Source File: `~/public_html/lab09.php`
Input: URL
Output: Standard Output
Value: 2

The ***Luhn algorithm*** or ***Luhn formula***, also known as the "modulus 10" or "mod 10" algorithm, is a simple checksum formula used to validate a variety of identification numbers, such as credit card numbers, IMEI numbers, National Provider Identifier numbers in US and Canadian Social Insurance Numbers. It was created by IBM scientist Hans Peter Luhn and described in U.S. Patent No. 2,950,048, filed on January 6, 1954, and granted on August 23, 1960.

The algorithm is in the public domain and is in wide use today. It is specified in ISO/IEC 7812-1. It is not intended to be a cryptographically secure hash function; it was designed to protect against accidental errors, not malicious attacks. Most credit cards and many government identification numbers use the algorithm as a simple method of distinguishing valid numbers from collections of random digits.

The formula verifies a number against its included check digit, which is usually appended to a partial account number to generate the full account number. This account number must pass the following test:

1. Counting from the check digit, which is the rightmost, and moving left, double the value of every second digit.
2. Sum the digits of the products (e.g., $10 \Rightarrow 1$, $14 \Rightarrow 5$) together with the undoubled digits from the original number.
3. If the total modulo 10 is equal to 0 then the number is valid according to the Luhn formula; else it is not valid.

Assume an example of an account number `49927398716`. The Luhn algorithm would be applied as follows:

Account number	4	9	9	2	7	3	9	8	7	1	6
Double every other	4	**18**	9	**4**	7	**6**	9	**16**	7	**2**	6
Sum (right-to-left)	70	66	57	48	44	37	31	22	15	8	6

Since the sum (70) modulo 10 is 0, the account number is probably valid.

Your program will accept a string from the URL via which the program is accessed. Let the input string represent an account number and determine its validity using the Luhn algorithm. The Luhn algorithm will work for account numbers of any number of digits; however, your program should verify that the input string has 16 or fewer characters. The output should be formatted as shown in the instructor's version of this program. The instructor used `printf` (see `http://php.net/manual/en/function.printf.php`). The first argument to `printf` is a format string. Information about the format string can be found at `http://php.net/manual/en/function.sprintf.php`.

Some additional notes for this assignment:

- Insert an HTML comment at the top of the document identifying you as the author, the class, and the assignment number.
- Add an echo statement to the beginning of the script section that will display your name, the course number, and the assignment number.

- Since this assignment uses several PHP code blocks, it's always a good idea to check for syntax errors. You can do this by using the `-l` option to the `php` command at the command line as in

```
1 newuser@csunix ~/public_html> php -l lab09.php
2 No syntax errors detected in lab09.php
```

- You should always validate the rendered HTML code. The validator is discussed near the top of p. 6 and in Appendix A on pp. 629–631. By including the following link and image, a user will be able to click the image and receive a report from the validator.

```
1 <?php
2   $location = 'https://' . $_SERVER['HTTP_HOST'] . $_SERVER['REQUEST_URI'];
3   $location = urlencode($location);
4   echo '<a href="https://validator.w3.org/nu/?doc=' . $location . '">';
5 ?>
6   <img src="https://www.w3.org/QA/Tools/I_heart_validator"
7    alt="I heart Validator logo" height="31" width="80" />
8 </a>
```

After the document is valid, open it in your Web browser to see how it renders.

Upon completion of this assignment, submit your source file via Blackboard.