Problem: Beautiful Strings

Program file: beautiful.{c, cpp, java}
Input file: beautiful.in

This problem involves manipulating strings. First, some terms:

- Given strings S1 and S2, we say that S2 can be *derived from* S1 if and only if we can start with S1 and remove selected characters until S2 results. We cannot rearrange or alter characters in S1, only remove them.
- A string is *unique* if and only if each character in it appears only once in that string.
- Given strings S1 and S2, we say that S1 is more *beautiful* than S2 if and only if S1 is longer, or (if they're the same length) S1 is lexicographically later than S2. ("Lexicographically later" means appearing later in a case-sensitive dictionary-order sort.)

Your task is to write a program to answer the following question: Given a string S, what is the most beautiful unique string that can be derived from it?

Your input file begins with an integer N, specifying the number of strings in the input file. This is followed by N strings. Each string is at least 3 characters long, no more than 100,000 characters long, and made up entirely of printable non-whitespace ASCII characters (no embedded spaces, tabs, or newlines, but digits or punctuation may be present). Do not assume each string is on a line by itself.

Output for each case is the most beautiful unique string that can be derived for that case, on a line by itself.

Sample Input	Sample Output
4 abCDef ffaabbccddeeffgg9032: ffaabbccd:"';deeffgg9032: IHaveADogHisNameIsFido	abCDef fabcdeg9032: fabcd:"';eg9032 IHaveADogisNmFd