

Name: \_\_\_\_\_

## Backend programming

1. Write a class named **test1** that will support the following methods:

1. **Static function last\_word(\$sentence)** – returns the length of the last word of a given sentence, or 0 for empty string.
2. **Static function extract\_string(\$str)** – returns a part of a string that is marked with square parenthesis, if exists. For example: for "The quick [brown fox]." It will return "brown fox". For "Hello World" it will return "".

2. Write a class named **test2** that will support the following methods:

1. **function is\_power(\$number,\$base)**– returns true if there is a number n so that  $\$base^{<n>} = \$number$ . For example `is_power(16, 4) = true`, `is_power(12, 3) = false`, `is_power(1,100) = true`
2. **function format\_number(\$str)** –returns a string after removing nonnumeric characters except comma and dot. For example: for "\$4,000.15A" it will return "4,000.15" .
3. **function sum\_digits(\$int)** –returns the sum of the digits of the given number. For example: `sum_digits(9123) = 15`, `sum_digits(-5555) = 20`

3. The below code is designed to fetch x,y coordinates of multiple locations, marked with a numeric *id*. The code works, but has some issues.

Which issues did you notice? How would you fix or improve it?

```
$data = [];
foreach ($ids as $id) {
    $result = $connection->query("SELECT 'x', 'y' FROM 'values' WHERE 'id' = " . $id);
    $data[] = $result->fetch_row();
}
```

# MySQL programming

4. Assume we have those two tables,

Table 'a'		Table 'b'	
Id	Name	Id	Grade
1	Eli	1	98
2	Moshe	3	55
3	Yossi	4	100

What will be the output of this query?

```
SELECT a.id, a.name, b.grade
FROM a
LEFT JOIN b on b.id = a.id
```

- 3 rows, 1 NULL value
- 2 rows, no NULL values
- 3 rows, 2 NULL values
- 9 rows, no NULL values

5. Assume we have those two tables:

Table salesman			
Id	Name	city	commission
5001	James Hoog	New York	0.15
5002	Nail Knite	Paris	0.13
5005	Pit Alex	London	0.11
5006	Mc Lyon	Paris	0.14
5003	Lauson Hen	Berlin	0.12
5007	Paul Adam	Rome	0.13

Table customer			
Id	Name	city	Salesman_id
3002	Nick Rimando	New York	5001
3005	Graham Zusi	California	5002
3001	Brad Guzan	London	5005
3004	Fabian Johns	Paris	5006
3007	Brad Davis	New York	5001
3009	Geoff Camero	Berlin	5003
3008	Julian Green	London	5002

- i. Write an SQL statement to get all customers' name, city and salesmans' names, which the salesman's commission is between 12% and 14%.

- ii. Write an SQL statement to get all salesmen that didn't sell anything.

6. Consider the following table definition in a MySQL database:

```
CREATE TABLE example (  
  id INT NOT NULL,  
  text1 VARCHAR(32) NOT NULL,  
  text2 VARCHAR(32) NOT NULL DEFAULT 'foo'  
);
```

Which of the following insert queries will success and which will fail? Explain your answer.

- a. INSERT INTO example (id,text2) VALUES(1,'test');
- b. INSERT INTO example (text1,id) VALUES('test',1);
- c. INSERT INTO example (text1,text2) VALUES('test','test');
- d. INSERT INTO example (id, text1,text2) VALUES(1,'test','test');

## Frontend programming

7. Assume you have an array filled with numbers, which has duplicate values inside. Remove duplicates of an array and return an array of only unique elements. Please explain your answer shortly.

For example:

```
var arr = [1, 2, 3, 5, 1, 5, 9, 1, 2, 8];  
function unique (arr){...};  
unique(arr); // [1, 2, 3, 5, 9, 8]
```

Note: you can use either ES5 or ES6

8. Implement the following function (Please explain your answer shortly):

```
function duplicate (arr, times){...};  
duplicate([1, 2, 3, 4, 5],3); // [1,2,3,4,5,1,2,3,4,5,1,2,3,4,5]
```

Note: you can use either ES5 or ES

9. The code below is for an application that requires defining a click handler for all buttons on the page, including those buttons that may be added later dynamically.  
When clicking on the appended button (id='btn\_\${res.id}'), nothing happens. Why? How can we fix it

```

$("button.clickable").on("click", function() {

    console.log("Button Clicked:", this);

}); // defines the click handler for all buttons

$.get({...}).success(function(res) {

    $("body").append("<button id=`btn_${res.id}` class=`clickable`>Click  
Alert!</button>");

} // dynamically add another button to the page

```

10. Assume you have a button with the following design. Using JQuery, write code that will increase by 1 the number in the badge each time the user clicks on that button. The value inside the .badge in the button is pre-initialized randomly in document.ready.

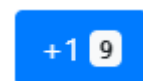
**Bonus: Make this code support multiple notification buttons.**

**Example:**

```

<button type="button" class="btn btn-primary">
  +1 <span class="badge badge-light">9</span>
</button>

```



## Full-stack programming

In this task, you will need to write a short page that will help a user to calculate Fibonacci sequence number, but will also display his input history of all users.

Design a page, using jQuery, ajax, bootstrap and [datatables](#) that will allow users to insert a username and a number, click “submit”, and it will display the result of a Fibonacci sequence with the position of that number. You should validate the input in both frontend and backend parts.

The calculation should be done and saved in server side, so the response should be a json with the data, including the user’s IP address.

On first load, the page will get the list of all users that have previously submitted numbers and usernames. The table should have pagination to prevent long loadings.

On the backend side, the data can be saved in any format, to make it the easiest to use (file, sqlite etc).

The page should show the history of the users’ requests in a table similar to this:

Username	User’s Input	Fibonacci Number
Jhon	0	0
Max	1	1
Alice	10	55

Design the frontend and the backend of this page. You should design it to be beautiful, intuitive and easy to use and understand.

Recommendations:

1. Reloading the page should be avoided as much as possible.
2. The table can grow big. it should be loaded after the page is loaded in a async way
3. The design of the frontend is more important than the design of the backend php part. It should be designed creatively for extra points.
4. You can add any extra features that you think that are relevant to this page.

