Reverse the digits of a 32-bit signed integer

Key concepts:

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

Algorithm strategy:

1. Extract digits: - repeatedly take $x \,\%\, 10$ to get the last digit
   - use integer division $x / 10$ to shrink x.
2. Build reversed number: - suppose your current result is rev.
   After extracting digit d, you want:
   `rev = rev * 10 + d;`
3. Check for overflow before assigning:
   - before doing `rev = rev * 10 + d` you must ensure it won't overflow: If `rev > INT_MAX / 10` or `rev == INT_MAX / 10` and `d > 7`, it will overflow
   - Likewise for INT_MIN with negative numbers.

Edge Cases:

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