How I approach Trial Lab Test 1 Set A

[**First step**](#_8xjzxvnqlwr2) **1**

[**Q1 Part A**](#_hqpjt8w1bdlr) **1**

[**Q1 Part B**](#_odouk3jd08q8) **3**

[**Q2 Part A**](#_kitwi4hwug85) **5**

[**Q2 Part B**](#_jgdzi4tu1ue0) **7**

[**Q3**](#_cyh98beclno1) **9**

[**Summary**](#_60lv37w216cj) **15**

# Download Files (used in this tutorial)

Click [here](https://smu.sg/2019-is113-trialLT1SetA-approach)

# First step

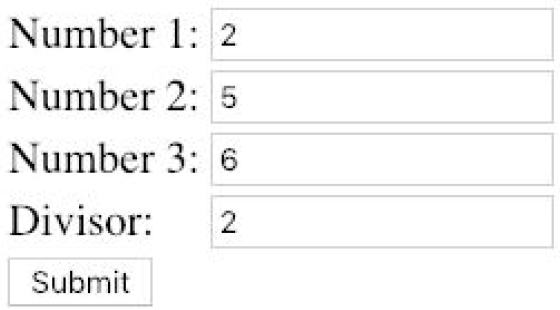
For all assessments, I will go through the whole paper first to decide which question I want to tackle first and approximately how much time I need on each question. Personally, I will solve questions that I’ve confidence in first to ensure I have sufficient time to work on them properly so as to secure the marks. I will tackle questions that I’m unsure of later.

My aim is to get as many marks as possible within the time limits. It is not to attempt all questions.

# Q1 Part A

Enter name and email first.

|  |
| --- |
| <!DOCTYPE html>  <!--  Name: Wilson Alexander Daniel the First  Email: wad1@smu.edu.sg  -->  <html> ... |

* *1. Update q1-A.html to include a form. The page displays as follows:*
* **
* *Note : "2", "5", "6" and "2" (divisor) are values entered by the user.*

The given resource q1-A.html has all the text fields and submit button required.

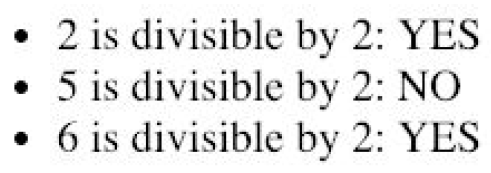
* *Upon clicking the SUBMIT button, it submits to q1-A.php via HTTP POST.*

|  |
| --- |
| ...  <html>  <body>  <form action='q1-A.php' method='post'>  <table>  <tr>  <td>Number 1:</td>  <td><input type='text' name='num1'></td>  </tr>  ...  <tr>  <td colspan='2'><input type='submit' name='check'></td>  </tr>  </table>  </form>  </body>  </html> |

**Test** that the form is submitted to q1-A.php. Don’t let the bugs accumulate.

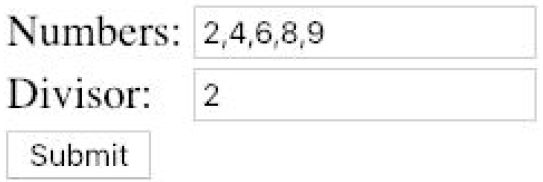
* *2. Implement a function called is\_divisible\_by($num, $n) in q1-A.php . The function returns true*
* *if $num is divisible by $n , false otherwise.*

|  |
| --- |
| <?php  /\*  Name: Wilson Alexander Daniel the First  Email: wad1@smu.edu.sg  \*/  function is\_divisible\_by($num, $n) {  return $num % $n == 0;  }  **// Write some test code for the function & test**  // var\_dump( is\_divisible\_by(4, 2) ); // true  // var\_dump( is\_divisible\_by(4, 3) ); // false  ?> |

