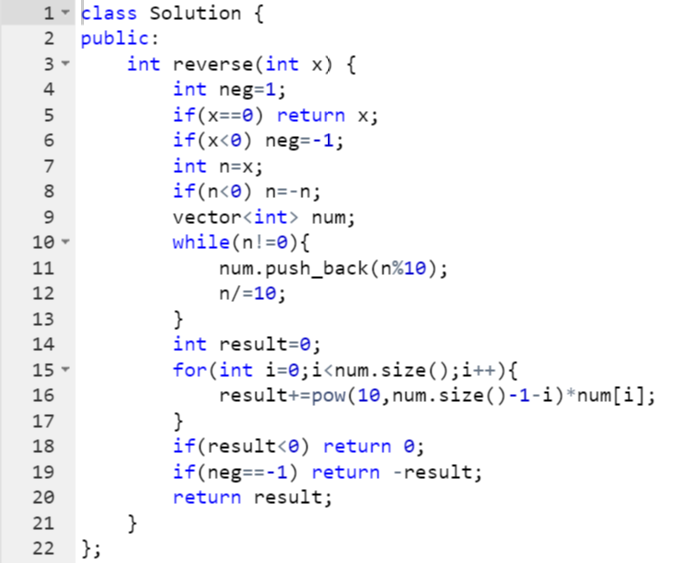
## 1 Reverse Integer

首先判断是不是负数，然后把当前的数字的字母取出，倒过来相加得到答案，再加上负号。

特殊的测试用例：0，反转以后溢出的情况（因为无法表达所以没有这个用例，还有就是INT\_MIN，因为此时取反直接溢出，但是因为这个数倒过来也没有意义，因为最后一位是8，因此也没有考虑，到时做题的时候应该注意这些特殊边界条件）



## 2 String to Integer

这个题目逻辑难度不大，看了提示以后按照可能的情况依次排除即可

特殊的测试用例：空，" a"," "," +123"," -123aa"

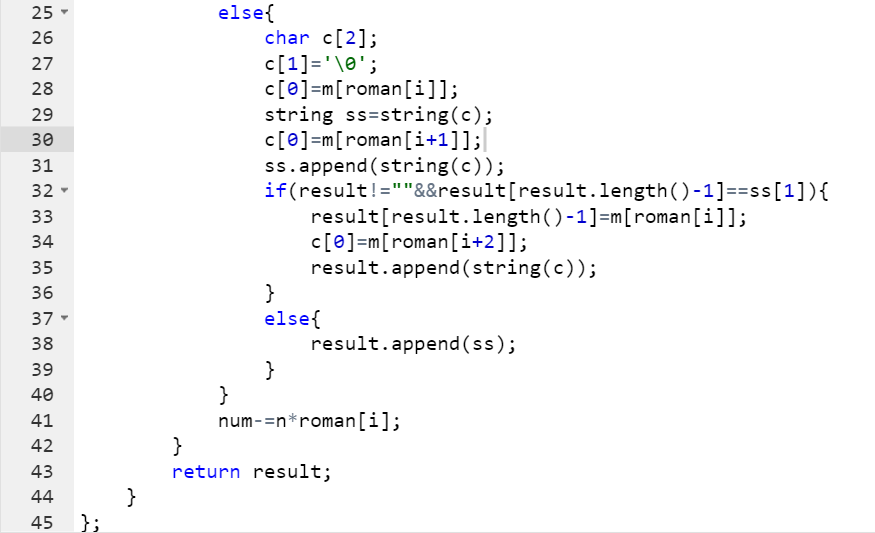


## 3 Integer to Roman

首先把所有单个罗马数字放进去，然后从大到小求，发现四个相同的就变成A[I]A[I+1]的情况，如果此时A[I-1]和A[I+1]相同，就变成A[I]A[I+2]的情况，如90的求法。还有就是对字符数组变成字符串，字符数组最后一定要用'\0'结尾

