**Lab: Arrays Advanced**

## Sum First Last

Write a function that calculates and prints the **sum** of the **first** and the **last** elements in an array.

The **input** comes as array of string elements holding numbers.

The **output** is the return value of your function.

### Examples

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Input** | **Output** |  | **Input** | **Output** |
| ['20', '30', '40'] | 60 |  | ['5', '10'] | 15 |

let numbers=function (arr){  
  
 let result;  
  
 result=arr[0]+arr[2];  
  
 return result;  
  
  
};  
  
  
  
***console***.log(numbers([20,30,40]));

let ***array***=[5,10,20,40,32];  
  
  
 let ***number***=***array***.slice(2,4);  
  
 ***console***.log(***array***);  
  
 ***console***.log(***number***);  
  
 let ***result***=***number***[0]+***number***[1];  
 ***console***.log(***result***);

**2Negative / Positive Numbers**

Write a function that processes the elements in an **array** one by one and produces a **new** array. **Prepend** each **negative** element at the front of the result and **append** each **positive** (or **0**) element at the end of the result.

The **input** comes as array of number elements.

The **output** is printed on the console, each element on a new line.

**Examples**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Input** | **Output** |  | **Input** | **Output** |
| [7, -2, 8, 9] | -2  7  8  9 | [3, -2, 0, -1] | -1  -2  3  0 |

**Hints**

* Use **unshift()** to add an element at the **first** position
* Use **push()** to add an element at the **last** position

function solve(arr){  
  
 let compare=(a,b)=>{  
 return a-b;  
 }  
  
 let result=arr.sort(compare);  
  
 ***console***.log(result);  
  
  
}solve([7, -2, 8, 9]);

1. **First and Last K Numbers**

Write a function that prints the first **k** and the last **k** elements from an **array of numbers**.

The **input** comes as **array of number** elements. The first element represents the number **k**, all other elements are from the array that needs to be processed.

The **output** is printed on the console on two lines. On the first line print the **first** **k** elements, separated by space. On the second line print the **last** **k** elements, separated by space.

**Examples**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Input** | **Output** |  | **Input** | **Output** |
| [**2**,  7, 8, 9] | 7 8  8 9 | [**3**,  6, 7, 8, 9] | 6 7 8  7 8 9 |

### Hints

* Use **slice()** to split the array into two parts

function solve(array){  
   
 let k=array.shift();  
   
  
 let firstElement=array.slice(0,2);  
  
 ***console***.log(firstElement);  
  
 let lastElement=array.slice(array.length-2);  
  
 ***console***.log(lastElement);  
  
 let result='';  
 let resultTwo='';  
  
  
 for(let i=0;i<k;i++){  
 result+= ` ${firstElement[i]}`;  
 resultTwo+=` ${lastElement[i]}`  
 }  
  
 ***console***.log(result);  
 ***console***.log(resultTwo);  
  
  
}solve([2, 7, 8, 9]);

function solve(input){  
  
  
 let element=input.shift();  
 ***console***.log(input);  
  
 let firstElement=input.slice(0,3);  
  
 ***console***.log(firstElement);  
  
 let lastElement=input.slice(1,4);  
 ***console***.log(lastElement);  
  
  
 let resultOne='';  
 let resultTwo='';  
  
  
 for(let i=0;i<element;i++){  
 resultOne+=` ${firstElement[i]}`;  
 resultTwo+=` ${lastElement[i]}`;  
 }  
  
  
 ***console***.log(`First array ${resultOne}`);  
  
 ***console***.log(`Second array ${resultTwo}`);  
  
  
}solve([3, 6, 7, 8, 9]);

1. **Last K Numbers Sequence**

You are given two integers **n** and **k**. Write a function that generates and prints the following sequence:

* The first element is **1.**
* Every following element equals the sum of the previous **k** elements.
* The length of the sequence is **n** elements.

The **input** comes as two number arguments. The first element represents the number **n**, and the second – the number **k**.

The **output** is printed on the console on a single line, separated by space.

