制数来表示一个字符，所以， 6.2.4 小节的内容，大家仅仅了解就行。

6.2.5 十六进制数转换成十进制数

2 进制，用两个阿拉伯数字：0、1；

8 进制，用八个阿拉伯数字：0、1、2、3、4、5、6、7；

10 进制，用十个阿拉伯数字：0 到 9；

16 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？

16 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 这五个字母来分别表示 10，11，12，13，14，15。字母不区分大小写。

十六进制数的第 0 位的权值为 16 的 0 次方，第 1 位的权值为 16 的 1 次方，第 2 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 10 进制呢？

用竖式计算：

2AF5 换算成 10 进制:

第 0 位：　5 * 160 = 5
第 1 位：　F * 161 = 240
第 2 位：　A * 162 = 2560
第 3 位：　2 * 163 = 8192 ＋

　　10997
直接计算就是：
5 * 160 ＋F * 161 + A * 162 + 2 * 163 = 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 10 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 * 103 + 2 * 102 + 3 * 101 + 4 * 100

6.2.6 十六进制数的表达方法
如果不使用特殊的书写形式，16 进制数也会和 10 进制相混。随便一个数：9876，就看不出它是 16 进制或 10 进制。
C，C++规定，16 进制数必须以 0x 开头。比如 0x1 表示一个 16 进制数。而 1 则表示一个十进制。另外如：0xff,0xFF,0X102A,等等。其中的 x 也也不区分大小写。(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：10 进制，8 进制，16 进制数的表达方式。最后一点很重要，C/C++中，10 进制数有正负之分，比如 12 表示正 12，而-12 表示负 12，；但 8 进制和 16 进制只能用达无符号的正整数，如果你在代码中里：-078，或者写：-0xF2,C,C++并不把它当成一个负数。

6.2.7 十六进制数在转义符中的使用

转义符也可以接一个 16 进制数来表示一个字符。如在 6.2.4 小节中说的 '?' 字符，可以有以下表达方式：

```
'?'    //直接输入字符
'\77'  //用八进制，此时可以省略开头的 0
'\0x3F' //用十六进制
```

同样，这一小节只用于了解。除了空字符用八进制数 '\0' 表示以外，我们很少用后两种方法表示一个字符。

6.3 十进制数转换到二、八、十六进制数
6.3.1 10 进制数转换为 2 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？

10 进制数转换成二进制数，这是一个连续除 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 ÷ 2，得到商是 3，余数是 0。 （不要告诉我你不会计算 6÷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 ÷ 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 ÷ 2, 得到商是 0，余数是 1 （拿笔纸算一下，1÷2 是不是商 0 余 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！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！

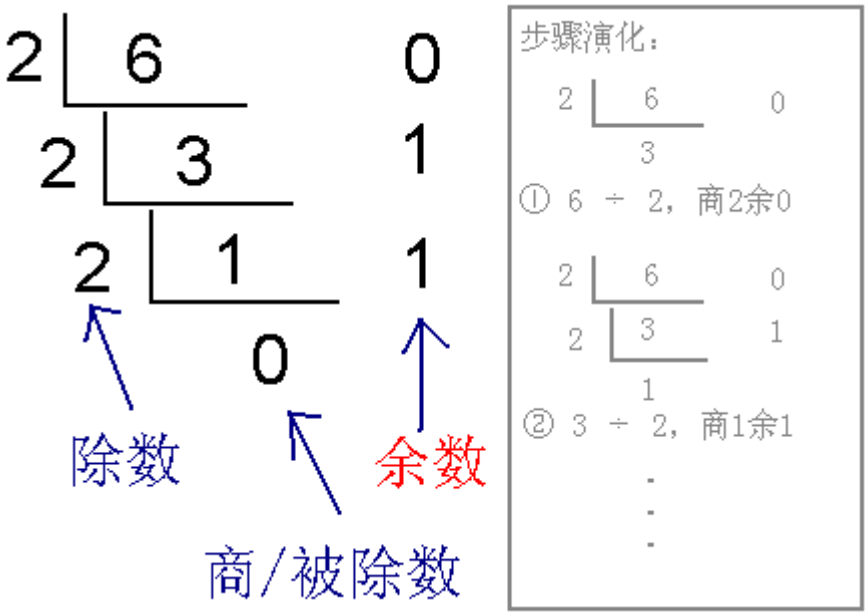
6 转换成二进制，结果是 110。

把上面的一段改成用表格来表示，则为：

被除数	计算过程	商	余数
6	6/2	3	0
3	3/2	1	1
1	1/2	0	1

（在计算机中，÷用 / 来表示）

如果是在考试时，我们要画这样表还是有点费时间，所更常见的换算过程是使用下图的连除：



（图：1）

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 10 进制数了，所以请现在就计算一下 110 换成 10 进制是否就是 6。

6.3.2 10 进制数转换为 8、16 进制数

非常开心，10 进制数转换成 8 进制的方法，和转换为 2 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。

用表格表示：

被除数	计算过程	商	余数
120	120/8	15	0
15	15/8	1	7
1	1/8	0	1

120 转换为 8 进制，结果为：170。

非常非常开心，10 进制数转换成 16 进制的方法，和转换为 2 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 16 进制则为：

