当书写十进制和八进制数时，我们可以利用下标标注来区别不同数字系统，避免混淆。下标 TEN 表示十进制，EIGHT 表示八进制。

这样，白雪公主遇到的小矮人数量是 7_{TEN} 或 7_{EIGHT}；

卡通人物的手指数是 8_{TEN} 或 10_{EIGHT}；

贝多芬所写的交响曲的数目是 9_{TEN} 或 11_{EIGHT}；

人类手指的数量是 10_{TEN} 或 12_{EIGHT}；

一年中的月份数是 12_{TEN} 或 14_{EIGHT}；

两个星期的天数是 14_{TEN} 或 16_{EIGHT}；

“情人”的生日庆祝会是 16_{TEN} 或 20_{EIGHT}；

一天中的小时数是 24_{TEN} 或 30_{EIGHT}；

拉丁字母表中的字母数是 26_{TEN} 或 32_{EIGHT}；

一夸脱的液体相当于的盎司数是 32_{TEN} 或 40_{EIGHT}；

一副纸牌的张数是 52_{TEN} 或 64_{EIGHT}；

棋盘的格数是 64_{TEN} 或 100_{EIGHT}；

日落大道上最有名的地址是 77_{TEN} 或 115_{EIGHT}；

美式足球场地的面积是 100_{TEN} 或 144_{EIGHT}；

温网首届女子单打的初赛人数是 128_{TEN} 或 200_{EIGHT}；

孟菲斯市的面积是 256_{TEN} 或 400_{EIGHT}。

注意，在上面一系列的八进制数中出现了几个好整数（nice round number），如 100_{EIGHT}、200_{EIGHT} 和 400_{EIGHT}。根据规定，在十进制中，好整数通常是指结尾有若干个零的数。在结尾有两个零的十进制数代表的是 100_{TEN}，而 100_{TEN} 表示 10_{TEN} 乘以 10_{TEN}。在八进制数中，结尾有两个零代表是 100_{EIGHT}，而 100_{EIGHT} 表示 10_{EIGHT} 乘以 10_{EIGHT}（或 8_{TEN} 乘以 8_{TEN}，即 64_{TEN}）。

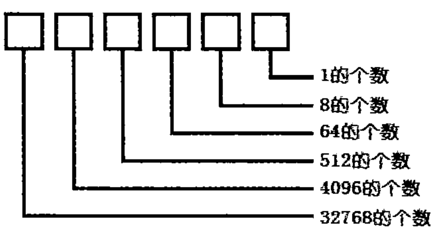
你可能还会注意到，这些好整数 100_{EIGHT}、200_{EIGHT} 和 400_{EIGHT} 在十进制中分别与 64_{TEN}、128_{TEN} 和 256_{TEN} 相等，它们全是 2 的整数次幂。这是非常有意义的。例如，400_{EIGHT} 等于 4_{EIGHT} 乘以 10_{EIGHT} 乘以 10_{EIGHT}，而这里所有的数都是 2 的整数次幂。任何 2 的整数次幂乘以 2 的整数次幂的结果依然是 2 的整数次幂。

下表给出了一些 2 的整数次幂的十进制及其对应的八进制的表示形式。

2 的整数次幂	十进制	八进制
2^0	1	1
2^1	2	2
2^2	4	4
2^3	8	10
2^4	16	20
2^5	32	40
2^6	64	100
2^7	128	200
2^8	256	400
2^9	512	1000
2^{10}	1024	2000
2^{11}	2048	4000
2^{12}	4096	10000

最右列的好整数暗示我们，十进制以外的数字系统可能对使用二进制码有所帮助。

在结构上，八进制数字系统与十进制数字系统并没有什么不同。它们只是在细节上存在一些差异。例如，八进制数中的每个位所代表的值是该位数字乘以 8 的整数次幂的结果。



这样，一个八进制数 3725_{EIGHT} 可以分解成如下形式：

$$3725_{\text{EIGHT}} = 3000_{\text{EIGHT}} + 700_{\text{EIGHT}} + 20_{\text{EIGHT}} + 5_{\text{EIGHT}}$$

