Source File: `~/public_html/lab08.php`
Input: URL
Output: Standard Output
Value: 3

The purpose of this assignment is to write your own version of a base 64 encoder. PHP has a builtin function that accomplishes this (see `https://www.php.net/manual/en/function.base64-encode.php`), but I want you to write your own. Some useful background information about base 64 encoding and decoding can be found at `https://en.wikipedia.org/wiki/Base64`. Pay special attention to the first three sections of this article.

Your program will accept a string from the URL via which the program is accessed. Let this string represent the string that is to be encoded. The output should be formatted as shown in the instructor's version of this program. You may find it helpful to view the page source after executing some of the test cases.

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.
- You can obtain the ASCII code for a character with the `ord` function. See `https://www.php.net/manual/en/function.ord.php`.
- All of the bitwise operators familiar to you from C++ are available in PHP. See `https://www.php.net/manual/en/language.operators.bitwise.php`.
- When displaying the original input string and the resulting base 64 encoded string, surround each of these sections with `<pre>` tags so that a monospaced font will be used.
- When displaying the base 64 encoded string, display 76 characters per line. See `https://www.php.net/manual/en/function.chunk-split.php`.
- 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 lab08.php
2 No syntax errors detected in lab08.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.