**🧠 Challenge Name:** *Frequency Frenzy*

**💻 Problem Statement:**

You are given a string containing only lowercase alphabets. Your task is to find the **character(s)** that appear **most frequently** in the string. If multiple characters have the same highest frequency, print them in **lexicographical order**.

**🔸 Input:**

A single line containing a string s (1 ≤ len(s) ≤ 10^5)

**🔸 Output:**

Print the most frequent character(s), **separated by a space**, in **lexicographical order**.

**🔹 Example:**

**Input:**

aabbbbccddeeffgggg

**Output:**

b g

**🔹 Explanation:**

* Frequencies: a:2, b:4, c:2, d:2, e:2, f:2, g:4
* Characters with highest frequency (4): b, g
* In lex order: **b g**

**🧠 Challenge Name:** *String Compression*

**💻 Problem Statement:**

Given a string s, compress it using the following algorithm:

* Begin with a string s and replace consecutive identical characters by a single character followed by the number of occurrences of that character.
* If the compressed string is not shorter than the original string, return the original string.

For example:

* "aabcccccaaa" becomes "a2b1c5a3".
* "abcd" remains "abcd" (since it's not shorter than the original).

**🔸 Input:**

* A string s (1 ≤ len(s) ≤ 1000)

**🔸 Output:**

* The compressed string if it's shorter than the original, otherwise the original string.

**🔹 Example:**

**Input:**

aabcccccaaa

**Output:**

a2b1c5a3

**Input:**

abcd

**Output:**

abcd