同样还可以写成若干种不同的形式，下面是利用十进制的 8 的整数次幂写出的一种形式：

$$
\begin{aligned}
3725_{\text{EIGHT}} = {} & 3 \times 512_{\text{TEN}} + \\
& 7 \times 64_{\text{TEN}} + \\
& 2 \times 8_{\text{TEN}} + \\
& 5 \times 1
\end{aligned}
$$

采用八进制形式的 8 的整数次幂，表现形式如下：

$$
\begin{aligned}
3725_{\text{EIGHT}} = {} & 3 \times 1000_{\text{EIGHT}} + \\
& 7 \times 100_{\text{EIGHT}} + \\
& 2 \times 10_{\text{EIGHT}} + \\
& 5 \times 1
\end{aligned}
$$

还有一种拆分形式：

$$
\begin{aligned}
3725_{\text{EIGHT}} = {} & 3 \times 8^3 + \\
& 7 \times 8^2 + \\
& 2 \times 8^1 + \\
& 5 \times 8^0
\end{aligned}
$$

如果用十进制数计算出结果，会得到 2005_{TEN}。这就是将八进制数转化为十进制数的方法。

我们可以像进行十进制数的加法和乘法那样，对八进制数进行加法或乘法运算。不同的是，我们要采用不同的表来对各个数位进行乘法或加法运算。下面是八进制数的加法表。

+	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
0	0	1	2	3	4	5	6	7
1	1	2	3	4	5	6	7	10
2	2	3	4	5	6	7	10	11
3	3	4	5	6	7	10	11	12
4	4	5	6	7	10	11	12	13
5	5	6	7	10	11	12	13	14
6	6	7	10	11	12	13	14	15
7	7	10	11	12	13	14	15	16

例如，$5_{EIGHT}+7_{EIGHT}=14_{EIGHT}$。我们还可以将两个更长的八进制数按照与十进制相同的方法相加：

$$\begin{array}{r} 135 \\ +\ 643 \\ \hline 1000 \end{array}$$

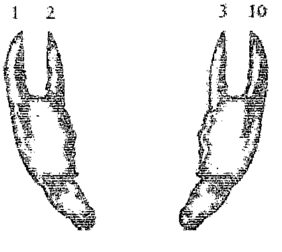
首先要从最右的一列开始，5 加 3 等于 10，结果为 0，进位 1；1 加 3 加 4 等于 10，结果为 0，进位 1；1 加 1 加 6 等于 10。

同样，在八进制中，2 乘以 2 结果依然为 4。但是 3 乘以 3 却不等于 9。那是多少呢？在这里，3 乘以 3 结果是 11_{EIGHT}，其与 9_{TEN} 所代表的数量相等。下面为八进制乘法的乘法表。

×	0	1	2	3	4	5	6	7
0	0	0	0	0	0	0	0	0
1	0	1	2	3	4	5	6	7
2	0	2	4	6	10	12	14	16
3	0	3	6	11	14	17	22	25
4	0	4	10	14	20	24	30	34
5	0	5	12	17	24	31	36	43
6	0	6	14	22	30	36	44	52
7	0	7	16	25	34	43	52	61

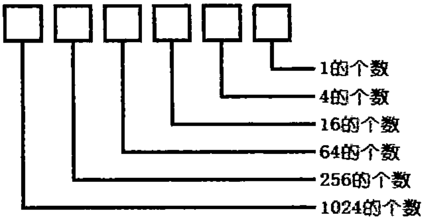
在这里，4×6 等于 30_{EIGHT}，而 30_{EIGHT} 与 24_{TEN} 等价，也就是与十进制中的 4×6 相等。

八进制数字系统与十进制数字系统一样，都是有效的。但是让我们更进一步来看，既然已经为卡通人物开发了一套数字系统，就让我们再制订一套适合龙虾的数字系统吧。实际上龙虾根本没有手指，但是在它们前爪的末端都有螯。适用于龙虾的数字系统是以 4 为基数的四进制（quaternary）数字系统。



四进制数字系统是像这样计数的：0，1，2，3，10，11，12，13，20，21，22，23，30，31，32，33，100，101，102，103，110，等等。

