## 66．plus-one

***给定一个由整数组成的非空数组所表示的非负整数，在该数的基础上加一。***

***最高位数字存放在数组的首位， 数组中每个元素只存储单个数字。***

***你可以假设除了整数 0 之外，这个整数不会以零开头。***

总结：

个人认为这题其实跟小学学做加法的步骤一样，只是这里是规定了加1，

相加过十高位加一。大部分情况下个位加一即可，当个位+1为10时，区模后的0为个数位，高一位就得相应的加一，以此类推。

如果输入为[9,9,9]这种情况，则可以新增一个新的长度+1的数组，索引0赋值1即可。

## 283．moveZeroes

***给定一个数组 nums，编写一个函数将所有 0 移动到数组的末尾，同时保持非零元素的相对顺序。***

总结：

使用双指针，一个指向zero的索引，一个遍历时用，如果nums[i] != 0 ，则交换位置即可。

## Two\_num

***给定一个整数数组 nums 和一个目标值 target，请你在该数组中找出和为目标值的那 两个 整数，并返回他们的数组下标。***

***你可以假设每种输入只会对应一个答案。但是，你不能重复利用这个数组中同样的元素。***

总结：

暴力破解法，遍历每个元素，target = x+y；

使用哈希表，将每个元素都存放在map，再循环判断y = target-x是否在map中，如果是则返回x和y的索引。这里可以将存放map和遍历合在一起，只需一次遍历即可。如果存在答案且y = target-x 不在map中，那当遍历x=target – y 时也可以判断x在map中。

## 88.merge\_sorted\_array

***给你两个有序整数数组*nums1 *和*nums2*，请你将*nums2 *合并到*nums1 *中*，*使*num1 *成为一个有序数组。***

总结：

1. 先合并再排序；
2. 双指针，从两数组末尾开始比较并将较大的值存放到nums1的末尾处，注意特殊情况，如m为0时，以及nums1的最小值比nums2的最大值还要大时，则要将nums2的值都copy到nums1的前面部分。

## 21.merge\_two\_sorted\_lists

***将两个有序链表合并为一个新的有序链表并返回。新链表是通过拼接给定的两个链表的所有节点组成的。***

总结：