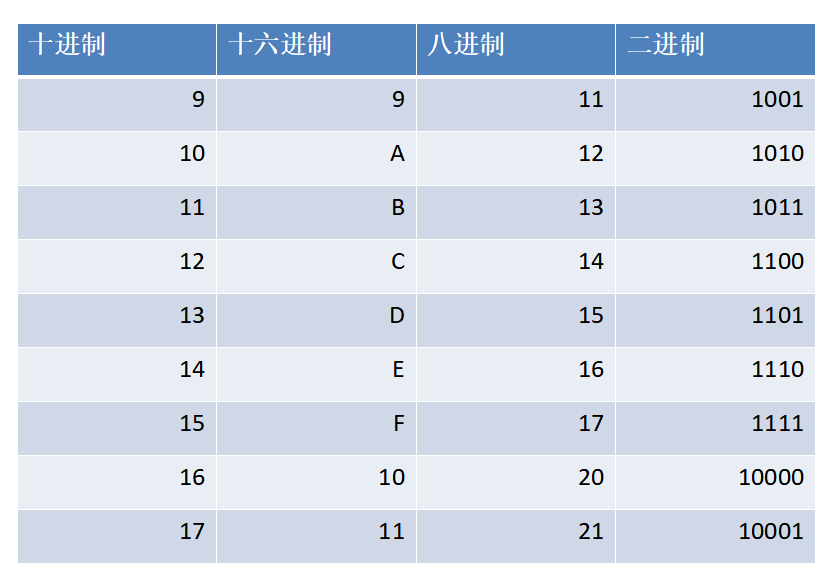
## 1.5 进制：

对于整数，有四种表示方式：

* 二进制 ：0 , 1 ，满2进1. 以0b或0B开头。
* 十进制 ：0 - 9 ，满10进1.
* 八进制 ：0 - 7 ，满8进1. 以数字0开头表示。023
* 十六进制 ：0-9及 A-F，满16进1. 以0x或0X开头表示。此处的A-F不区分大小写。0x23





### 1.5.1 二进制：

1）所有数字在计算机底层都以二进制的补码形式存在，0代表正，1代表负；

（以为二进制的原码和反码，与实际不符）。

2）原码：直接将一个十进制的数值换成二进制数。最高位是符号位：最高位是0代表正数，1代表负数；

|  |
| --- |
| *举例：3 ： 0000 0011*  *-3 ：1000 0011*  *十进制：3 --> 0011*  *二进制： 0 0 0 0 0 0 0 0*  *256 128 64 32 16 8 4 2 1*  *2的8次方* |

3）反码：

正数 ： 反码 = 原码；

负数 ： 反码 = 是对原码按位取反，只是最高位（符号位）确定为1。

|  |
| --- |
| 3 ： 0000 0011 源码、反码、补码  源码： -3 ：1000 0011  反码： -3 ：1111 1100 |

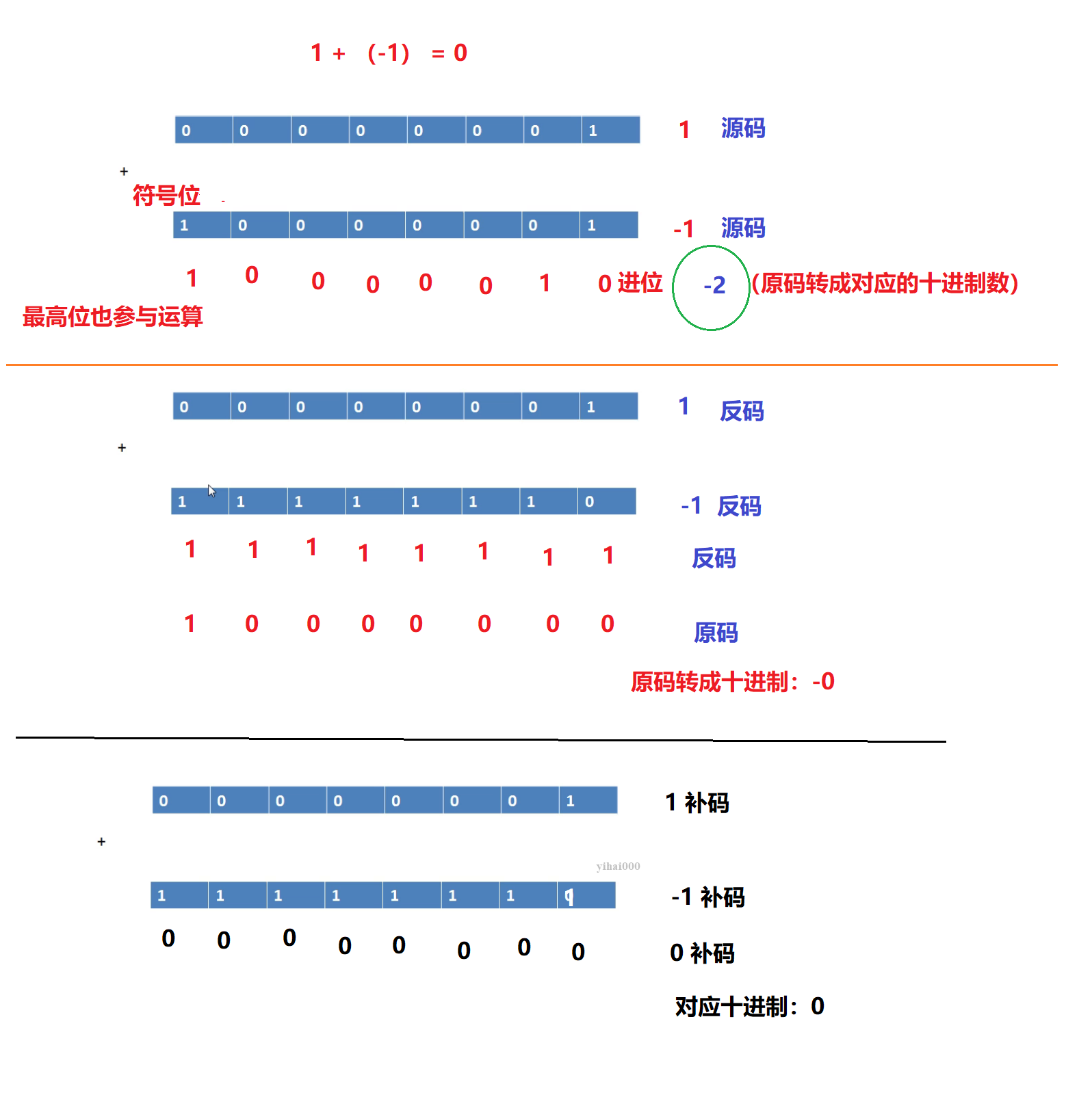
4）补码：

正数 ： 补码 = 原码；

负数 ： 反码 + 1；

|  |
| --- |
| 源码： -3 ：1000 0011  补码： -3 ：1111 1101 |

特点：正数的原码、反码、补码都相同；



Java整数常量默认是int类型，当用二进制定义整数时，其第32位是符号位；当是long类型时，二进制默认占64位，第64位是符号位；

### 1.5.2 进制间转化规律：

1. 二、八、十六转十：

对应相乘（从右往左），然后相加；

1. 十转其他:

对应相除，从下往上取余;

1. 二转八：卡3位；
2. 二转十六：卡4位；

