

Problem F. Encoded Strings I

Input file: **standard input**
Output file: **standard output**
Time limit: 1 second
Memory limit: 512 megabytes

Tommy has just invented an interesting string encoding algorithm, which is described below.

- For every string S , we may define a character-mapping function F_S , which maps every character occurring in S to a lowercase letter, as below

$$F_S(c) = \text{chr}(G(c, S))$$

where $\text{chr}(i)$ is the $(i+1)$ -th lowercase Latin letter, and $G(c, S)$ is the number of different characters after the last occurrence of c in S .

- To encode a string S by Tommy's algorithm, replace every character c in S by $F_S(c)$ simultaneously.

For example, the encoded string of **abc** is **cba**, and the encoded string of **cac** is **aba**.

You are given a string of length n and then encode all the n **nonempty prefixes**. Your task is to find the encoded string that has the greatest lexicographical order among all the encoded strings.

Input

The first line contains an integer n ($1 \leq n \leq 1\,000$).

The second line contains a string of length n , which consists of only the first 20 lowercase letters, **a** to **t**.

Output

Output the encoded string that has the greatest lexicographical order among all the encoded strings.

Examples

standard input	standard output
4 aacc	bbaa
3 aca	ba

Note

In the first sample case, the four nonempty prefixes **a**, **aa**, **aac** and **aacc** will be encoded to **a**, **aa**, **bba** and **bbaa** respectively, where **bbaa** has the greatest lexicographical order.

In the second sample case, the three nonempty prefixes **a**, **ac** and **aca** will be encoded to **a**, **ba** and **aba** respectively, where **ba** has the greatest lexicographical order.