**Examples**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Input** | **Output** |  | **Input** | **Output** |
| 6, 3 | 1 1 2 4 7 13 | 8, 2 | 1 1 2 3 5 8 13 21 |

### Hints

The 2nd element (1) is the sum of the 3 elements before it, but there is only 1, so we take that. The third element, is the sum of the first 2 (1 and 1) and the 4th – the sum of 1, 1 and 2. The 5th element is the sum of the 2nd, 3rd and 4th (1, 2 and 4) and so on.

1. **Process Odd Numbers**

You are given an **array of numbers**. Write a function that prints the elements at **odd positions** from the array, **doubled** and in **reverse** order.

The **input** comes as array of number elements.

The **output** is printed on the console on a single line, separated by space.

**Examples**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Input** | **Output** |  | **Input** | **Output** |
| [10, 15, 20, 25] | 50 30 | [3, 0, 10, 4, 7, 3] | 6 8 0 |

### Hints

* Counting in arrays starts from 0
* For example –we receive 10, 15, 20, 25
* The elements at odd positions are 15 (index 1) and 25 (index 3)
* We need to take these two elements and multiply them \* 2
* Finally, we print them on the console in reversed order
* function solve(array){  
    
   let odd=(x)=>{  
   return x%2!==0;  
   }  
    
    
   let oddNumber=array.filter(odd);  
    
   ***console***.log(` Odd numbers:${oddNumber}`);  
    
   let result=oddNumber.map((x)=>x\*2)  
    
   ***console***.log(` \*2 ${result}`);  
    
    
  }solve([10, 15, 20, 25]);
* function solve(array){  
    
    
   ***console***.log(array);  
    
   let evenNumber=array.filter((x)=>x%2===0);  
    
   ***console***.log(evenNumber);  
    
   let result=evenNumber.map((x)=>x\*2);  
    
   ***console***.log(result);  
     
    
  }solve([3, 0, 10, 4, 7, 3]);

1. **Smallest Two Numbers**

Write a function that prints the **two smallest** elements from an **array of numbers**.

The **input** comes as array of number elements.

The **output** is printed on the console on a single line, separated by space.

**Examples**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Input** | **Output** |  | **Input** | **Output** |
| [30, 15, 50, 5] | 5 15 | [3, 0, 10, 4, 7, 3] | 0 3 |

### Hints

* You can use the following function to sort the numbers in the array:



* Afterwards the **first two** elements in the array are the **smallest**
* You can use **slice()** to take the first two numbers

function solve(array){  
  
 let small=function off(a,b){  
  
 return a-b;  
 }  
  
 let sorted=array.sort(small);  
  
 ***console***.log(sorted);  
  
 let twoSmallest=sorted.slice(0,2);  
 ***console***.log(twoSmallest);  
  
  
}solve([30, 15, 50, 5])

function solve(array){  
  
 let small=function(a,b){  
 return a-b;  
 }  
  
 let smallest=array.sort(small);  
  
 ***console***.log(smallest);  
  
 let result=smallest.slice(0,2);  
  
 ***console***.log(result);  
  
  
 let multiply=result.map((x)=>x\*2);  
  
 ***console***.log(multiply);  
  
  
}solve([3, 0, 10, 4, 7, 3]);

1. **List of Products**

You will receive an **array of products**. Print a **numbered array** of all the products **ordered by name**.

**Example**

|  |  |
| --- | --- |
| **Input** | **Output** |
|  | 1.Apples  2.Onions  3.Potatoes  4.Tomatoes |

**Hints**

* The **sort function** rearranges the array in ascending order



* Finally, we have to **print our sorted** array. To do that we **loop through the array**



* We use **i + 1**, because we want to **start counting from 1**

function solve(array){  
  
 let result=array.sort();  
  
 for(let i=0;i<result.length;i++){  
  
 ***console***.log(`${i+1}.${result[i]} `);  
 }  
  
  
}solve(["Potatoes", "Tomatoes", "Onions", "Apples"]);

1. **Array Manipulations**

Write a function that manipulates an **array of numbers**.