被除数	计算过程	商	余数
120	120/16	7	8
7	7/16	0	7

120 转换为 16 进制，结果为：78。

请拿笔纸，采用（图：1）的形式，演算上面两个表的过程。

6.4 二、十六进制数互相转换

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 * 20 + 1 * 21 + 1 * 22 + 1 * 23 = 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的权值，并且是从高位往低位记，：8、4、2、1。
即，最高位的权值为 23 ＝ 8，然后依次是 22 ＝ 4，21＝2， 20 ＝ 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 10 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	= 10	9
....			
0001	= 0 + 0 + 0 + 1	= 1	1
0000	= 0 + 0 + 0 + 0	= 0	0

二进制数要转换为十六进制，就是以 4 位一段，分别转换为十六进制。
如(上行为二制数，下面为对应的十六进制)：

1111 1101 ， 1010 0101 ， 1001 1011
F D ， A 5 ， 9 B

反过来，当我们看到 FD 时，如何迅速将它转换为二进制数呢？
先转换 F：
看到 F，我们需知道它是 15（可能你还不熟悉 A～F 这五个数），然后 15 如何用 8421 凑呢？应该是 8 + 4 + 2 + 1，所以四位全为 1 ：1111。
接着转换 D：
看到 D，知道它是 13，13 如何用 8421 凑呢？应该是：8 + 2 + 1,即：1011。
所以,FD 转换为二进制数，为： 1111 1011

由于十六进制转换成二进制相当直接，所以，我们需要将一个十进制数转换成 2 进制数时，也可以先转换成 16 进制，然后再转换成 2 进制。
比如，十进制数 1234 转换成二制数，如果要一直除以 2，直接得到 2 进制数，需要计算较多次数。所以我们可以先除以 16，得到 16 进制数:

被除数	计算过程	商	余数
1234	1234/16	77	2
77	77/16	4	13 (D)
4	4/16	0	4

结果 16 进制为： 0x4D2

然后我们可直接写出 0x4D2 的二进制形式： 0100 1011 0010。
其中对映关系为：
0100 -- 4
1011 -- D
0010 -- 2

同样，如果一个二进制数很长，我们需要将它转换成 10 进制数时，除了前面学过的方法是，我们还可以先将这个二进制转换成 16 进制，然后再转换为 10 进制。
下面举例一个 int 类型的二进制数：
01101101 11100101 10101111 00011011
我们按四位一组转换为 16 进制： 6D E5 AF 1B

6.5 原码、反码、补码

结束了各种进制的转换，我们来谈谈另一个话题：原码、反码、补码。

我们已经知道计算机中，所有数据最终都是使用二进制数表达。
我们也已经学会如何将一个 10 进制数如何转换为二进制数。
不过，我们仍然没有学习一个负数如何用二进制表达。

比如，假设有一 int 类型的数，值为 5，那么，我们知道它在计算机中表示为：
00000000 00000000 00000000 00000101
5 转换成二制是 101，不过 int 类型的数占用 4 字节（32 位），所以前面填了一堆 0。
现在想知道，-5 在计算机中如何表示？

在计算机中，负数以其正值的补码形式表达。
什么叫补码呢？这得从原码，反码说起。

原码：一个整数，按照绝对值大小转换成的二进制数，称为原码。
比如 00000000 00000000 00000000 00000101 是 5 的 原码。

反码：将二进制数按位取反，所得的新二进制数称为原二进制数的反码。
取反操作指：原为 1，得 0；原为 0，得 1。（1 变 0; 0 变 1）
比如：将 00000000 00000000 00000000 00000101 每一位取反，得 11111111 11111111 11111111 11111010。
称：11111111 11111111 11111111 11111010 是 00000000 00000000 00000000 00000101 的反码。
反码是相互的，所以也可称：
11111111 11111111 11111111 11111010 和 00000000 00000000 00000000 00000101 互为反码。

补码：反码加 1 称为补码。
也就是说，要得到一个数的补码，先得到反码，然后将反码加上 1，所得数称为补码。
比如：00000000 00000000 00000000 00000101 的反码是：11111111 11111111 11111111 11111010。
那么，补码为：
11111111 11111111 11111111 11111010 + 1 = 11111111 11111111 11111111 11111011

所以，-5 在计算机中表达为：11111111 11111111 11111111 11111011。转换为十六进制：0xFFFFFFFB。

再举一例，我们来看整数-1 在计算机中如何表示。
假设这也是一个 int 类型，那么：

1、先取 1 的原码：00000000 00000000 00000000 00000001
2、得反码：　　11111111 11111111 11111111 11111110
3、得补码：　　11111111 11111111 11111111 11111111

可见，－1 在计算机里用二进制表达就是全 1。16 进制为：0xFFFFFF。

一切都是纸上说的……说－1 在计算机里表达为 0xFFFFFF，我能不能亲眼看一看呢？当然可以。利用 C++ Builder 的调试功能，我们可以看到每个变量的 16 进制值。