

Reverse the digits of a 32-bit signed integer

Key concepts:

- Integer limits: minimum: $\text{INT_MIN} = -2^{31}$
maximum: $\text{INT_MAX} = 2^{31}$
you must check overflow before it happens
- No 64-bit integers allowed; you need to work digit by digit.

Algorithm strategy:

- Extract digits: - repeatedly take $x \% 10$ to get the last digit
use integer division $x // 10$ to shrink x .
- Build reversed number: - suppose your current result is rev .
After extracting digit d , you want:
 $\text{rev} = \text{rev} * 10 + d$;
- Check for overflow before assigning:
 - before doing $\text{rev} = \text{rev} * 10 + d$ you must ensure it won't overflow: If $\text{rev} > \text{INT_MAX} // 10$ & $\text{rev} \geq \text{INT_MAX} // 10$ and $d > 8$, it will overflow
 - Likewise for INT_MIN with negative numbers.

Edge Cases:

- reversing $0 \rightarrow 0$
- reversing $1000 \rightarrow 1$ (leading zeros are dropped)
- reversing $-120 \rightarrow -21$
- reversing values near INT_MAX or $\text{INT_MIN} \rightarrow$ check for overflow!