
Problem Statement

Michelle has created a word game for her students. The word game begins with Michelle writing a string and a number, K , on the board. The students must find a substring of size K such that there is exactly one character that is repeated once; in other words, there should be $K - 1$ distinct characters in the substring.

Write an algorithm to help the students find the correct answer. If no such substring can be found, return an empty list; if multiple such substrings exist, return all of them, without repetitions. The order in which the substrings are returned does not matter.

Input

The input to the function/method consists of two arguments -
inputString, representing the string written by the teacher;
num, an integer representing the number, K , written by the teacher on the board.

Output

Return a list of all substrings of *inputString* with K characters, that have $K-1$ distinct character i.e. exactly one character is repeated, or an empty list if no such substring exists in *inputString*. The order in which the substrings are returned does not matter.

Constraints

The input integer can only be greater than or equal to 0 and less than or equal to 26 ($0 \leq num \leq 26$)
The input string consists of only lowercase alphabetic characters.

Examples

Input:

inputString = awaglk
num = 4

Output:

[awag]

Explanation:

The Substrings are {awag, wagl, aglk}

The answer is awag as it has 3 distinct characters in a string of size 4, and only one character is repeated twice.

Problem Statement

You are working on developing a movie with Amazon Video and want to devise an application to easily break up individual shots (short sequence from a particular camera angle) in a video into scenes (a sequence of shots). Each shot is labeled with a letter. There is already an algorithm that breaks the video up into shots and labels them.

Write a function which will partition a sequence of shots into minimal subsequences so that no shot appears in more than one subsequence. The output should be the length of each subsequence.

Input

The input to the function/method consists of an argument - *inputList*, a list of characters representing the sequence of shots.

Output

Return a list of integers representing the length of each scene, in the order in which it appears in the given sequence of shots.

Examples

Example 1:

Input

inputList = [a, b, c]

Output

[1, 1, 1]

Explanation:

Because there are no recurring shots, all shots can be in the minimal length 1 subsequence.

Example 2:

Input

inputList = [a, b, c, a]

Output

[4]

Explanation:

Because 'a' appears more than once, everything between the first and last appearance of 'a' must be in the same list.

Example 3:

Input:

inputList = [a, b, a, b, c, b, a, c, a, d, e, f, e, g, d, e, h, i, j, h, k, l, i, j]

Output:

[9, 7, 8]