这里我不打算在四进制数上花太多的时间，因为接下来我们即将转入一个更为重要的话题。通过下图我们可以看出四进制中每一位是如何跟 4 的某个整数次幂相对应的。



四进制数 31232 可以写做如下形式：

$$
\begin{aligned}
31232_{\text{FOUR}} = {} & 3 \times 256_{\text{TEN}} + \\
& 1 \times 64_{\text{TEN}} + \\
& 2 \times 16_{\text{TEN}} + \\
& 3 \times 4_{\text{TEN}} + \\
& 2 \times 1
\end{aligned}
$$

等价于：

$$
\begin{aligned}
31232_{\text{FOUR}} = {} & 3 \times 10000_{\text{FOUR}} + \\
& 1 \times 1000_{\text{FOUR}} + \\
& 2 \times 100_{\text{FOUR}} + \\
& 3 \times 10_{\text{FOUR}} + \\
& 2 \times 1
\end{aligned}
$$

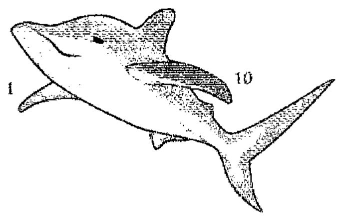
同样可以写成：

$$
\begin{aligned}
31232_{\text{FOUR}} = {} & 3 \times 4^4 + \\
& 1 \times 4^3 + \\
& 2 \times 4^2 + \\
& 3 \times 4^1 + \\
& 2 \times 4^0
\end{aligned}
$$

如果以十进制数的形式计算其结果，我们会发现 31232_{FOUR} 等价于 878_{TEN}。

现在，我们要做一个跳跃并且是最远的一跳。如果我们是海豚，那么就必须用两个鳍来计数。这个数字系统称为以 2 为基数的数字系统，或二进制。这样的话我们似乎只有两个数字了，这两个数分别为 0 和 1。

0 和 1 现在已不是我们要处理的全部问题了，而我们还需要练习一下才能习惯使用二进制数。二进制数最大的问题是数字用完得很快。例如，下图就是海豚如何用它的鳍来计数的例子。



是的，在二进制中，1 的下一个数字是 10。这让人惊讶，但也并不是什么意外。无论使用哪种计数系统，当单个的数字用完时，第一个两位数就是 10。二进制系统这样计数的：

0，1，10，11，100，101，110，111，1000，1001，1010，1011，1100，1101，1110，1111，10000，10001……

这些数看起来似乎很大，但实际上并不是这样的。准确地说，二进制数的长度增长得很快而非数值增大得快：

人类的头的个数是 1_{TEN} 或 1_{TWO}；

海豚身上鳍的个数是 2_{TEN} 或 10_{TWO}；

一个大汤匙中的小茶匙的数目为 3_{TEN} 或 11_{TWO}；

正方形的边数是 4_{TEN} 或 100_{TWO}；

人类一只手的手指数是 5_{TEN} 或 101_{TWO}；

昆虫的腿数是 6_{TEN} 和 110_{TWO}；

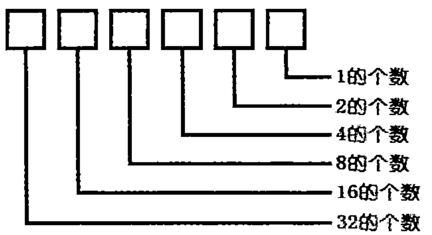
一周中的天数是 7_{TEN} 或 111_{TWO}；

八重奏中演奏者的人数是 8_{TEN} 或 1000_{TWO}；

银河系中行星（包括冥王星）的总数是 9_{TEN} 或 1001_{TWO}；

牛仔帽的重量以加仑计算为 10_{TEN} 或 1010_{TWO}。

在一个多位二进制数中，数字的位置和 2 的整数次幂的对应关系为：



因此，任何一个以 1 开头而后面全是 0 的二进制数一定都是 2 的整数次幂。幂指数就等于这个二进制数中 0 的个数。

以下是 2 的各整数次幂的扩展表，它可以用来说明这条规则。

2 的整数次幂	十进制数	八进制数	四进制数	二进制数
2^0	1	1	1	1
2^1	2	2	2	10
2^2	4	4	10	100
2^3	8	10	20	1000
2^4	16	20	100	10000
2^5	32	40	200	100000
2^6	64	100	1000	1000000
2^7	128	200	2000	10000000
2^8	256	400	10000	100000000
2^9	512	1000	20000	1000000000
2^{10}	1024	2000	100000	10000000000
2^{11}	2048	4000	200000	100000000000
2^{12}	4096	10000	1000000	1000000000000

