

Problem 1. Quoted text

Create a string variable with quoted text in it. For example: He said, "I don't care."

Note: To show the string, you can use one of the following methods:

```
document.writeln(string);
```

```
console.log(string);
```

Create a p tag with ID, `document.getElementById("p_tag_ID").innerHTML = string;`

Problem 2. Parsing numbers

Try parsing the following strings to numbers using `parseInt`, `parseFloat`, `Number`, `+` and `| 0`.

Fill the answers for yourself in the table below.

[illegible]

Problem 3. Compare Array of Objects

Description

Write a function to sort an array of product objects in increasing order of product's price.

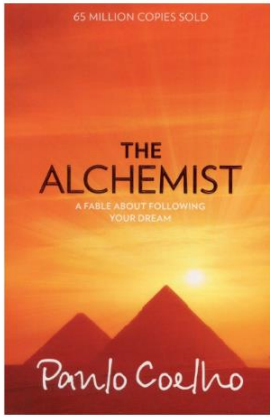
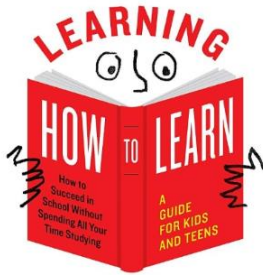
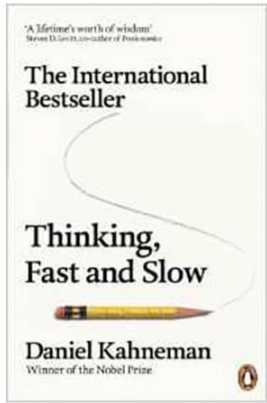
Input

One array contains product objects

Each of them has three properties: Product Name, Product Image and Product Price

Output

Return one array that contains the product objects in increasing order of their price

<p>The Alchemist</p>  <p>182.700 VND</p>	<p>Learning How to Learn</p> <p><small>Copyrighted Material</small></p> <p><small>From the bestselling author of <i>A Mind for Numbers</i> and the creator of the popular online course <i>Learning How to Learn</i></small></p>  <p><small>BARBARA OAKLEY, PhD, AND TERRENCE SEJNOWSKI, PhD, WITH ALISTAIR MCCONVILLE</small></p> <p>251.100 VND</p>	<p>Thinking, Fast and Slow</p> <p><small>'A lifetime's worth of wisdom' Steven D. Levitt, co-author of <i>Freaky Economics</i></small></p> <p>The International Bestseller</p>  <p>299.700 VND</p>
---	--	--

Problem 4. Product List

Create an array that contains a product list. Each product consists of ID, Product Name, Quantity, Price, Image

- Sort the product list by the product name
- Print the products in the list
- Add a new product to head of the list
- Add a new product to tail of the list
- Search products by the product name
- Remove the product by the product ID
- Edit the product detail by the product ID

```
Initial Products:
ID: 1, Name: Laptop, Quantity: 5, Price: 1200, Image:
laptop_image.jpg
ID: 2, Name: Phone, Quantity: 10, Price: 800, Image:
phone_image.jpg
ID: 3, Name: Tablet, Quantity: 8, Price: 500, Image:
tablet_image.jpg

Search Result:
ID: 3, Name: Tablet, Quantity: 8, Price: 500, Image:
tablet_image.jpg

Updated Products:
ID: 4, Name: Smartwatch, Quantity: 15, Price: 200, Image:
watch_image.jpg
ID: 1, Name: Laptop, Quantity: 5, Price: 1200, Image:
laptop_image.jpg
ID: 3, Name: Updated Tablet, Quantity: 10, Price: 550,
Image: updated_tablet_image.jpg
ID: 5, Name: Camera, Quantity: 3, Price: 600, Image:
camera_image.jpg
```

Problem 5. Numbers from 1 to N

Description

Implement a JavaScript function that accepts a positive 32-bit integer N number and prints all the numbers from 1 to N inclusive, on a single line, separated by a whitespace.

Input

The input will consist of a positive 32-bit integer N number.

Output

The output should consist of a single line - the numbers from 1 to N, separated by a whitespace.

Numbers from 1 to N	Numbers from 1 to N
Enter a positive integer N: <input type="text" value="12"/> <input type="button" value="Print Numbers"/>	Enter a positive integer N: <input type="text"/> <input type="button" value="Print Numbers"/>
1 2 3 4 5 6 7 8 9 10 11 12	Please enter a valid positive integer.

Problem 6. Prime check

Description

Implement a JavaScript function that accepts an integer N number (N will always be a valid 32-bit integer number) and uses an expression to check if given N is prime (i.e., it is divisible without remainder only to itself and 1).

Note

You should check if the number is positive.

Input

The only element will be the integer N number

Output

Output true if the number is prime and false otherwise.

You can use console.log to print the results or you can use return to return the answer. Both are correct.

Prime Check

Enter a positive integer N:

Please enter a valid positive integer.

Prime Check

Enter a positive integer N:

true

Prime Check

Enter a positive integer N:

false

Problem 7. MMSA of N Numbers

Description

Implement a JavaScript function that accepts an array of floating-point numbers and returns the minimal, the maximal number, the sum and the average of all numbers (displayed with 2 digits after the decimal point).

Input: The array that will be passed as a parameter to your function will contain the numbers of the sequence.

Output: The output must always consist of exactly 4 lines like the following format:

Min = 3.00

Max = 6.00

Sum = 9.00

Avg = 4.50

Constraints

$1 \leq N \leq 100.000.000$

All numbers will be valid floating-point numbers that will be in the range [0, 100.000.000]

MMSA of N Numbers

Enter numbers separated by commas:

Min = 13.00

Max = 20.00

Sum = 65.00

Avg = 16.25