9. Palindrome Number

Some hints:

Could negative integers be palindromes? (ie, -1)

no

If you are thinking of converting the integer to string, note the restriction of using extra space.

You could also try reversing an integer. However, if you have solved the problem "Reverse Integer", you know that the reversed integer might overflow. How would you handle such case?

There is a more generic way of solving this problem.

现在让我们考虑如何恢复上一半的数字。对于号码 1221，如果我们这样做 1221 % 10，我们得到最后一个数字 1，得到第二个数字，我们需要删除最后一个数字 1221，我们可以通过将它除以 10 来实现 1221 / 10 = 122。然后我们可以通过将模数乘以 10 得到最后一个数字 122 % 10 = 2，如果我们将最后一位乘以 10，并添加第二个最后一个数字 1 * 10 + 2 = 12，则它给出我们想要的还原数。继续这个过程将给我们更多的数字的恢复的数字。

```
class Solution(object):
    def isPalindrome(self, x):
        if x<0:
            return False
        elif x==0:
            return True
        temp=x
        y=0
        while x:

            y = y*10 + x%10;
            x=x/10
        if temp==y:
            return True
        else:
            return False
```

附上刚开始学习 Python 时练习的用 Queue 的回文检测办法

```
#   palindromic letter checking 回文检测
import collections
def palCheck(string):
    equal=True
    palDeque = collections.deque()
    for char in string:
        palDeque.appendleft(char)
    while palDeque.__len__()>1 and equal   :
        front= palDeque.pop()
        rear = palDeque.popleft()
        if front!= rear:
            return False
    return equal
```