* **Add {number}:** add a number to the **end** of the array
* **Remove {number}:** remove number from the array
* **RemoveAt {index}:** removes number at a **given index**
* **Insert {number} {index}:** inserts a number at a **given index**

**Note: All the indices will be valid!**

Print the final state of the array (**separated by single space**).

The **input** comes as **array of strings**. First element will be a string containing the **array to manipulate**. Every other **command** you receive will also be a string.

The **output** is the manipulated array printed on the console on a single line, **separated by space**.

**Example**

|  |  |
| --- | --- |
| **Input** | **Output** |
| ['4 19 2 53 6 43',  'Add 3',  'Remove 2',  'RemoveAt 1',  'Insert 8 3'] | 4 53 6 8 43 3 |

**Hints**

First we receive the whole input:



* After that we take the **first** element from the commands and **convert** it to an **array of numbers**:



* Then we loop through the commands array and obtain **each element** from the command and cast both numbers. This event is called [destructuring](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Operators/Destructuring_assignment):



* We check if the command is equal to one of the given: "**Add**", "**Remove**", etc. 
* To add element at the end, use **push()**



* To remove **all occurrences** of a particular element from the array, you can use **filter()**



* To remove or insert at an index, you can use **splice()**



**Note:** Removing elements with **splice()** receives two parameters:

* Start Index
* Count of elements you want to remove

**Note:** Inserting elements with **splice()** receives three parameters:

* Start Index
* Count of elements to remove – if none enter 0
* Elements to insert at that position
* function solve(commands) {  
   let arr = commands.shift().split(' ').map(***Number***);  
   for (let i = 0; i < commands.length; i++) {  
   let [command, firstNum, secondNum] = commands[i].split(' ');  
   firstNum = ***Number***(firstNum);  
   secondNum = ***Number***(secondNum);  
   switch (command) {  
   case 'Add':  
   add(firstNum);  
   break;  
   case 'Remove':  
   remove(firstNum);  
   break;  
   case 'RemoveAt':  
   removeAt(firstNum);  
   break;  
   case 'Insert':  
   insert(firstNum, secondNum);  
   break;  
   }  
   }  
   function add(el) {  
   arr.push(el);  
   }  
   function remove(num) {  
   arr = arr.filter(el => el !== num);  
   }  
   function removeAt(index) {  
   arr.splice(index, 1);  
   }  
   function insert(num, index) {  
   arr.splice(index, 0, num);  
   }  
   ***console***.log(arr.join(' '));  
  }  
  solve(['4 19 2 53 6 43',  
   'Add 3',  
   'Remove 2',  
   'RemoveAt 1',  
   'Insert 8 3']  
  );

function solve(input){  
  
 let first=input.slice();  
 let array=first[0].split(' ').map(makeMeNumber);  
  
  
  
 function makeMeNumber(element){  
 return ***Number***(element);  
 }  
  
 ***console***.log(array);  
  
 let addNumber=function(arr){  
 array.push(3);  
 return arr;  
  
 }  
  
 ***console***.log(addNumber(array));  
  
 let remove=function (arr){  
  
 array.splice(2,2,53);  
 return arr;  
 }  
 ***console***.log(remove(array));  
  
 let removeAt=function(arr){  
 array.splice(1,1);  
 return arr  
 }  
 ***console***.log(removeAt(array));  
  
  
 let insert=function (arr){  
  
 array.splice(3,4,8,43,3);  
 return arr;  
  
 }  
  
 ***console***.log(`Result is :`);  
  
 ***console***.log(insert(array));  
  
  
}solve(['4 19 2 53 6 43',  
 'Add 3',  
 'Remove 2',  
 'RemoveAt 1',  
 'Insert 8 3']  
);

function solve(array){  
  
 let firstElement=array.slice();  
  
 let first=firstElement[0].split(' ').map(makeMeNumber);  
  
 function makeMeNumber(el){  
 return ***Number***(el);  
 }  
  
 ***console***.log(first);  
  
  
}solve(['4 19 2 53 6 43',  
 'Add 3',  
 'Remove 2',  
 'RemoveAt 1',  
 'Insert 8 3']  
);