假定有一个二进制数 101101011010，它可以写成：

$$
\begin{aligned}
101101011010_{TWO} = {} & 1 \times 2048_{TEN} + \\
& 0 \times 1024_{TEN} + \\
& 1 \times 512_{TEN} + \\
& 1 \times 256_{TEN} + \\
& 0 \times 128_{TEN} + \\
& 1 \times 64_{TEN} + \\
& 0 \times 32_{TEN} + \\
& 1 \times 16_{TEN} + \\
& 1 \times 8_{TEN} + \\
& 0 \times 4_{TEN} + \\
& 1 \times 2_{TEN} + \\
& 0 \times 1_{TEN}
\end{aligned}
$$

同样也可以用下面这种方式表示：

$$
\begin{aligned}
101101011010_{TWO} = {} & 1 \times 2^{11} + \\
& 0 \times 2^{10} + \\
& 1 \times 2^{9} + \\
& 1 \times 2^{8} + \\
& 0 \times 2^{7} + \\
& 1 \times 2^{6} + \\
& 0 \times 2^{5} + \\
& 1 \times 2^{4} + \\
& 1 \times 2^{3} + \\
& 0 \times 2^{2} + \\
& 1 \times 2^{1} + \\
& 0 \times 2^{0}
\end{aligned}
$$

如果将各个部分以十进制数的形式相加，会得到 2048 + 512 + 256 + 64 + 16 + 8 + 2，即 $2{,}906_{TEN}$。

为了更简明、更便捷地将二进制数转换为十进制数，你可能会更喜欢借助我准备的

模板来进行转换：

□	□	□	□	□	□	□	□	
×128	×64	×32	×16	×8	×4	×2	×1	
□+	□+	□+	□+	□+	□+	□+	□=	□

这个模板可以将一个数转换最大长度为 8 的二进制数，但是它很容易扩展。在使用模板的时候，将 8 位二进制数填入到上面一行的格子里，每个格子一位。做 8 个乘法运算，然后将结果填入到下面一行的 8 个小格子里。再将这 8 个格子中的数相加就会得到最终结果。下面就举例说明如何得到与二进制数 10010110 相等的十进制数。

1	0	0	1	0	1	1	0	
×128	×64	×32	×16	×8	×4	×2	×1	
128 +	0 +	0 +	16 +	0 +	4 +	2 +	0 =	150

将十进制数转换成二进制数就没那么直接了。下面是一个能够帮你将 0～255 范围内的十进制数转换为二进制数的模板。

□	□	□	□	□	□	□	□
÷128	÷64	÷32	÷16	÷8	÷4	÷2	÷1
□	□	□	□	□	□	□	□

这个转换实际上比看上去要麻烦得多，因此一定要仔细地按照下面的指导来操作。将整个十进制数（小于或等于 255）填入到上面一行最左端的格子中。用第一个除数（128）去除这个数。所得的商填入正下方的格子（左下角的格子），余数填入右边的格子（上面一行左数第二个格子）。用第一个余数再除以下一个除数 64。依照模板的顺序用同样的方法继续进行下去。

要记住，每次求得的商只能是 0 或者 1。如果被除数小于除数，商为 0，余数就是被除数。如果被除数大于或等于除数，那么商为 1，余数就是被除数减去除数所得之差。下面以 150 为例进行转换。

150	22	22	22	6	6	2	0
÷128	÷64	÷32	÷16	÷8	÷4	÷2	÷1
1	0	0	1	0	1	1	0

如果需要对两个二进制数进行加法或乘法，直接运算可能会比转换成十进制再进行

运算要简单些。这将是你真正喜欢二进制数的地方。想象一下，如果你记住了如下表格，掌握加法将是一件多么迅速的事情啊。

+	0	1
0	0	1
1	1	10

让我们用这个表格来计算两个二进制数的和：

$$\begin{array}{r} 1100101 \\ +\,0110110 \\ \hline 10011011 \end{array}$$

从最右一列开始：1 加 0 等于 1。右边第二列：0 加 1 等于 1。第三列：1 加 1 等于 0，进 1。第四列：1（进位）加 0 加 0 等于 1。第五列：0 加 1 等于 1。第六列：1 加 1 等于 0，进 1。第七列：1（进位）加 1 加 0 等于 10。