* *3. Use the function is\_divisible\_by($num, $n) to produce the following result when the user submits the values shown above through q1-A.html :*
* **
* *HINT: Explore the use of HTML unordered list <ul>*

|  |
| --- |
| <?php  /\*  Name: Wilson Alexander Daniel the First  Email: wad1@smu.edu.sg  \*/  function is\_divisible\_by($num, $n) {  return $num % $n == 0;  }  **// process form**  **/\***  **I notice that I'm going to repeat the same steps for the 3 numbers.**  **Hence, I put them in an array so that I can use a loop and not repeat code.**  **\*/**  $nums = [ $\_POST['num1'], $\_POST['num2'], $\_POST['num3'] ];  $divisor = $\_POST['divisor'];  ?>  <html>  <body>  <ul>  <?php  foreach ($nums as $num) {  if ( is\_divisible\_by($num, $divisor) ) {  $results = 'YES';  } else {  $results = 'NO';  }    **// 1. display HTML**  **// 2. introduce variables for parts that change; e.g. $results**  **// 3. determine the value of the variables before echo**  echo "  <li>$num is divisible by $divisor: $results </li>  ";  }  ?>  </ul>  </body>  </html> |

# Q1 Part B

* *1. Update q1-B.html to include a form. The page displays as shown below. This time, one or more numbers are input together in ONE form input field where the numbers are separated by a comma :*
* **
* *Note : "2,4,6,8,9" and "2" are values entered by the user.*

The given resource q1-A.html has all the text fields and submit button required.

Enter name and email first.

|  |
| --- |
| <!DOCTYPE html>  <!--  Name: Wilson Alexander Daniel the First  Email: wad1@smu.edu.sg  -->  ... |

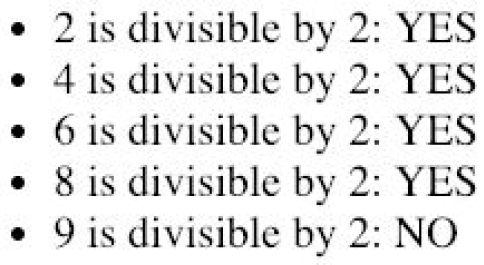
* *Upon clicking the SUBMIT button, it submits to q1-B.php via HTTP GET.*

|  |
| --- |
| ...  <html>  <body>  <form action='q1-B.php'>  <table>  <tr>  <td>Numbers:</td>  <td><input type='text' name='numbers'></td>  </tr>  ...  <tr>  <td colspan='2'><input type='submit' name='check'></td>  </tr>  </table>  </form>  </body>  </html> |

Test in browser that the form is submitted to q1-B.php. Don’t let the bugs accumulate.

* *2. q1-B.php uses the same function is\_divisible\_by($num, $n) that you implemented in Part A .*
* *Copy and paste the function from q1-A.php to q1-B.php .*

|  |
| --- |
| <?php  /\*  Name: Wilson Alexander Daniel the First  Email: wad1@smu.edu.sg  \*/  // copy and paste  function is\_divisible\_by($num, $n) {  return $num % $n == 0;  }  ?> |

* *3. Use the function is\_divisible\_by($num, $n) to produce the following result when the user submits the values shown above through q1-B.html :*
* **
* *HINT : Explore the use of PHP explode function*

Again, an unordered list is required.

Function explode() is for splitting up the submitted value of text field ‘numbers’.

|  |
| --- |
| <?php  …  function is\_divisible\_by($num, $n) {  return $num % $n == 0;  }  // process form  /\*  Basically, all of my code in q1-A.php can be copied and pasted here with modifications to the following 2 lines.  \*/  $nums = explode(',', $\_GET['numbers'] );  $divisor = $\_GET['divisor'];  ?>  <html>  …  </html> |

# Q2 Part A

* *1. Modify q2.php such that if the user clicks on the words ("Apple", "Orange" or "Pear"), the corresponding checkboxes will be selected/unselected.*

We need <label> tags.

|  |
| --- |
| <!--  Name: Wilson Alexander Daniel the First  Email: wad1@smu.edu.sg  -->  <html>  <body>  <form method='post' action='q2-display.php'>  <label>  <input type="checkbox" value="apple" name="fruit[]">Apple  </label>  <label>  <input type="checkbox" value="orange" name="fruit[]">Orange  </label>  <label>  <input type="checkbox" value="pear" name="fruit[]">Pear  </label>  <br>  <input type='submit'>  </form>  </body>  </html> |

Test that the labels work correctly.

