**BIGGER IS GREATER**

This Node.js program solves the **“Next Lexicographical Permutation”** problem. Given a string w, the goal is to find the next string that is **lexicographically larger** using the same characters. If no such string exists, the program outputs "no answer".

The core logic is implemented in the biggerIsGreater function. First, the string is converted into an array of characters to allow easy manipulation. The program then scans the string from right to left to identify the **pivot**—the rightmost character that is smaller than the character immediately following it. This pivot marks the position where a change can be made to create a larger permutation.

If no pivot is found, the string is already in descending order, which means it is the largest possible permutation, and the function returns "no answer". Otherwise, the program searches for the smallest character to the right of the pivot that is larger than the pivot itself and swaps them. This ensures that the resulting string is just slightly larger than the original. Finally, the characters following the pivot are reversed to form the **smallest possible suffix**, guaranteeing that the result is the immediate next permutation.

The program efficiently handles multiple test cases and outputs the results for each. Its algorithm runs in **linear time relative to the string length**, making it suitable for even long strings. Overall, this code provides a clear and efficient solution to the problem of finding the next lexicographical string.