乘法表甚至比加法表还要简单，因为该表可以由两个基本的乘法规则推导出来：任何数乘以 0 结果都为 0，任何数乘以 1，结果都是这个数本身。

×	0	1
0	0	0
1	0	1

以下是 13_{TEN} 与 11_{TEN} 的二进制乘法运算过程：

$$\begin{array}{r} 1101 \\ \times\,1011 \\ \hline 1101 \\ 1101 \\ 0000 \\ 1101 \\ \hline 10001111 \end{array}$$

结果为 143_{TEN}。

人们在使用二进制数的时候通常将它们写成带有前导零的形式（即第一个 1 的左边有零）。例如 0011，而不是写做 11。这样写不会改变数字的大小，仅仅是为了美观。例如，以下是前 16 个二进制数以及与它们等价的十进制数。

二进制数	十进制数
0000	0
0001	1
0010	2
0011	3
0100	4
0101	5
0110	6
0111	7
1000	8
1001	9
1010	10
1011	11
1100	12
1101	13
1110	14
1111	15

让我们再看一下这一列二进制数。仔细考虑这 4 个垂直列中每一列的 1 和 0，注意它们在一列中自上而下是以怎样的规律交替的。

- 最右边的一列一直在 0 和 1 之间交替。
- 右数第二列是在每两个 0 和两个 1 之间相互交替。
- 下一列是在每四个 0 和每四个 1 之间相互交替。
- 再下一列是在每八个 0 和每八个 1 之间相互交替。

这是很有条理的，难道不是么？事实上，只要再重复这 16 个数字并且在每个数字的前面加一个 1 就可以很容易地写出后面的 16 个数字。

二进制数	十进制数
10000	16
10001	17
10010	18
10011	19
10100	20
10101	21

续表

10110	22
10111	23
11000	24
11001	25
11010	26
11011	27
11100	28
11101	29
11110	30
11111	31

从另外一种角度来看：当以二进制计数的时候，最右边的一位（最低位）以 0 和 1 交替。每当该位由 1 变为 0，从右边数的第二位（次低位）也随之改变——不是由 0 变到 1，就是由 1 变到 0。因此，每次只要有一个二进制数位的值由 1 变到 0，紧挨着的高位数字也会发生变化，而且其变化不是由 0 到 1 就是由 1 到 0。

在书写一个比较大的十进制数的时候，通常在每三位数字之间用一个逗号隔开，这样会让人很清楚地读出数字的大小。例如，当你看到数字 12000000 时，你可能不得不去数一下其中 0 的个数才知道这个数是多少，但是，如果数字写做 12,000,000，你一眼就会知道它是一千二百万。

二进制数的位长度增加得特别快。例如，一千二百万在二进制中应表示为：101101110001101100000000。为了让二进制数更易读，通常在每四个数字之间用一个连字符来分开，例如，1011-0111-0001-1011-0000-0000，或者每四位空出一个空格：1011 0111 0001 1011 0000 0000。本书的后面，我们将看到更简单的二进制数的表示方法。

通过将数字系统减少至只有 0 和 1 两个数字的二进制数字系统，我们已经讨论得足够深入。不可能再找到比二进制数字系统更简单的数字系统了。此外，二进制数字系统还在算术与电子技术之间架起了一座桥梁。在之前的章节中，我们所看到的开关、电线、灯泡，还有继电器等物体，都可以用来表示二进制数 0 和 1。

电线可以表示二进制数字。如果有电流流过这根电线就代表二进制数字 1，如果没有，则代表二进制数字 0。

开关可以表示二进制数字。如果开关接通（或闭合）就代表二进制数字 1，如果开关关断（或断开），则代表二进制数字 0。

灯泡可以表示二进制数字。如果灯泡点亮，就代表二进制数字 1，如果没点亮，则代表二进制数字 0。

电报继电器可以表示二进制数字。如果继电器闭合，就代表二进制数字 1，如果断开，则代表二进制数字 0。

二进制数与计算机之间有着紧密的联系。

大约在 1948 年，美国数学家约翰·威尔德·特克（John Wilder Turkey，生于 1915 年）就意识到随着计算机的普及，二进制数很可能会在未来发挥更重要的作用。他决定创造一个新的、更短的词语来代替使用起来很不方便的五音节词——binary digit。他曾经考虑使用 bigit 和 binit，但是最终他还是选用了这个短小、简单、精巧而且非常可爱的词——bit。

