Stack Puzzle

There are two sequences of stack operations converting the word TROT to TORT:

```
[
i i i i o o o o
i o i i o o i o
]
```

where `i` and `o` stands for push and pop operation respectively. In this problem, you are given two words and you are asked to find out all sequences of stack operations converting the first word to the second.

Input
The input consists of two lines, the first of which is the source word and the second is the target word.

Output
Your program should print a sorted list of valid `i/o` sequences. The list is delimited by

```
[
]
```

and the sequences are sorted in lexicographical order. Within each sequence, `i`'s' and `o`'s are separated by a single space and each sequence is terminated by a new line.

Process
Given an input word, a valid `i/o` sequence implies that every character of the word is pushed and popped exactly once, and no attempt is ever made to pop an empty stack. For example, if the word FOO is input, then the sequence:

- `i i o i o o` is valid and produces OFF
- `i i o` is not valid (too short),
- `i i o o o i` is not valid (illegal pop from an empty stack)

A valid sequence infers a permutation of the letters in the input word. For example, given the input word FOO, both sequences `i i o i o o` and `i i i o o o` give the word OOF.

Example 1

```
Input:
madam
adamm

Output:
[
i i i i o o o i o o
i i i i o o o o i o
i i o i o i o i o o
i i o i o i o o i o
]
```

Example 2

```
Input:
bahama
bahama
```

```
Output:
[
i o i i i o o i i o o o
i o i i i o o o i o i o
i o i o i o i i i o o o
i o i o i o i o i o i o
]
```

Example 3

```
Input:
long
short

Output:
[
]
```

Example 4

```
Input:
eric
rice

Output:
[
i i o i o i o o
]
```