Java Substring Comparisons

We define the following terms:

- Lexicographical Order, also known as *alphabetic* or *dictionary* order, orders characters as follows:

  For example, `ball` < `cat`, `dog` < `dorm`, `Happy` < `happy`, `Zoo` < `ball`.

- A substring of a string is a contiguous block of characters in the string. For example, the substrings of `abc` are `a`, `b`, `c`, `ab`, `bc`, and `abc`.

Given a string, , and an integer, , complete the program so that it finds the lexicographically *smallest* and *largest* substrings of length .

Function Description

Complete the *getSmallestAndLargest* program in the editor below.

getSmallestAndLargest has the following parameters:

- *string s:* a string
- *int k:* the length of the substrings to find

Returns

- *string:* the string ' + "\n" + ' where and are the two substrings

Input Format

The first line contains a string denoting .
The second line contains an integer denoting .

Constraints

- 1 <= |s| <= 1000
- `s` consists of English alphabetic letters only (i.e., `[a-zA-Z]`).

Sample Input 0

```
welcometojava
3
```

Sample Output 0

```
ava
wel
```

Explanation 0

String has the following lexicographically-ordered substrings of length :

```
 "ava", "com", "alc", "etc", "jav", "lco", "met", "oja", "ome", "toj", "ove"
```

We then return the first (lexicographically smallest) substring and the last (lexicographically largest) substring as two newline-separated values (i.e., `ava\nwel`).

The stub code in the editor then prints `ava` as our first line of output and `wel` as our second line of output.