9

二进制数

托尼·奥兰多[1]在他 1973 年所写的一首歌中这样请求他挚爱的人："请在橡树上系上一条黄丝带"。他没有要求爱人进行详细的解释或者进行过多的讨论。他不想听到任何的"如果"、"而且"和"但是"。尽管这首歌是根据那些可能在真实生活中发生过的复杂感情和动人的往事所写的，但这个男人真正想要的答案仅仅是一个简单的"是"或"不是"。他希望在树上系一条黄丝带来表示"是的，尽管你做了很多错事，并且入狱三年，但我依然希望你可以回来和我共同生活。"他希望用树上没有系黄丝带来表示："你连停在这里都别想。"

这是两个界线分明、相互排斥的答案。托尼·奥兰多并没有唱"如果你想考虑一下，

1 托尼·奥兰多（Tony Orlando），美国歌手，1944 年出生于纽约，是一位希腊和波多黎各的混血儿，《老橡树上的黄丝带》是他的经典名曲，脍炙人口。歌曲描述一个在监狱服刑的丈夫在即将服刑期满的前夕，因担心远在家乡的妻子不接受他，写信告诉妻子，如果愿意再次接纳他，请在他出狱返家的当天，在家门口的老橡树上系上一条黄丝带；如果他没有看到黄丝带的话，他就会识相地默默离开。结果这位丈夫出狱返家时，家门口的老橡树上绑满了数以百计的黄丝带在空中飞扬，令他十分感动。后来，绑黄丝带成为全美国的一种风俗，表示迎接久别的亲人回家。——译者注

就系半条黄丝带吧”或者“如果你不再爱我，但我们依然是好朋友，就请系上蓝丝带吧。”相反，他将问题简化得非常非常简单。

还有几个和系黄丝带具有同样效果的例子（但可能无法用在诗里）：选择一种交通标志放在院子外面，来表示“请进”或“此路不通”。

或者在门上挂一个牌子：上面写着“关门”或“营业”。

或者用窗子里的一盏灯的亮灭来表示：“关闭”或“打开”。

如果这就是你想说的，你可以选择很多方法来表示“是”或者“不是”。你不需要用一句话、一个单词甚至一个字母来表达。你所需要的只是一个比特，也就是只需要一个 0 或 1 即可。

就如前面章节所提到的，十进制与其他数字系统相比并没有什么不同，只是我们通常使用它来计数。很明显，我们使用十进制数字系统是因为我们有十个手指。如果我们将数字系统建成八进制数字系统（如果我们是卡通人物）或四进制数字系统（如果我们是龙虾），甚至二进制数字系统（如果我们是海豚），也是合乎情理的。

但是二进制数字系统存在一点特殊性。这个特殊性就在于它是人们所能得到的最简单的数字系统。二进制数字系统中只有两个数字——0 和 1。如果想进一步简化这个数字系统，就只好把 1 去掉，最后我们就剩下的就只有一个数字 0 了。但仅用一个 0 是做不了任何事情的。

单词“bit（比特）”被创造出来代表英文“binary digit”，它的确是和计算机有关的词语中最可爱的一个。当然，这个词也有它一般的意义：“一小部分，程度很低或数量很少”。这个意义很贴切，因为 1 比特——一个二进制数字位——确实是一个非常小的量。

通常当一个新的词语出现的时候，它都会有自己新的意义。比特这个词也是这样的。1 比特的意思并不仅仅是海豚用来计数的二进制数字位所包含的意义。在计算机时代，比特被看做是组成信息块的基本单位。

这是一个大胆的声明。当然，二进制数不是传达信息的唯一方法。字母、单词、莫尔斯码、布莱叶盲文和十进制数都可以来表达信息。关键在于，比特所传递的信息量极少。1 比特是可能存在的最小的信息量。任何小于 1 比特的内容都根本算不上是信息。由