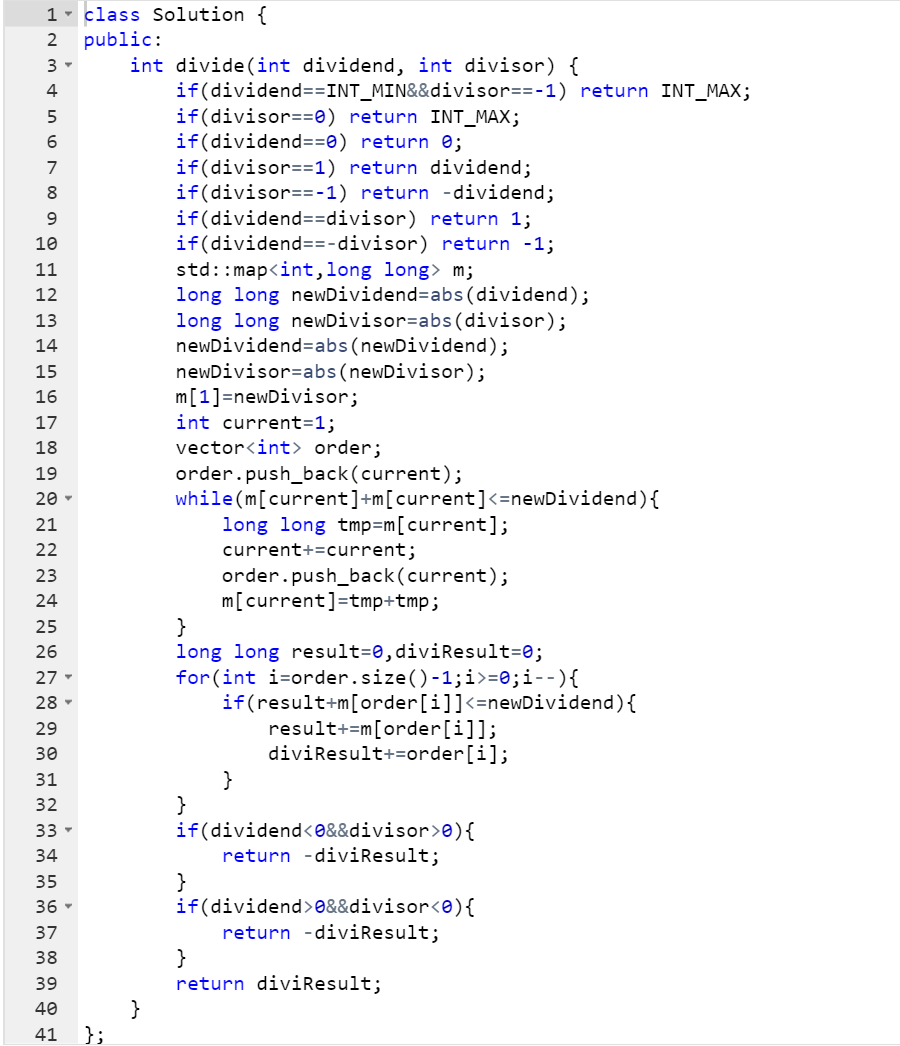
需要注意的测试用例99





## 4 Divide Two Integers

这个题目边界条件很多，首先要考虑算出来溢出的情况，也只有两种情况，一是除数为0，二是被除数是INT\_MIN，除数是-1（反转以后-INT\_MIN=INT\_MAX+1），因此首先排除他们。然后因为不能用乘除和取摩，因此用二分法，求出上限，然后从上限往下面一次累加到1倍的除数，最后看结果。其中一个关键的地方是必须注意后面计算的时候要用长整形，否则一旦被除数大于2的31次方，此时还不能结束循环，但是一旦求出2的32次方就会溢出，因此后面的计算应该用长整形。还有就是如果结果是负数，不是向下取整，而是向上取整，因此整个过程可以看成先求得绝对值的商，再根据情况加上负号



## 5 Multiply Strings

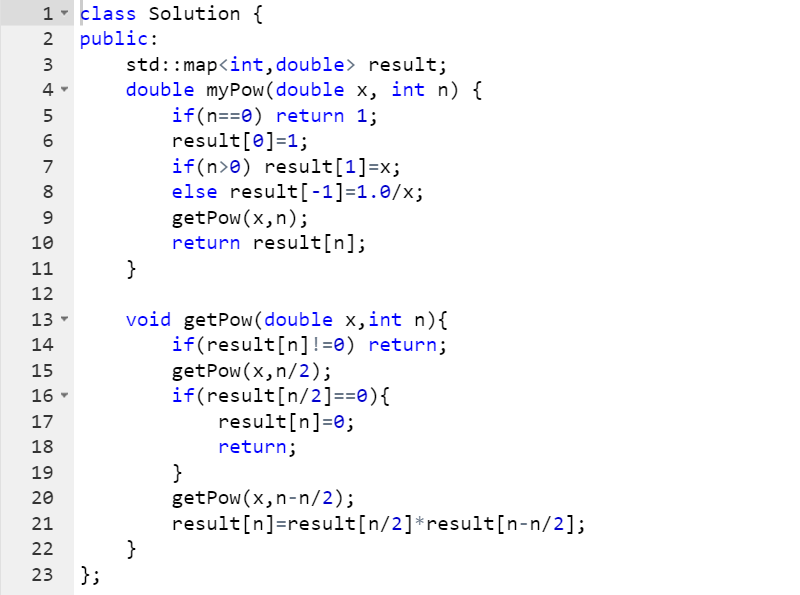
因为字符串任意大，因此不能变成整形以后求的在变回来。方法是按照小学学习乘法时候一样做，只不过累加的时候不是依次累加，而是将结果对应的位直接累加当前结果。要注意每个字符串是从左边开始数。

