<https://blog.csdn.net/will_lee_buaa/article/details/48960755>

在java 中byte 占用一个字节，shot 占用两个字节，char 使用的是unicode编码，占用两个字节，int 占用四个字节,long 占用八个字节

在使用long 的时候，有个地方一定要注意,java 虚拟机默认把程序里出现的常量数字当做是int型。因此，如果想定义一个大于2^31（2147483648）的整数时,

如果直接使用long var = 数字常量（数字常量大于等于2^31）,结果达不到实际效果，实际上，此时的var 的值为溢出后的值。看几个例子。

1. long var = 1<<31;System.out.println(var); 打印结果为 -2147483648，1<<32 实际上溢出了,成为了负数，因为java虚拟机把1当成int型，然后对int型的1，左移31位，

导致结果溢出，至于为什么是-2147483648 很简单明了

2.long  var = 2147483648; System.out.println(var);  编译不通过，JIT 编译器直接识别出 2147483648 这个数字常量大于有符号整数所能表示的最大范围的。

3.long var = 0x80000000; System.out.println(var);  打印结果为  -2147483648 ，原因和1 一样的，java虚拟机把0x80000000,当成了int型的数，因此结果溢出

4.long var = ((long)1)<<31; System.out.println(var);  打印结果为 2147483648 ，符合预期结果。例子4和例子1的区别在于在移位操作前，先把int 型的1转换为了long类型。这样后续进行的移位操作就是long类型的移位操作了，结果不会溢出。

java 虚拟机的这个特点，使得我们不能用数字常量直接定义一个大于Integer.MAX\_VALUE变量，这点很容易出错。