**ELITE EXAM**

**PHP**

**“SHORTEST WORD”**

You have given a string of words, return the length of the shortest word(s).

String will never be empty, and you do not need to account for different data types.

TEST CASES:

TEST CASE: “TRUE FRIENDS ARE ME AND YOU”

OUTPUT: **2** – BECAUSE THE SHORTEST WORD IS “ME”

TEST CASE: “I AM THE LEGENDARY VILLAIN”

OUTPUT: **1** – BECAUSE THE SHORTEST WORD IS “I”

**MY ANSWER:**A white text box with black text

Description automatically generated

**“COUNT THE ISLANDS”**

Implement an algorithm which analyzes a two-color image and determines how many isolated areas of a single color the image contains.

Islands

An "island" is a set of adjacent pixels of one color (1) which is surrounded by pixels of a different color (0). Pixels are considered adjacent if their coordinates differ by no more than 1 on the X or Y axis.

Below you can see an example with 2 islands:

* on the left in the form of a matrix of 1's and 0's
* on the right in an equivalent stringified form using "X" and "~" characters for better readability

**SAMPLE:**Text

Description automatically generated

**TEST CASE**

**GIVEN a 4x4 matrix.**

**[ 1,1,1,1 ]  
[ 0,1,1,0 ]**

**[ 0,1,0,1 ]  
[ 1,1,0,0 ]**

**Output: “XXXX”  
 “~XX~”**

**“~X~X”**

**“XX~~”**

**MY ANSWER:**

A screenshot of a computer

Description automatically generated

**“WORD SEARCH”**

You are given a word target and a list of sorted (by length(increasing), number of upper-case letters(decreasing), natural order) unique words; words which always contain target, your task is to find the index (0 based) of target in words, which would always be in the list.

MAX ARRAY LENGTH: 1000;

TEST CASES:

[“I”,”TWO”,”FORTY”,”THREE”,’JEN’,”TWO”,”tWo”,”Two”];

TARGET **= “TWO”**

OUTPUT **= INDEX 1 and INDEX 5 // [1,5]**

**MY ANSWER:**

A screenshot of a computer

Description automatically generated

**MYSQL:**

**NOTE: EXCEL / CSV FILE WILL BE PROVIDED**

**Based on the data given:**

1. **Display total number of albums sold per artist**

[SELECT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/select.html) artists.name AS Artist, [COUNT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/group-by-functions.html#function_count)(albums.id) AS TotalAlbumsSold FROM artists [LEFT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/string-functions.html#function_left) JOIN albums ON artists.id = albums.artist\_id GROUP BY artists.name;

A screenshot of a computer

Description automatically generated

1. **Display combined album sales per artist**

[SELECT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/select.html) artists.name AS Artist, [SUM](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/group-by-functions.html#function_sum)(albums.sales) AS CombinedSales FROM artists [LEFT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/string-functions.html#function_left) JOIN albums ON artists.id = albums.artist\_id GROUP BY artists.name;

A screenshot of a computer

Description automatically generated

1. **Display the top 1 artist who sold most combined album sales**

[SELECT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/select.html) artists.name AS Artist, [SUM](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/group-by-functions.html#function_sum)(albums.sales) AS CombinedSales FROM artists [LEFT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/string-functions.html#function_left) JOIN albums ON artists.id = albums.artist\_id GROUP BY artists.name ORDER BY CombinedSales DESC LIMIT 1;

A close up of numbers

Description automatically generated

**4. Display the top 10 albums per year based on their number of sales**

A screenshot of a computer

Description automatically generated

**5. Display list of albums based on the searched artist**

[SELECT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/select.html) albums.name AS Album\_Name, albums.year AS Album\_Year, albums.sales AS Album\_Sales, albums.last\_update AS Album\_Last\_Update, albums.cover\_image AS Album\_Cover\_Image FROM albums JOIN artists ON albums.artist\_id = artists.id WHERE artists.name = 'Enhypen';

A screenshot of a computer

Description automatically generated

**LARAVEL**

**NOTE: SAME CSV FILE FROM MYSQL EXAM**

**MODELS**

**ARTISTS**

**FIELDS**

**CODE**

**NAME**

**FUNCTIONS**

**- read the full details of created artist**

**- update the full details of created artist**

**- delete the details of created artist**

**ALBUMS**

**FIELDS**

**YEAR**

**NAME**

**SALES**

**FUNCTIONS**

**- read the full details of album**

**- update**

**- delete**

**- add a picture of album cover**

**Migration and DB Relation**

**- use the csv file to populate the artist and albums and use "faker" for the other details that is not in the csv file**

**Login**

**- admin user can perform logout functionality**

**- credentials must be username and password**

**Dashboard**

**- Display total number of albums sold per artist**

**- Display combined album sales per artist**

**- Display the top 1 artist who sold most combined album sales**

**- Display list of albums based on the searched artist**

**Bonus:**

**- Translate each sql scripts to a Laravel eloquent format**

**- Provide routes for each scripts for each scenarios**

**- API: Output must be in json format**

**- Provide an authentication function that will grant a bearer token that can be used to access the said routes**

**Create API route under route.php in Laravel using the created controller on CRUD basic in above activity.**

**• Endpoint Can perform GET, POST, PUT/PATCH, DELETE for**

**• Artist**

**• Album**