Ruby - Strings - Methods II

In this tutorial, we'll learn about the methods in String class that help us to search and replace portions of the string based on a text or pattern.

- `String.include?(string)` - Returns `true` if *str* contains the given string or character. Very simple!

```
> "hello".include? "lo"   #=> true
> "hello".include? "ol"   #=> false
```

- `String.gsub(pattern, <hash|replacement>)` - Returns a new string with *all* the occurrences of the *pattern* substituted for the second argument: . The pattern is typically a *Regexp*, but a string can also be used.

```
"hello".gsub(/[aeiou]/, '*')              #=> "h*ll*"
"hello".gsub(/([aeiou])/, '')          #=> "hll"
```

Either method will depend upon the problem you are trying to solve, and the nature of input-output behavior you desire.

In this challenge, your task is to write the following methods:

- `mask_article` which appends strike tags around certain words in a text. The method takes 2 arguments: A string and an array of *words*. It then replaces all the instances of *words* in the text with the modified version.
- A helper method `strike`, given one string, appends strike off HTML tags around it. The strike off HTML tag is `<strike></strike>`.

For example:

```
> strike("Meow!") # => "<strike>Meow!</strike>"
> strike("Foolan Barik") # => "<strike>Foolan Barik</strike>"
> mask_article("Hello World! This is crap!", ["crap"])
"Hello World! This is <strike>crap</strike>!"
```

Apply the helper method in completing your main method.