You are given a string and two markers (the initial and final). You have to find a substring enclosed between these two markers. But there are a few important conditions:

- The initial and final markers are always different.
- If there is no initial marker, then the first character should be considered the beginning of a string.
- If there is no final marker, then the last character should be considered the ending of a string.
- If the initial and final markers are missing then simply return the whole string.
- If the final marker comes before the initial marker, then return an empty string.

Input: Three arguments. All of them are strings. The second and third arguments are the initial and final markers.

Output: A string.

Example:

```
between_markers('What is >apple<', '>', '<') == 'apple'
between_markers('No[/b] hi', '[b]', '[/b]') == 'No'
```

How it is used: *for parsing texts*

Precondition: *can't be more than one final marker and can't be more than one initial. Marker can't be an empty string*