Passing Variables Exercises.

Exercise One

Here is a PHP program that extracts and displays two of the variables provided by the web server in the $_SERVER superglobal array. It then displays the full contents of the superglobal.

```
<?php
$host = $_SERVER['HTTP_HOST'];
$agent = $_SERVER['HTTP_USER_AGENT'];

echo "<html><body>";

echo $host;
echo "<br></br>";
echo $agent;
echo "<br></br>Full Contents of Server Array<br></br>";
print_r($_SERVER);

echo "</body></html>";
?>
```

Try it out, and then modify it to see if the college webserver also implements the $_ENV superglobal array.

Exercise Two

Here are the two pages which accept a username and password and check them for correctness.

Save this as:

login.html

```
<html>
  <head></head>
  <body>
    <!-- Form with two fields and a submit button -->
    <form name="logscreen" action="check.php" method="get">
      User Name:
      <input type="text" name="user"></input>
      <br />
      Password:
      <input type="text" name="pass"></input>
      <br />
      <input type="submit" value="Login"></input>
    </form>
  </body>
</html>
```

and save the next page as:

check.php

```
<?php

// Get the values from the URL and copy them into variables
$uname = $_GET['user'];
$pword = $_GET['pass'];

// Compare the variables with the valid name and password and redirect
// user to the appropriate page

if (($uname == "kevin") && ($pword == "room123"))
    header( "Location: listing.html");
else
    header( "Location: forbidden.html");

?>
```

You will also need a file called **listing.html** and another called **forbidden.html**.

They should just display messages such as:

Welcome to my website

and

You are not allowed to access this website

Modify this so that it uses the POST method instead of the GET method to pass it's values to the second page. Confirm that it is more secure.

Exercise Three

Here is the start of a PHP program which can be used as a simple calculator:

calc.html

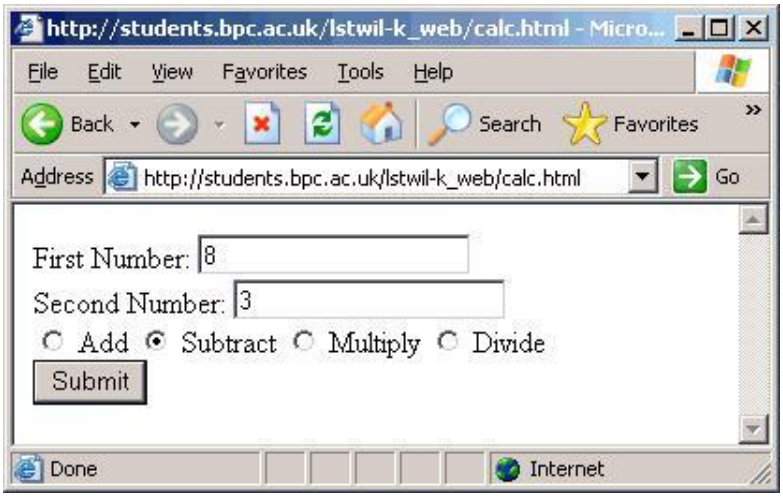
```
<html>
  <head></head>
  <body>
    <form name="calculator" action="result.php" method="get">
      First Number:
      <input type="text" name="num1"></input>
      <br />
      Second Number:
      <input type="text" name="num2"></input>
      <br />
      <input type="submit" value="Add"></input>
    </form>
  </body>
</html>
```

result.php

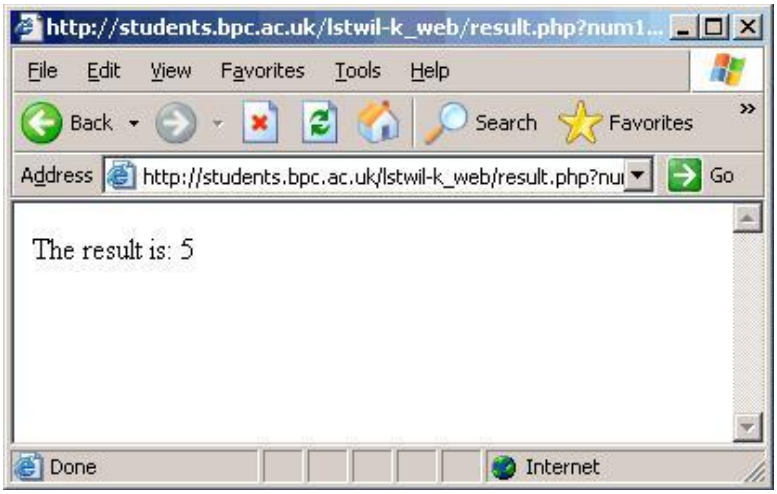
```
<?php

// Get the values from the URL and copy them into variables
$first = $_GET['num1'];
$second = $_GET['num2'];

$answer = $first + $second;
?>

<html>
  <body>
    The result is:  <?php echo $answer; ?>

  </body>
</html>
```

Try this out and see if it works.

Modify **calc.html** so that it produces something like this:



and modify **result.php** so that it will take the type of operation into account and produce the correct answer like this:



Passing Variables Answers.

Exercise Three

calc.html

```
<html>
  <head></head>
  <body>
    <form name="calculator" action="result.php" method="get">

      First Number:
      <input type="text" name="num1"></input>
      <br />
      Second Number:
      <input type="text" name="num2"></input>
      <br />

      <input type="radio" name="op" value="add"> Add </input>
      <input type="radio" name="op" value="sub"> Subtract </input>
      <input type="radio" name="op" value="mul"> Multiply </input>
      <input type="radio" name="op" value="div"> Divide </input>
      <br />
      <input type="submit" value="Submit"></input>
    </form>
  </body>
</html>
```

result.php

```
<?php

$first = $_GET['num1'];
$second = $_GET['num2'];
$operation = $_GET['op'];

switch ($operation )
  {
    case "add":  $answer = $first + $second; break;
    case "sub":  $answer = $first - $second; break;
    case "mul":  $answer = $first * $second; break;
    case "div":  $answer = $first / $second; break;
  }

?>

<html>
  <body>
    The result is:  <?php echo $answer; ?>
  </body>
</html>
```