# Lab: Advanced Functions

Problems for in-class lab for the [JavaScript Advanced” course @ SoftUni](https://softuni.bg/courses/javascript-advanced). Submit your solutions in the SoftUni judge system at <https://judge.softuni.bg/Contests/1528>.

## Aggregates

Write a program that uses a **reducer** function to **display** information about an **input array**.

### Input

You will receive an **array** of **numeric** values.

### Output

The output should be the **printed** on the console. Display the **sum** of all elements in the array, the value of the **smallest**, the value of the **biggest**, the **product** of all elements and a string of all elements **concatenated**.

### Examples

|  |  |
| --- | --- |
| Sample Input | Output |
| [2, 3, 10, 5] | Sum = 20  Min = 2  Max = 10  Product = 300  Join = 23105 |
| [5, -3, 20, 7, 0.5] | Sum = 29.5  Min = -3  Max = 20  Sum = -1050  Join = 5-32070.5 |

## Currency Format

Write a higher-order function that fixes some of the parameters of another function. Your program will receive a function that **takes 4 parameters** and **returns a formatted string** (a monetary value with currency symbol).

Your task is to **return another function** that only **takes one parameter** and **returns** the **same formatted string**.

You will receive the following function:

|  |
| --- |
| currencyFormatter |
| function currencyFormatter(separator, symbol, symbolFirst, value) {  let result = Math.trunc(value) + separator;  result += value.toFixed(2).substr(-2,2);  if (symbolFirst) return symbol + ' ' + result;  else return result + ' ' + symbol;  } |

Set the following parameters to fixed values:

separator: ","

symbol: "$"

symbolFirst: true

The final parameter value is the one that the returned function will receive.

### Input

You will receive a function parameter

### Output

You need to **return a function** that takes one parameter - value

### Examples

|  |
| --- |
| Sample Input |
| let dollarFormatter = result(currencyFormatter);  console.log(dollarFormatter(5345)); *// $ 5345,00*  console.log(dollarFormatter(3.1429)); *// $ 3,14*  console.log(dollarFormatter(2.709)); *// $ 2,71* |

## Command Processor

Write a program that keeps a string **inside its context** and can execute different **commands** that modify or print the string on the console.

append(string) - append the given **string** at the end of the internal string

removeStart(n) - **remove** the **first** **n** characters from the string, n is an integer

removeEnd(n) - **remove** the **last n** characters from the string, n is an integer

print - **print** the stored string on the **console**

### Input

You will receive an **array** of **strings**. Each element is a command that may be a single word or contain an argument, separated by space.

### Output

Whenever you receive the command print, the output should be the **printed** on the console.

### Examples

|  |  |
| --- | --- |
| Sample Input | Output |
| ['append hello',  'append again',  'removeStart 3',  'removeEnd 4',  'print'] | loa |
| ['append 123',  'append 45',  'removeStart 2',  'removeEnd 1',  'print'] | 34 |

## Max Element

Write a program that takes an **array** of **numeric elements** as an input and **returns** the **largest** element of the array.

### Input

You will receive an **array** of **numbers**.

### Output

The **output** should be the **return** value of your function. It represents the **largest element** of the array.

### Examples

|  |  |
| --- | --- |
| Sample Input | Output |
| [10, 20, 5] | 20 |
| [1, 44, 123, 33] | 123 |

## Next Article

Write a JS program that sequentially **displays articles** on a web page when the user **clicks** a button. You will receive an **array of strings** that will initialize the program. You need to **return** a function that **keeps the initial array** in its **closure** and every time it is called, it **takes the first element** from the array and **displays it** on the web page, inside a div with ID "content". If there are no more elements left, your function shouldn’t do anything.

Your function will be called automatically, there is **no need** to attach any event listeners.

### Input

You will receive and **array** of strings.

### Output

Return a **function** that displays the array elements on the web page.

### Examples





