public class Main {

// Problem 3: Find the longest substring in alphabetical order

public static String longestAlphabeticalSubstring(String s) {

// Initialize variables to track the longest substring and the current substring

String longestSubstring = "";

String currentSubstring = "";

// Loop through the string to find the longest alphabetical substring

for (int i = 0; i < s.length(); i++) {

// If the current substring is empty or the next character is >= the current one

if (currentSubstring.isEmpty() || s.charAt(i) >= currentSubstring.charAt(currentSubstring.length() - 1)) {

currentSubstring += s.charAt(i); // Add the current character to the current substring

} else {

// Update the longest substring if the current one is longer

if (currentSubstring.length() > longestSubstring.length()) {

longestSubstring = currentSubstring;

}

currentSubstring = String.valueOf(s.charAt(i)); // Start a new substring

}

}

// After the loop, check the final currentSubstring

if (currentSubstring.length() > longestSubstring.length()) {

longestSubstring = currentSubstring;

}

return longestSubstring; // Return the longest alphabetical substring

}

public static void main(String[] args) {

String s = "azcbobobegghakl";

System.out.println("Longest substring in alphabetical order is: " + longestAlphabeticalSubstring(s)); // Expected output: "beggh"

}

}