WAD Trial Lab Test 1 (1.5 hours) - Set B [20 marks]

General Instructions:

- You can refer to any offline resources already on your laptop, but you must disable all networking and Bluetooth connections during the test. You must not communicate with anyone via any means during the test.

- Just before the test, you will be given instructions by the invigilator as to how to obtain resource files required for the lab test and how to submit your solutions.

- No questions will be entertained during the test. If necessary, make your own assumptions.

- You are allowed to use only standard PHP classes and functions in your solutions – do not use any third party libraries.

- Use meaningful names for classes, methods, functions and variables, as well as indent your code correctly. Use 4 spaces for indentation. Otherwise, you may attract penalty of up to **20%** of your score for the corresponding question.

- You **MUST** include your name as author in the comments of all your submitted source files. Failure to do so WILL attract a penalty of up to **20%** of your score for the corresponding question.

  For example, if your registered name is "Ye Jin SON" and email ID is yejin.son.2020, include the following comment at the beginning of each source file you write.

  ```
  <!--

      Name:  Ye Jin SON

      Email: yejin.son.2020

  -->
  ```

- You may wish to comment out the parts in your code which cause errors. But commented code will not be marked.

- **Resources:** Click here
- **Solutions:** Click here

DO NOT TURN OVER UNTIL YOU ARE TOLD TO DO SO

Question 1 (Difficulty Level: */*)   **[8 marks]**

Given:
- `q1-A.html`
- `q1-A.php`
- `q1-B.php`
- `q1-B-display.php`
- (Image Files) `trump.jpg, clinton.jpg, kim.jpg, moon.jpg`

Part A (4 marks) - Difficulty Level (*)

1. Modify `q1-A.html` such that if the user clicks on the words ("Trump", "Clinton", "Kim", or "Moon"), the corresponding radio button options will be selected/unselected.
2. Once the user selects an option (e.g. a person's name) and click on the SUBMIT button on page `q1-A.html`, the form submits to `q1-A.php`.
3. Modify `q1-A.php` such that appropriate message will be displayed at the top of the page using Heading-1.
   - For example, if the user selects "Trump" in `q1-A.html`, then `q1-A.php` must display:

   ## Make America Great Again

4. Modify `q1-A.php` such that appropriate image will be displayed.
   - For example, if the user selects "Trump" in `q1-A.html`, then `q1-A.php` must display the image file `trump.jpg`.

Sample Output

Action Taken (q1-A.html)	Sample Output (q1-A.php)
○ Trump ○ Clinton ○ Kim ○ Moon [Submit] **None selected**	## You must select a person! **(Note:** Use HTML Heading-1**)**
⦿ Trump ○ Clinton ○ Kim ○ Moon [Submit] **"Trump" selected**	## Make America Great Again

Part B (4 marks) - Difficulty Level (*)

1. Modify `q1-B.php` such that it displays people from the associative array `$people`. It displays them in an HTML table as shown below.

Person ☐ Donald Trump ☐ Hillary Clinton ☐ Kim Jong Un ☐ Moon Jae In Submit	Each person is displayed in a new table row as a checkbox option. For example, for "Donald Trump": ● The checkbox option **value** is **"trump"**. ● The label/text associated with this checkbox option is **"Donald Trump"**. ● This information can be retrieved from the associative array `$people`.

2. Once the user selects people and clicks on the SUBMIT button on page `q1-B.php`, the form submits to `q1-B-display.php`.
3. Modify `q1-B-display.php` such that appropriate messages **AND** images will be displayed in an HTML table.

Sample Output

Action Taken (q1-B.php)	Sample Output (q1-B-display.php)
Person ☐ Donald Trump ☐ Hillary Clinton ☐ Kim Jong Un ☐ Moon Jae In Submit **None selected**	# You must select at least one person! **(Note:** Use HTML Heading-1**)**
Person ☐ Donald Trump ☐ Hillary Clinton ☑ Kim Jong Un ☑ Moon Jae In Submit **Two options selected**	

Question 2 (Difficulty Level: */**/**)   [6 marks]

Given:
- `q2.html`
- `q2-calculate.php`

q2.html

Page **`q2.html`** allows the user to specify:
1. **`quantity`:** Number of __number sets__ where each **number set** consists of THREE (3) integers ranging between ZERO (0) and NINE(9), both numbers inclusive.
   - E.g. **2, 7, 9** → this is an example of a **number set**, and the THREE (3) integers are **randomly generated** in page **`q2-calculate.php`**.
2. **`lucky_number`**: Integer ranging between ZERO (0) and NINE(9), both numbers inclusive)
3. **`bet_amount`**: Numeric

Upon clicking the SUBMIT button, the form submits to **`q2-calculate.php`**.

Part A (1 mark) - Difficulty Level (*)

1. Complete page **`q2-calculate.php`** such that it retrieves the following user input from the form.
   - **`quantity`**
   - **`lucky_number`**
   - **`bet_amount`**

2. The page then must display the information at the top of the page using HTML Heading-3.

q2.html (user input)	q2-calculate.php (after form submission)
	Lucky Number: 7 **Bet Amount: 10**

Part B (2 marks) - Difficulty Level (**)

1. Inside `q2-calculate.php`, complete the function `generateRandomSets($quantity)`.
2. The function takes ONE (1) parameter `$quantity`, which is a user-defined value (from **Part A**).
3. The function generates `$quantity` number of **number sets**, where:
    o **Each number set** consists of THREE (3) **integers** and;
    o Each **integer** ranges between ZERO (0) and NINE (9), both numbers inclusive.
4. Please see the partial implementation of the function inside the file for details of what the return value looks like.

