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
```

Constraints:

- -2$_{31}$ <= x <= 2$_{31}$ - 1

Approach :

32 bit integer maximum is 2^31.

So our number must be within 2^31 and -2^31.

Step - 1 : initialize rev = 0

Step - 2 : reverse integer until x is not equal 0

Step - 3 : for overflow cases if rev > = 2^31 and rev < -2^31 then it returns 0.