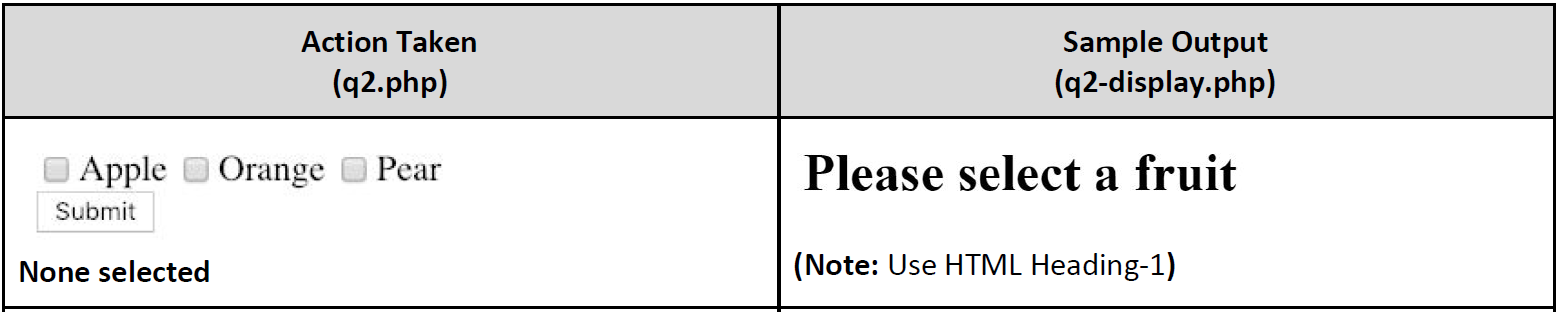
* *2. Once the user selects the fruits and press on the SUBMIT button on page q2.php , the form submits to q2-display.php .*

The given resource has the <form> tag and is correct.

* *3. Modify q2-display.php such that appropriate images will be displayed in a tabular format (use border='1' for your table).*
* *○ For example, if the user selects “Apple” in q2.php , then page q2-display.php must display the image file apple.jpg .*

|  |
| --- |
| <?php  /\*  Name: Wilson Alexander Daniel the First  Email: wad1@smu.edu.sg  \*/  // process form  $fruits = $\_POST['fruit'];  ?>  <html>  <body>  <table border='1'>  <?php  foreach ($fruits as $fruit) {  echo "  <tr>  <td> <img src='$fruit.jpg' /> </td>  </tr>  ";  }  ?>  </table>  </body>  </html> |

Test. Don’t let the bugs accumulate.

* 

This scenario is not handled properly.

|  |
| --- |
| <?php  /\*  Name: Wilson Alexander Daniel the First  Email: wad1@smu.edu.sg  \*/  // default values  $msg = '';  $fruits = [];  // process form  if ( isset($\_POST['fruit']) ) {  // 1 or more checkbox selected  $fruits = $\_POST['fruit'];  } else {  // no checkbox selected  $msg = 'Please select a fruit';  }  ?>  <html>  <body>  <!--  1. This is the HTML required.  2. Introduce variable $msg as it may change.  3. Determine value of $msg above.  -->  <h1><?=$msg?></h1>  <table border='1'>  <?php  // As I default $fruits to empty array, foreach loops zero time.  foreach ($fruits as $fruit) {  echo "  <tr>  <td> <img src='$fruit.jpg' /> </td>  </tr>  ";  }  ?>  </table>  </body>  </html> |

Test all scenarios given in the question paper plus more that you can think of.

# Q2 Part B

* *1. In page q2-one.php , once the user selects the fruits and press on the SUBMIT button on page, the form submits back to itself (to q2-one.php ).*

The given resource has the correct <form> tag.

|  |
| --- |
| <?php  /\*  Name: Wilson Alexander Daniel the First  Email: wad1@smu.edu.sg  \*/  ?>  <html>  <body>  <?php  ?>  <form method='post' action='q2-one.php'>  <input type="checkbox" value="apple" name="fruit[]">Apple  <input type="checkbox" value="orange" name="fruit[]">Orange  <input type="checkbox" value="pear" name="fruit[]">Pear  <br>  <input type='submit' name='send'>  </form>  </body>  </html> |

* *2. Modify q2-one.php such that:*
* *○ Appropriate images will be displayed in a tabular format (use border='1' for your table).*
* *○ Appropriate message will be displayed at the top of the page.*

Basically, I can reuse code in q2-display.php with modifications.