HINT: Explore the use of PHP **rand** function

Part C (3 marks) - Difficulty Level (**)

1. Complete `q2-calculate.php`. Inside `q2-calculate.php`, complete the function `calculate($random_sets, $lucky_number)`.
2. The function takes TWO (2) parameters:
    o `$random_sets` is the value returned by the function `generateRandomSets()` (from **Part B**).
        ■ It is an **(indexed) array of (indexed) arrays**.
    o `$lucky_number` is the user-defined form input (from **Part A**).
3. The function checks *each* **number set** and calculates **number of matches** for that **number set**.
4. Please see the partial implementation of the function inside the file for details of what the return value looks like.

Sample Input & Output

q2.html (user input)	q2-calculate.php (after form submission)
How many sets do you want? `3 ▾` Choose Your Lucky Number (0 to 9, INTEGER only) `7` Bet Amount (NUMERIC) `10` `Submit`	**Lucky Number: 7** **Bet Amount: 10** <table><tr><td>Number Set</td><td>Number of Matches</td><td>Winning Amount</td></tr><tr><td>0, 8, 9</td><td>0</td><td>0</td></tr><tr><td>1, 9, 7</td><td>1</td><td>10</td></tr><tr><td>7, 4, 7</td><td>2</td><td>20</td></tr><tr><td>Total Winning Amount</td><td>30</td></tr></table>

- The user also chose THREE (**3**) **sets** of numbers in `q2.html`.
- The user chose number **7** as the **lucky_number** in `q2.html` with the **bet_amount** of **10** in `q2.html`.
- The user clicks on the SUBMIT button.
- Page `q2-calculate.php` generated THREE (**3**) **sets** of numbers, where **each set** consists of THREE (3) randomly selected integers.
- `q2-calculate.php` displays the user input **lucky_number** and **bet_amount** at the top of the page.
- For *each* **number set**, the page calculates **winning amount** and displays it in an HTML table.
    o Above, the user's second number set (1, 9, 7) contains **lucky_number** (7) ONCE. Hence, the winning amount for **this set** is **10** (**bet_amount** of 10 *times* 1, which is the number of occurrences of the **lucky_number**).
    o The third number set (7, 4, 7) contains **lucky_number** (7) TWICE. Hence, the winning amount for **this set** is **20** (**bet_amount** of 10 *times* 2).
    o Finally, the **Total Winning Amount** is calculated to be **30** as it is the sum of all winning amounts.

HINT: Explore the use of PHP **implode** function for displaying numbers in "Number Set" column in the above table
Question 3 (Difficulty Level: *//**/***)** [6 marks]

Given:
- `q3.php`

NOTE: Marks will be deducted for overly repetitive code that could have been simplified, e.g., by using a loop, or by writing a function.

Part A (0.5 mark) - Difficulty (*)

Complete `q3.php` such that it displays people's names in an HTML **table format** as shown below:

- Do NOT **HARD-CODE** the names.
- Make use of `$people` associative array provided in the resource file.

Part B (1.5 mark) - Difficulty (**)

After the user makes selections and clicks the SUBMIT button in `q3.php`, the form submits back to `q3.php`.

Complete `q3.php` such that it displays appropriate **messages** at the top of the page using Heading-1.
- If no one is selected, display **"You didn't select anyone! Select at least THREE (3) people!"**
- If one or two people are selected, display **"Select at least THREE (3) people!"**
- Page `q3.php` must remember the user's original selections (if any) and display it correctly.

q3.php (user input)	q3.php (after form submission)
 None selected	
 Less than 3 people selected	

Part C (2 marks) - Difficulty (**)

Modify `q3.php` as per the below requirements.

If the user selects THREE (3) or more people in `q3.php` as shown below:

and clicks on the SUBMIT button, `q3.php` must display:
- An HTML table listing all people where the user's selections are **checked**.
- An HTML table showing the selected people's images.
  - It is an **N by N matrix**.
  - The displayed images are selected **randomly** with an equal probability of each person being selected.

HINT: Explore the use of PHP **rand** function

Sample Output

Part D (2 marks) - Difficulty (***)

Continuing from **Part C**, modify `q3.php` to check if all of the displayed images on the **diagonal** are the **same**. Specifically,

1) All images on the **diagonal** from the **Top Left Corner** to the **Bottom Right Corner** are the same (showing the same person), **OR**

2) All images on the **diagonal** from the **Top Right Corner** to the **Bottom Left Corner** are the same (showing the same person)

Display the following messages if **1)** or **2)** are satisfied, using HTML Heading-1.
- If **1)** is satisfied, display **"Top Left to Bottom Right Diagonal FOUND"**.
- If **2)** is satisfied, display **"Top Right to Bottom Left Diagonal FOUND"**.

Sample Output 1	Sample Output 2
 Top Right to Bottom Left Diagonal FOUND	 **Top Left to Bottom Right Diagonal FOUND** **Top Right to Bottom Left Diagonal FOUND**

- END -