

Given a positive integer N and a non-negative integer K, you need to find the maximum possible number that can be formed by performing at most K swaps of digits in N.

In one swap operation, you can choose any two positions in the number and swap the digits at those positions. Your goal is to maximize the resulting number using at most K such swap operations.

**Input Format**

- First line contains two space-separated integers N and K
- N is the given positive integer ( $1 \leq N \leq 10^{18}$ )
- K is the maximum number of swaps allowed ( $0 \leq K \leq 10$ )

**Constraints**

- $1 \leq N \leq 10^{18}$
- $0 \leq K \leq 10$
- The number of digits in N is at most 18
- The result will not have leading zeros

**Output Format**

Print the maximum possible number after performing at most K swaps.

**Sample Input 0**

12345 0

**Sample Output 0**

12345

**Sample Input 1**

54321 0

**Sample Output 1**

54321