给定一个数组和一个值，在原地删除与值相同的数字，返回新数组的长度。

元素的顺序可以改变，并且对新的数组不会有影响。

您在真实的面试中是否遇到过这个题？

Yes

**样例**

给出一个数组**[0,4,4,0,0,2,4,4]**，和值 4

返回 4 并且4个元素的新数组为**[0,0,0,2]**

**思路：遍历数组，然后遇到相同的数值进行删除。注意点在于删除list之后list的数目会减小，所以直接按序号访问的必然会存在问题。**

class Solution:

"""

@param A: A list of integers

@param elem: An integer

@return: The new length after remove

"""

def removeElement(self, A, elem):

# write your code here

i=0

while i <=len(A)-1:

if A[i] == elem:

del A[i]

else:

i=i+1

给定一个链表，删除链表中倒数第n个节点，返回链表的头节点。

 注意事项

链表中的节点个数大于等于n

您在真实的面试中是否遇到过这个题？

Yes

**样例**

给出链表**1->2->3->4->5->null**和 n = 2.

删除倒数第二个节点之后，这个链表将变成**1->2->3->5->null.**

思路：删除倒数第几个就要知道整个链表的长度，那么长度的计算实在Null截至，这样在通过减法找到正向所对应的值，然后替换即可，最好可以保存地址。

"""

Definition of ListNode

class ListNode(object):

def \_\_init\_\_(self, val, next=None):

self.val = val

self.next = next

"""

class Solution:

"""

@param head: The first node of linked list.

@param n: An integer.

@return: The head of linked list.

"""

def removeNthFromEnd(self, head, n):

# write your code here

a=getnums(head,a={},i=1)

length=len (a)

h=length-n

if h ==1:

return head.next

a[h-1].next=a[h+1]

return head

def getnums(head,a,i):

while head !=None:

a[i]=head

head=head.next

i+=1

a[i]=None

return a

#### 3．反转整数

* [描述](http://www.lintcode.com/zh-cn/problem/reverse-integer/#description)
* [笔记](http://www.lintcode.com/zh-cn/problem/reverse-integer/#note)
* [数据](http://www.lintcode.com/zh-cn/problem/reverse-integer/#testcase)
* [评测](http://www.lintcode.com/zh-cn/problem/reverse-integer/#judge)

将一个整数中的数字进行颠倒，当颠倒后的整数溢出时，返回 0 (标记为 32 位整数)。

您在真实的面试中是否遇到过这个题？

Yes

**样例**

给定 x = 123，返回 321

给定 x = -123，返回 -321

思路:要知道溢出的条件，那么反转后不会有问题，下面就是反转了，怎么反转，转换成字符串进行反转好像是比较简单得，那么如果数是负数呢，就要判断下，把符号拿出来。

class Solution:

# @param {int} n the integer to be reversed

# @return {int} the reversed integer

def reverseInteger(self, n):

# Write your code here

a=[]

if n<0:

n=abs(n)

j=1

else:

j=2

while n//10!=0:

h=n

a.append(h)

n=n//10

a.append(n)

length=len(a)

sum=0

for i in range(length):

sum=sum+a[i]\*pow(10,(length-i-1))

#print(sum)

if sum>pow(2,32):

return 0

return sum\*pow(-1,j)

4. 翻转一个链表

您在真实的面试中是否遇到过这个题？

Yes

**样例**

给出一个链表**1->2->3->null**，这个翻转后的链表为**3->2->1->null**

**思路：反转链表就是再找个list把每个值存进去，然后到着取，一个个赋值就行**

**唯一问题就是返回列表头是要注意最好存起来，不然容易乱**

**"""**

**Definition of ListNode**

**class ListNode(object):**

**def \_\_init\_\_(self, val, next=None):**

**self.val = val**

**self.next = next**

**"""**

**class Solution:**

**"""**

**@param head: The first node of the linked list.**

**@return: You should return the head of the reversed linked list.**

**Reverse it in-place.**

**"""**

**def reverse(self, head):**

**# write your code here**

**a=[]**

**while head!=None:**

**a.append(head.val)**

**head=head.next**

**a=a[::-1]**

**if len(a)==0:**

**return None**

**headcopy=ListNode(a[0])**

**t=createnode(headcopy,a[1:])**

**return t**

**def createnode(head,a):**

**if len(a)>0:**

**head.next=ListNode(a[0])**

**createnode(head.next,a[1:])**

**return head**

5. 给定一个字符串，逐个翻转字符串中的每个单词。

您在真实的面试中是否遇到过这个题？

Yes

**说明**

* 单词的构成：无空格字母构成一个单词
* 输入字符串是否包括前导或者尾随空格？可以包括，但是反转后的字符不能包括
* 如何处理两个单词间的多个空格？在反转字符串中间空格减少到只含一个

思路：先把字符串给strip（），然后在split（）这样每个单词就被存在list里，然后list的单词还要再判断只允许出现字母，然后把list取出来一个个加就变成想要的str

class Solution:

# @param s : A string

# @return : A string

def reverseWords(self, s):

# write your code here

b=s.strip().split()

return" ".join(b[::-1])

