LC 7: Reverse Integer!

根本不好做！尤其 Python 这种动态类型，坑多

强烈建议做！

快速、典型的 python 解法：

```
def reverse(self, x):
    s = cmp(x,0)
    r = int(`s*x`[::-1])
    return s*r*(r<2**31)
```

这个操作我真心惊呆了！

"s*x" 是用 ` 号括起来的，居然能把这样一个 variable 的 value 转成 string！

S 本身是个 bool 值，在这里却用作 int！

把 x 与它的 sign 相乘，就一定会得到一个正数！

这样就不用分正负情况，从而无论是从第0个开始反转还是从第1个开始reverse了。

个人认为，这种解法只是好看，不general，对于解题能力的提高，意义很小!!

Java solution:

if overflow exists, the new result will not equal previous one.

No flags needed.

No hard code like 0xf??? ???? needed.

```
public int reverse(int x) {
    int result = 0;
    while (x != 0) {
        int tail = x % 10;
        int newResult = result * 10 + tail;
        if ((newResult - tail) / 10 != result)
            { return 0; }
        result = newResult;
        x = x / 10;
    }
    return result;
}
```

overflow是发生在求法这种运算上.

理解这一行的关键在于，不是要判断input是否overflow，而是要判断reverse以后的是否overflow

根据这一思路的一个变体：

```
int reverse(int x) {
    int sign = x < 0 ? -1 : 1;
    x = abs(x);
    int res = 0;
    while (x > 0) {
        if (INT_MAX / 10 < res || (INT_MAX - x % 10)
< res * 10) {
            return 0;
        }
        res = res * 10 + x % 10;
        x /= 10;
    }
    return sign * res;
}
```

res * 10 + x % x > INT_MAX