A string is called **K**-symmetrical if it can be represented as **K** concatenated copies of another string. For example, the string "abababab" is simultaneously **1**-symmetrical (1 × "abababab"), **2**-symmetrical (2 × "abab") and **4**-symmetrical (4 × "ab"), but not **3**-symmetrical or **6**-symmetrical. Obviously every string is **1**-symmetrical.

You’re given a string **S**, consisting of lowercase latin letters and a natural number **K**. Your task is to rearrange the letters in the string **S** in such a way that the resulting string becomes **K**-symmetric or determine that it is impossible to do so.

**Input**

On the first line of the file kstring.in the string **S** and the number **K** are given.

**Output**

On one line in the file kstring.out, print the rearranged letters of **S** so that they form a **K**-symmetrical string, or **"-1"** if this is not possible. If there is more than one solution, print any of them.

**Constraints**

$$1\leq |S|\leq 10^{5}$$

$$1\leq K\leq |S|$$

**Time limit: 0.2 sec.**

**Memory limit: 256 MB.**

**Sample tests**

|  |  |
| --- | --- |
| **Input (kstring.in)** | **Output (kstring.out)** |
| abacbc 2  | abcabc |

|  |  |
| --- | --- |
| **Input (kstring.in)** | **Output (kstring.out)** |
| abbaba 3 | bababa |

|  |  |
| --- | --- |
| **Input (kstring.in)** | **Output (kstring.out)** |
| abccaba 2 | -1 |