需要注意的测试用例：任一数为0



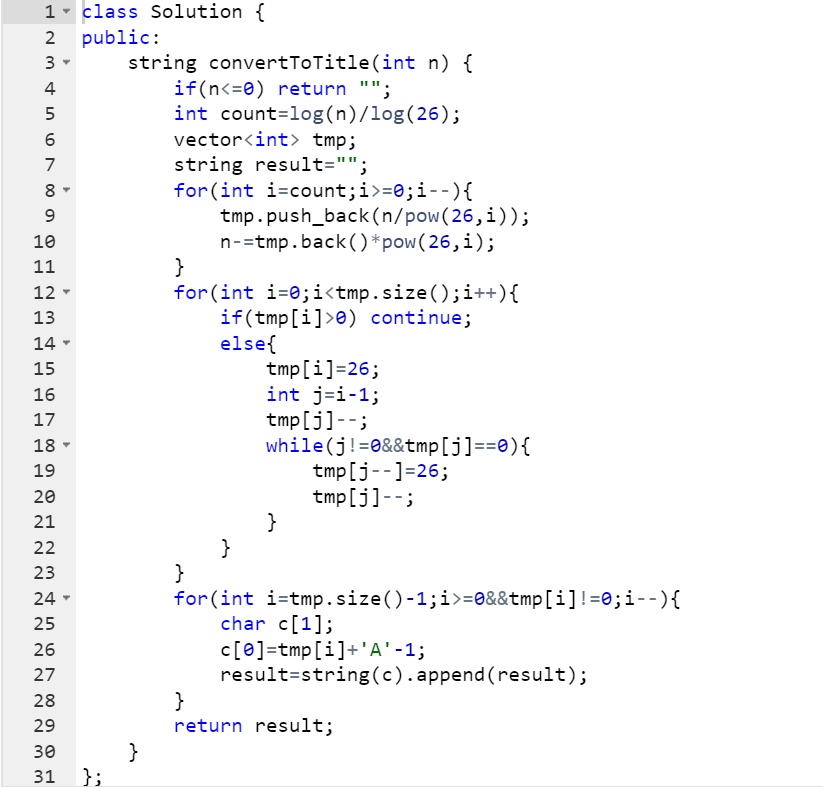
## 6 Pow(x, n)

一般的乘法太慢，显然会超时，因此用二分法，这里如果将对应结果放在数组里面，空间浪费太大，因此运用哈希映射。将计算过的存储起来，也有一点动态规划的感觉



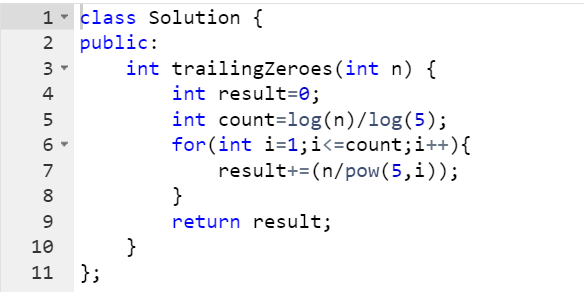
## 7 Excel Sheet Column Title

这个题目看着好像很容易，其实很容易做错，因为数是从1开始的，而且没有零这个数，26是Z，27不是A0而是AA。因此做起来有一些不好入手。大概做法是按照次方求得一个26进制的数组，里面可能有零，然后从最高位开始往下，如果碰到了0，就从上面以为借位，当前位变成Z，上一位减一，如果为零再往上面递推（最坏情况在最高位停止，如得到A000的时候）。



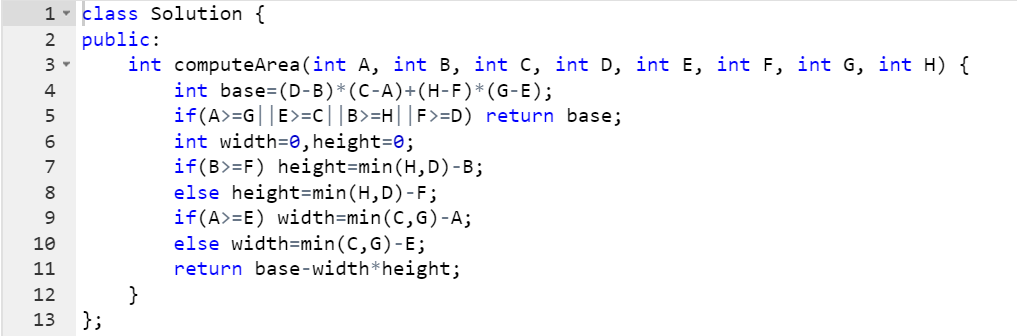
## 8 Factorial Trailing Zeroes

其实就是数5和2的因子数量，因为2的因子数量永远比5多，因此只用数5的。也就是从5的一次方开始，然后二次，然后三次，相加即可



## 9 Rectangle Area

首先求出两个矩形面积和，然后求公共部分。首先如果他们X方向不相交或者Y方向不相交，直接返回总面积，否则求出公共面积，再用总面积减去公共面积即可



## 10 Power of Two

首先要排除小于1的情况，因为小于1没有意义，然后依次对二取摩，如果为1，返回FALSE，否则对当前数除以2，如果得到的数为1则返回真。

需要注意的测试用例 0,1

