# Reverse integer

**7. Reverse Integer**

Medium 11061 12405 Add to List Share

Given a signed 32-bit integer `x`, return `x` *with its digits reversed*. If reversing `x` causes the value to go outside the signed 32-bit integer range `[-2^31, 2^31 - 1]`, then return `0`.

**Assume the environment does not allow you to store 64-bit integers (signed or unsigned).**

**Example 1:**

```
Input: x = 123
Output: 321
```

**Example 2:**

```
Input: x = -123
Output: -321
```

**Example 3:**

```
Input: x = 120
Output: 21
```

$x = 123 \quad r = 0$

Remainder of x

$r = x \% 10$

$r = 123 \% 10 \Rightarrow 3, 2$

then add r to the sum

$sum = (sum * 10) + r$

↓

increase to next digit

i.e $sum = (0 * 10) + 3$

$sum = 321$

reduce x i.e $x = x/10 = \frac{123}{10} = 12, 1$

return sum if (ans

```cpp
class Solution {
public:
    int reverse(int x) {
        int ans=0;
        while(x!=0){
            int lastdigit=x%10;
            if(ans > (INT_MAX/10) || ans< (INT_MIN/10))
            {
                return 0;
            }
            ans=(ans*10)+lastdigit;
            x/=10;
        }
        return ans;
    }
};
```