Exercise 2

Naive URL Validation

Expected Time: 15 min

Create a regular expression to validate URLs. The URLs should start with `http://` or `https://` and can contain letters, numbers, periods, hyphens, and slashes. Use the following snippet as a test case:

```
http://condense.tech # => OK
ftp://condense.tech # => NO
http:condense.tech # => NO
http//condense.tech # => NO
http:/condense.tech # => NO
http://asdadajksdad.1231239-asdasd # => OK
http://asdasd   asdasdad # => NO
http://cond,com # => NO
```

Phone Number Extraction

Expected Time: 15 min

Given a string that contains various text, including phone numbers in the format `(123) 456-7890`, write a regular expression to extract all the phone numbers from the string. Use the following snippet as a test case:

```
My phone number is (039) 02-121212 # => (039) 02-121212
None of my 32 friends has a phone. One of them gave me the number 123-123123 but it isn't valid # => no match
Phone numbers cannot have letters. Numbers like (800) STAR-121212 are not valid # => no match
Every country has its own rules for phone numbers. Even a number like (1) 1-2 can be a valid phone number
```

HTML Tag Matching

Expected Time: 30 min

Create a regular expression to match and extract the content within HTML anchor tags (`<a></a>`). For example, given the input `<a href="https://example.com">Visit Example</a>`, the regular expression should extract `Visit Example` as the output.