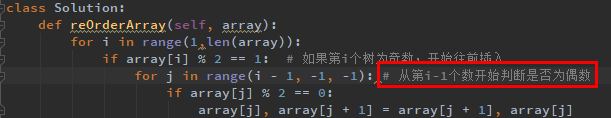
# 剑指offer

## 1、调整数组顺序使奇数位于偶数前面

输入一个整数数组，实现一个函数来调整该数组中数字的顺序，使得所有的奇数位于数组的前半部分，所有的偶数位于数组的后半部分，并保证奇数和奇数，偶数和偶数之间的相对位置不变。

思路：利用插入排序，从第二个位置开始，当该位置是奇数时，开始插入排序，遇到偶数就交换顺序，知道遍历到最后一个位置。





## 2、二进制中1的个数

输入一个整数，输出该数二进制表示中1的个数。其中负数用补码表示。

思路：直接对十进制数进行移位或与或非运算时，会自动转化为二进制，此处利用的是计算机里存的本来就是补码的概念，把给定数循环往右移动一位，分别与1相与，将所有为1的情况累加，就是该数1的个数。



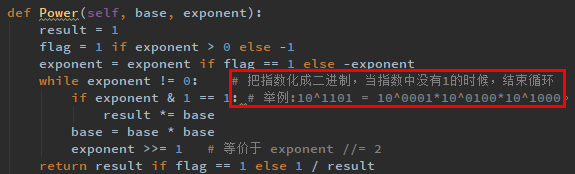
## 3、数值的整数次方

给定一个double类型的浮点数base和int类型的整数exponent。求base的exponent次方。

保证base和exponent不同时为0

思路：值得注意的两点：1、指数可能是负数，如果是负数的话，结果要取倒数。

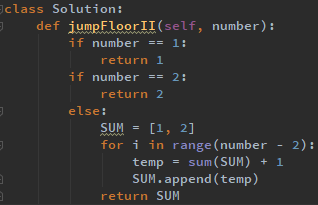
2、这里用来求幂时采用的是求快速幂的方法，将指数转化为二进制，并采用了移位的方法。



## 4、变态跳台阶

一只青蛙一次可以跳上1级台阶，也可以跳上2级……它也可以跳上n级。求该青蛙跳上一个n级的台阶总共有多少种跳法

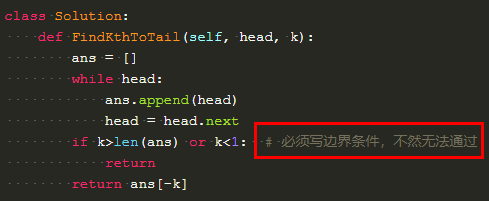
思路：与普通的跳台阶问题相比，普通的跳台阶问题是假设第一步走一步或两步，而这里确实第一步能走1....n步，所以第n项不是前两项和，而是前n-1项和 + 1。 后面的加1表示能够一步跳到n阶。



## 5、链表中倒数第k个结点

输入一个链表，输出该链表中倒数第k个结点

思路：转化为数组，值得注意的是，必须写边界条件，不然无法通过全部测试用例。



6、合并两个排序的链表

输入两个单调递增的链表，输出两个链表合成后的链表，当然我们需要合成后的链表满足单调不减规则。

思路：1、非递归: 就是归并排序的思路。

2、递归：直到其中有一个为空，那么就返回另一个列表。





## 6、树的子结构

输入两棵二叉树A，B，判断B是不是A的子结构。（ps：我们约定空树不是任意一个树的子结构）

思路：二叉树的子树和子结构  
 子树的意思是只要包含了一个结点，就得包含这个结点下的所有节点.  
 子结构的意思是包含了一个结点，可以只取左子树或者右子树，或者都不取。

子结构包含子树



## 7、矩阵中的路径：

矩阵题，首先应该定义一个全局的方向dx=[0,1,0,-1] dy=[1,0,-1,0]。如果指定了起始位置，那么就定义一个全局的visit数组，如果没有指定起始位置，那么就要定义一个局部的visit数组，注意，传入vistit之前，起始位置要置1.

## 数组中重复的数字

采用各就各位的方法，因为每个数组都是在0-n-1范围内，那么就可以将其放到原位，如果有位置已被占据，那么就说明重复了。

应用：递增式排序（相差1）