6. 给定一个字符串和一个偏移量，根据偏移量旋转字符串(从左向右旋转)

您在真实的面试中是否遇到过这个题？

Yes

**样例**

对于字符串 "abcdefg".

offset=0 => "abcdefg"

offset=1 => "gabcdef"

offset=2 => "fgabcde"

offset=3 => "efgabcd"

思路：所谓的旋转很像交换。但数目不对等怎么交换，pop之后找个list append下然后在插入到前面就行。插入的时候最好倒着插。

class Solution:

# @param s: a list of char

# @param offset: an integer

# @return: nothing

def rotateString(self, s, offset):

# write you code here

length=len(s)

if length==0:

length=1

offset=offset%length

t=s[0:length-offset]

del(s[0:length-offset])

s+=t

7. 给定一个排序数组和一个目标值，如果在数组中找到目标值则返回索引。如果没有，返回到它将会被按顺序插入的位置。

你可以假设在数组中无重复元素。

您在真实的面试中是否遇到过这个题？

Yes

**样例**

**[1,3,5,6]**，5 → 2

**[1,3,5,6]**，2 → 1

**[1,3,5,6]**， 7 → 4

**[1,3,5,6]**，0 → 0

递归二分吧，把出口设置为长度等于2，就把所有情况包含进去了。下面的这种也可以，只不过很难想，出口为长度等于1，这样找不到的时候就是该插入的位置，一个加一个减。这样前面

class Solution:

"""

@param A : a list of integers

@param target : an integer to be inserted

@return : an integer

"""

def searchInsert(self, A, target):

# write your code here

length=len(A)

pre=0

end=length-1

while pre<=end:

mid=(pre+end)//2

if A[mid]<target:

pre=mid+1

elif A[mid] > target:

end=mid-1

else:

return mid

return pre

8.

写出一个高效的算法来搜索 *m* × *n*矩阵中的值。

这个矩阵具有以下特性：

* 每行中的整数从左到右是排序的。
* 每行的第一个数大于上一行的最后一个整数。

您在真实的面试中是否遇到过这个题？

Yes

**样例**

考虑下列矩阵：

[

[1, 3, 5, 7],

[10, 11, 16, 20],

[23, 30, 34, 50]

]

给出 target = 3，返回 true

搜索矩阵那么二分吧，先找中间值判断大小，递归吧。列矩阵二分，行矩阵也是二分。先是二分找到找到行。而后二分找列。

class Solution:

"""

@param matrix, a list of lists of integers

@param target, an integer

@return a boolean, indicate whether matrix contains target

"""

def searchMatrix(self, matrix, target):

#递归算法cost 866 ms

#非递归算法cost864ms

# write your code here

if len(matrix)==0:

return False

if target<matrix[0][0]:

return False

if target>matrix[-1][-1]:

return False

m=len(matrix);n=len(matrix[0])

flag,index=self.getm(matrix,0,m,target)

if flag:

start=0

end=n

while 1:

if start>=end:

print(1)

return False

mid=(start+end)/2

if target==matrix[index][mid]:

return True

if target>matrix[index][mid]:

start=mid+1

else:

end=mid

else:

return False

#递归表达会慢一些

'''

def getm(self,matrix,start,end,target):

if start>=end:

return False,-1

mid=(start+end)/2

if target<=matrix[mid][-1] and target>=matrix[mid][0]:

return True,mid

elif target<matrix[mid][0]:

return self.getm(matrix,start,mid,target)

elif target>matrix[mid][-1]:

return self.getm(matrix,mid+1,end,target)

'''

def getm(self,matrix,start,end,target):

while 1:

if start>=end:

return False,-1

mid=(start+end)/2

if target<=matrix[mid][-1] and target>=matrix[mid][0]:

return True,mid

elif target<matrix[mid][0]:

end=mid

elif target>matrix[mid][-1]:

start=mid+1

9. 给出2\*n + 1 个的数字，除其中一个数字之外其他每个数字均出现两次，找到这个数字。

您在真实的面试中是否遇到过这个题？

Yes

**样例**

给出**[1,2,2,1,3,4,3]**，返回 4

思路所有的数只有一个出现一次，要是相同的数能变成0那么一切都解决了。如果用减法好像不太可行，剩下相同变成0也就异或了但异或满足交换率吗？应该是满足的。

class Solution:

"""

@param A : an integer array

@return : a integer

"""

def singleNumber(self, A):

# write your code here

x=0

for i in range(len(A)):

x=x^A[i]

return x

10.

**单例** 是最为最常见的设计模式之一。对于任何时刻，如果某个类只存在且最多存在一个具体的实例，那么我们称这种设计模式为单例。例如，对于 class Mouse (不是动物的mouse哦)，我们应将其设计为 singleton 模式。

你的任务是设计一个 getInstance 方法，对于给定的类，每次调用 getInstance 时，都可得到同一个实例。

您在真实的面试中是否遇到过这个题？

Yes

**样例**

在 Java 中:

A a = A.getInstance();

A b = A.getInstance();

a 应等于 b.

设计模式的问题。