Problem Statement

Title: *The Secret Time Machine - Hard Mode*

In a distant land, a time machine allows you to manipulate a **string of digits** (0-9).
You are given a **number as a string** s and an **integer** k.

You are allowed to perform the following operation **exactly k times** where:

- **Pick any two indices i and j** (i < j).
- **Swap the digits** at positions i and j **if and only if** the **absolute difference** between the digits is **at least 3**.
  That is, you can swap s[i] and s[j] if |int(s[i]) - int(s[j])| >= 3.

The goal is:

- **After exactly k swaps**, make the **lexicographically largest** possible string.

If not possible (can't make k moves satisfying the rule), output "IMPOSSIBLE".

Input

- A string s of digits ($1 \leq len(s) \leq 10^4$)
- An integer k ($1 \leq k \leq 10^4$)

Output

- A string: the lexicographically largest string achievable after exactly k operations, or "IMPOSSIBLE".

Examples

Example 1

Input:

s = "2731"

k = 2

Output:

7321

Explanation:

- Swap (0,1): 2 and 7 → |2-7| = 5 (ok) → 7231
- Swap (1,2): 2 and 3 → |2-3| = 1 (not allowed), but 7 and 3 → |7-3|=4 (allowed) → swap 7 and 3 → 7321

Example 2

Input:

s = "555"

k = 1

Output:

IMPOSSIBLE

Explanation:

- All digits are same; |5-5| = 0 < 3 → no swaps allowed.

Important points:

- This time, **|digit difference| ≥ 3** constraint replaces the simple 'larger digit' condition.
- Strategy must ensure each swap is useful to **maximize** the lexicographic order.