# Exercises: JavaScript Basics and DOM

Problems for exercises and homework for the [“Software Technologies” course @ SoftUni](https://softuni.bg/courses/software-technologies).

## Multiply a Number by 2

You are given a number **N**. Create a program that **multiplies** that **number by 2** and prints the result.

### Examples

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Input** | **Output** |  | **Input** | **Output** |  | **Input** | **Output** |  | **Input** | **Output** |
| 2 | 4 |  | 3 | 6 | 30 | 60 | 13 | 26 |

## Multiply a Number by X

You are given a number N and a number X. Create a program that multiplies N \* X and prints the result.

### Examples

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Input** | **Output** |  | **Input** | **Output** |  | **Input** | **Output** |  | **Input** | **Output** |
| 2  3 | 6 |  | 13  13 | 169 | 1  2 | 2 | 0  50 | 0 |

## Multiply / Divide a Number by a given second number

You are given a number N and a number X. Create a program that multiplies N \* X if X is greater than or equal to N and divides N / X if N is greater than X.

### Examples

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Input** | **Output** |  | **Input** | **Output** |  | **Input** | **Output** |  | **Input** | **Output** |
| 2  3 | 6 |  | 13  13 | 169 | 3  2 | 1.5 | 144  12 | 12 |

## Product of 3 numbers

You are given a number X, Y and Z. Create a program that finds if X \* Y \* Z (the product) is negative or positive. Try to do this **WITHOUT** multiplying the 3 numbers.

### Examples

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Input** | **Output** |  | **Input** | **Output** |  | **Input** | **Output** |
| 2  3  -1 | Negative |  | 5  4  3 | Positive | -3  -4  5 | Positive |

## Print Numbers from 1 to N

You are given a number N. Create a program that loops through all of the numbers from **1 to N** and prints them. N will always be positive.

### Examples

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Input** | **Output** |  | **Input** | **Output** |
| 5 | 1  2  3  4  5 |  | 2 | 1  2 |

## Print Numbers from N to 1

You are given a number N. Create a program that loops through all of the numbers from **1 to N** and prints them. N will always be positive.

### Examples

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Input** | **Output** |  | **Input** | **Output** |
| 5 | 1  2  3  4  5 |  | 2 | 1  2 |

## Print lines

You will be, continuously, given input lines, until you receive the command “Stop”. Print each of those lines at the moment you read them, until you reach the ending command. Do **NOT** print the ending command.

### Examples

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Input** | **Output** |  | **Input** | **Output** |
| Line 1  Line 2  Line 3  Stop | Line 1  Line 2  Line 3 |  | 3  6  5  4  Stop  10  12 | 3  6  5  4 |

## Print numbers in reversed order

You will be given random numbers as input. You need to print them backwards, compared to the order they were received, each on a new line.

### Examples

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Input** | **Output** |  | **Input** | **Output** |  | **Input** | **Output** |
| 10  15  20 | 20  15  10 |  | 5  5.5  24  -3 | -3  24  5.5  5 | 20  1  20  1  20 | 20  1  20  1  20 |

## Set values to indexes in an array

You will be given **N** –an integer specifying the length of an array. Then you will start receiving an index and a value. For each received line you must set the value at the given index to the given value. When you’ve processed all input data, print the array’s elements each on a new line.

### Examples

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Input** | **Output** |  | **Input** | **Output** |  | **Input** | **Output** |
| 3  0 - 5  1 – 6  2 – 7 | 5  6  7 |  | 2  0 – 5  0 – 6  0 – 7 | 7  0 | 5  0 – 3  3 - -1  4 – 2 | 3  0  0  -1  2 |

## Add / Remove elements

You will be given input lines containing 2 elements separated by a space. You can receive **add** command which’s second element is a number. You need to add that number to an array. You can also receive **remove** command and an index. You must remove the element at that index. If the index is nonexistent, ignore that input line. When you process all input data, print the array’s elements each on a new line.

### Examples

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Input** | **Output** |  | **Input** | **Output** |  | **Input** | **Output** |
| add 3  add 5  add 7 | 3  5  7 |  | add 3  add 5  remove 1  add 2 | 3  7 | add 3  add 5  remove 2  remove 0  add 7 | 5  7 |

## Working with key-value pairs

You will be given input lines containing 2 elements separated by a space. The first is the key and the second is the value. If a key is already entered you need to replace the old value with the new one. At the last line of input you will receive a key. You must print the value corresponding to that key. If there is no such, just print **None**.

### Examples

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Input** | **Output** |  | **Input** | **Output** |  | **Input** | **Output** |
| key value  key eulav  test tset  key | eulav |  | 3 test  3 test1  4 test2  4 test3  4 test5  4 | test5 | 3 bla  3 alb  2 | None |

## Multiple values on one key

You will be given input lines containing 2 elements separated by a space. The first is the key and the second is the value. You need to store the given values to the given keys. At the last line of input you will receive a key. You must print all the values corresponding to that key. If there are no such, just print **None**.

### Examples

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Input** | **Output** |  | **Input** | **Output** |  | **Input** | **Output** |
| key value  key eulav  test tset  key | value  eulav |  | 3 test  3 test1  4 test2  4 test3  4 test5  4 | test2  test3  test5 | 3 bla  3 alb  2 | None |

## Storing objects

You will be given input lines containing information about students – name, age and grade. Extract that information from the input lines, and create objects to hold the data. Then print the data exactly in the format specified below.

### Examples

|  |  |
| --- | --- |
| **Input** | **Output** |
| Pesho -> 13 -> 6.00  Ivan -> 12 -> 5.57  Toni -> 13 -> 4.90 | Name: Pesho  Age: 13  Grade: 6.00  Name: Ivan  Age: 12  Grade: 5.57  Name: Toni  Age: 13  Grade: 4.90 |

## Parse JSON objects

You will be given input lines containing object data in JSON format. Use the JSON.parse function to parse the data into JavaScript objects, and then print it.

### Examples

|  |  |
| --- | --- |
| **Input** | **Output** |
| “{“name”:”Gosho”,”age”:”10”,”date”:”19/06/2005”}”  “{“name”:”Tosho”,”age”:”11”,”date”:”04/04/2005”}” | Name: Gosho  Age: 10  Date: 19/06/2005  Name: Tosho  Age: 11  Date: 04/04/2005 |

## Turn object into string

You will be given input lines containing information about an object’s members. Add to that object the given member with its value. After you’ve processed all of the input data, print the stringified version of the object. Use the JSON.stringify function.

### Examples

|  |  |
| --- | --- |
| **Input** | **Output** |
| name -> Angel  surname -> Georgiev  age -> 20  grade -> 6.00  date -> 23/05/1995  town -> Sofia | {"name":"Angel","surname":"Georgiev","age":20,"grade":6,"date":"19/05/1995","town":"Sofia"} |