|  |
| --- |
| <?php  ...  // default values  $msg = '';  $fruits = [];  $formSubmitted = false;  // Check for the presence of submit button’s name  if ( isset($\_POST['send'])) {  $formSubmitted = true;  // process form  if ( isset($\_POST['fruit']) ) {  // 1 or more checkbox selected  $fruits = $\_POST['fruit'];  // handle the appropriate message  $count = count($fruits);  if ($count == 1) {  $msg = 'You selected 1 fruit';  } else {  $msg = "You selected $count fruits";  }  } else {  // no checkbox selected  $msg = 'Please select a fruit';  }  }  ?>  <html>  <body>  <?php  if ( $formSubmitted ) {  ?>  <h1><?=$msg?></h1>  <table border='1'>  <?php  foreach ($fruits as $fruit) {  echo "  <tr>  <td> <img src='$fruit.jpg' /> </td>  </tr>  ";  }  ?>  </table>  <?php  }  ?>  <form method='post' action='q2-one.php'>  <input type="checkbox" value="apple" name="fruit[]">Apple  <input type="checkbox" value="orange" name="fruit[]">Orange  <input type="checkbox" value="pear" name="fruit[]">Pear  <br>  <input type='submit' name='send'>  </form>  </body>  </html> |

Test all scenarios given in the question paper plus more that you can think of. Don’t let the bugs accumulate.

**Note:**

* When isset( $\_POST['fruit' ] ) returns false, it may means
  + There is NO form submitted aka the page is loaded for the first time, OR
  + There IS form submitted but no checkbox is selected.
* Hence, it cannot be used to determine if a form submitted or not.
* The above point is true for
  + checkboxes,
  + radio buttons with no default checked value,
  + multi-select drop down list.
* We have to identify other form fields to be used to determine form submission. In this case, we make use of the submit button.

# Q3

3-star questions tend to require a lot of time (tedious or complicated) to solve but the marks are little. Low ROI. My approach is to solve as many little parts as I can when I am sure I have solved all the other questions that I’m confident in.

The **first** part that I will tackle is the function getStudentNames($students).

|  |
| --- |
| <?php  /\*  Name: Wilson Alexander Daniel the First  Email: wad1@smu.edu.sg  \*/  …  // INPUT : $students Array  // OUTPUT : Return an Array of student names (String)  function getStudentNames($students) {  $arr = [];  // Part A  // YOUR CODE GOES HERE  foreach ($students as $student) {  $arr[] = $student['name'];  }  return $arr;  /\*  If I dunno how to solve this part,  I will return a hardcoded array (i.e. give up marks for this part)  so that I can try the drop down list part.  return [ 'Jong Un Kim', 'Donald Trump', 'Hugo Chavez' ];  \*/  }  ... |

* *1. Complete q3.php such that the name drop-down menu will display the distinct names of students.*

This is the **second** part that I will tackle.

|  |
| --- |
| …  Name:  <select name='student\_name'>  <?php  // Part A  // YOUR CODE GOES HERE  $student\_names = getStudentNames($students); // DO NOT MODIFY THIS LINE  // YOUR CODE CONTINUES HERE  foreach ($student\_names as $name) {  echo "  <option value='$name'> $name </option>  ";  }  ?>  </select>  ... |

Test to verify the drop down list is correct.

* *2. When a student’s name is selected in the drop-down menu and the “ Show Timetable ” SUBMIT button is clicked, the form submits to itself (to q3.php page). It then:*
* *...*
* *b. The student’s name should be " selected " in the drop-down menu.*

|  |
| --- |
| <?php  …  // default value  $student\_name = '';  // process form  if ( isset($\_POST['student\_name']) ) {  $student\_name = $\_POST['student\_name'];  }  ?>  ...  Name:  <select name='student\_name'>  <?php  // Part A  // YOUR CODE GOES HERE  $student\_names = getStudentNames($students); // DO NOT MODIFY THIS LINE  // YOUR CODE CONTINUES HERE  foreach ($student\_names as $name) {  if ( $student\_name == $name) {  $selected = 'selected';  } else {  $selected = '';  }  // 1. Introduce variable $selected for parts that change  // 2. Determine the value of $selected before echo  echo "  <option value='$name' $selected> $name </option>  ";  }  ?>  </select>  ... |

Test to verify it works correctly. Don’t let the bugs accumulate.

