12 Integer to Roman

Given an integer, convert it to a roman numeral.

Input is guaranteed to be within the range from 1 to 3999.

解答思路：

思路1：

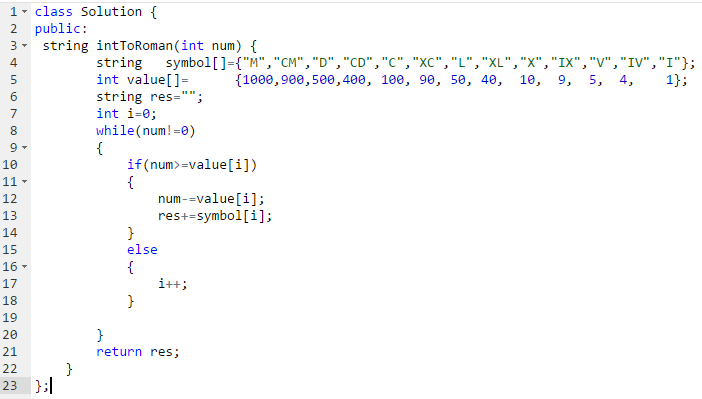
【罗马数字】

{"M","CM","D","CD","C","XC","L","XL","X","IX","V","IV","I"};

{1000,900,500,400, 100, 90, 50, 40, 10, 9, 5, 4, 1};//基数数组

对整数开始检查是否大于基数数组的元素值，大于等于时相减并链接相应字符；小于时基数组取下一个元素。

代码如下：



思路2：

【罗马数字】

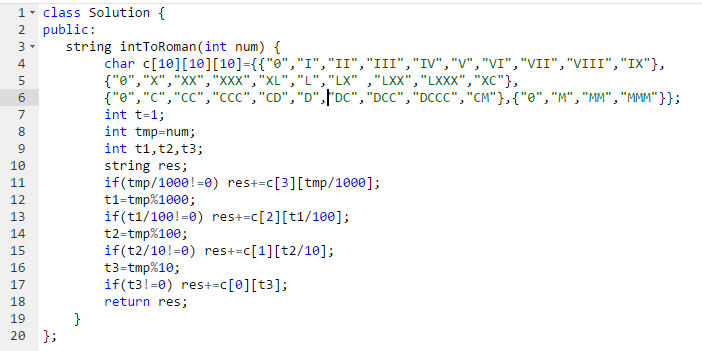
1~9: {"I", "II", "III", "IV", "V", "VI", "VII", "VIII", "IX"};

10~90: {"X", "XX", "XXX", "XL", "L", "LX", "LXX", "LXXX", "XC"};

100~900: {"C", "CC", "CCC", "CD", "D", "DC", "DCC", "DCCC", "CM"};

1000~3000: {"M", "MM", "MMM"}.

分别用1000，100,10，1除整数，得到商i(i>0) 就取相应第i个字符进行链接；然后相应取余数进行下一个数的计算。

代码如下：

41 First Missing Positive

Given an unsorted integer array, find the first missing positive integer.

For example,  
Given [1,2,0] return 3,  
and [3,4,-1,1] return 2.

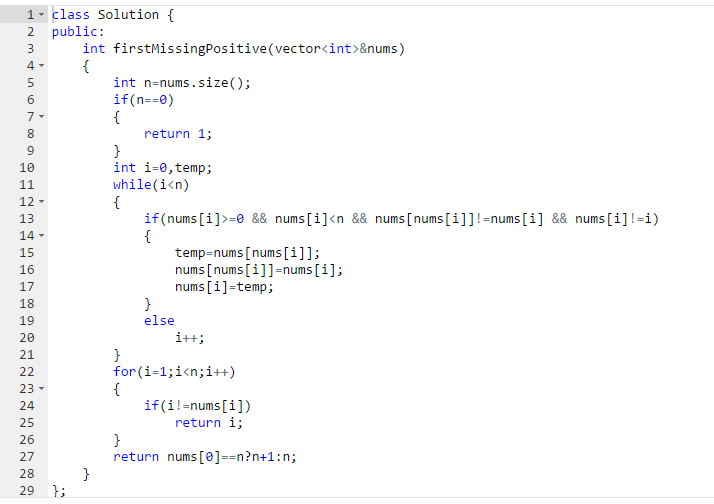
Your algorithm should run in *O*(*n*) time and uses constant space.

解题思路：

把对应的数放到对应的索引上，例如i放到A[i]上，这样只需要O(n)的遍历就能完成，然后用一个O(n)的遍历找第一个没有放到索引上的数返回。

最后就是可能A[0] == n，这时就要有个特殊情况的处理。

代码如下:



54 Spiral Matrix

Given a matrix of *m* x *n* elements (*m* rows, *n* columns), return all elements of the matrix in spiral order.

For example,  
Given the following matrix:

[

[ 1, 2, 3 ],

[ 4, 5, 6 ],

[ 7, 8, 9 ]

]

You should return [1,2,3,6,9,8,7,4,5].

解题思路：

1,设置每次顺时针旋转的起点坐标x,y，设置每次的行列边界row,col；

2,每次按照最上一行，最右一列，最下一行，最左一列进行扫描每个元素，注意边界判断条件；

3,每次循环x++，y++,row--,col--。

代码如下：