* *3. When the page loads for the first time, the timetable displayed will be for the first student in the drop-down menu (follow the order of student names - as returned by the getStudentNames() function).*

|  |
| --- |
| …  <select name='student\_name'>  <?php  // Part A  // YOUR CODE GOES HERE  $student\_names = getStudentNames($students); // DO NOT MODIFY THIS LINE  // YOUR CODE CONTINUES HERE  $student\_name = getStudentNames($students)[0];  … |

* *2. …*
* *a. Displays the student’s timetable …*

|  |
| --- |
| <?php  ...  // You can create your own function to keep your code simple  function getTimetable($student\_name, $students) {  foreach ($students as $student) {  if ( $student\_name == $student['name'] ) {  return $student['courses'];  }  }  return false;  }  $courses = getTimetable($student\_name, $students);  ...  ?>  ...  <?php  var\_dump($courses); // Testing purpose  ?>  </body>  </html> |

* *2. ...*
* *a. Displays the student’s timetable in an HTML table; and*
* *…*
* *4. Marks will be deducted for overly repetitive code that could have been simplified, e.g., by using a loop, or by writing a function.*

I will try to create an empty table first.

|  |
| --- |
| …  <table border='1'>  <tr>  <th></th>  <th>08:30am - 10:00am</th>  <th>10:00am - 11:30am</th>  <th>12:00nn - 1:30pm</th>  <th>1:30pm - 3:00pm</th>  <th>3:00pm - 4:30pm</th>  <th>4:30pm - 6:00pm</th>  </tr>  <?php  $courses = getTimetable($student\_name, $students);  // I know I will need the day and time for each table cell.  $dayArr = [ 'MON', 'TUE', 'WED', 'THU' , 'FRI'];  $timeArr = [ '0830', '1000', '1200', '1300', '1500', '1630' ];  foreach ($dayArr as $day) {  echo "  <tr>  <td> $day </td>  ";  foreach ($timeArr as $time) {  echo "  <td></td>  ";  }  echo "  </tr>  ";  }  ?>  </table>  … |

Test to check empty table. Don’t let the bugs accumulate.

* *2. ...*
* *a. Displays the student’s timetable in an HTML table; and*
* *…*
* *HINT: Explore the use of align='center' attribute of <td>*

Next I will try to display the course details in each cell without caringly about colspan.

|  |
| --- |
| …  // Again, I create a function to keep my code simple  function getCourse($day, $time, $courses) {  foreach ($courses as $course) {  // E.g. ['IS111', 'Intro Programming', 'MON', '1000', 1]  if ( $day == $course[2] && $time == $course[3]) {  return $course;  }  }  return false;  }  …  ?>  ...  foreach ($dayArr as $day) {  echo "  <tr>  <td> $day </td>  ";  foreach ($timeArr as $time) {  $course = getCourse($day, $time, $courses);  if ($course === false) {  echo "  <td></td>  ";  } else {  // E.g. ['IS111', 'Intro Programming', 'MON', '1000', 1]  echo "  <td align='center'>  {$course[0]} <br/>  {$course[1]}  </td>  ";  }  }  echo "  </tr>  ";  }  ... |

Test to check table.

Finally, I tackle the colspan.

|  |
| --- |
| …  // default  $skipped = false;  foreach ($timeArr as $time) {    if ( $skipped ) {  // skip printing <td>  // reset $skipped  $skipped = false;  } else{  $course = getCourse($day, $time, $courses);  if ($course === false) {  echo "  <td></td>  ";  } else {  // E.g. ['IS111', 'Intro Programming', 'MON', '1000', 1]  $colspan = $course[4];  echo "  <td align='center' colspan='$colspan' >  {$course[0]} <br/>  {$course[1]}  </td>  ";    if ( $colspan == 2) {  /\*  1. I introduce a variable $skipped to keep track whether  to skip printing the next <td>.  2. Then I decide what is the default value of $skipped.  3. And when/where should I reset $skipped to false.  \*/  $skipped = true; // skip printing next cell  }  }  }  }  ... |

Note, there are other ways to solve this.

# Summary

1. Read through the whole paper first.
2. Aim to get as many marks as possible within the time limits.
3. Break down a complex problem into smaller parts.
4. Solve one small part at a time.
5. Test a lot and frequently. Don’t let the bugs accumulate.