

Exercise: JavaScript Syntax and Operators

Problems for exercises and homework for the ["JavaScript Fundamentals" course @ SoftUni](#)

Submit your solutions in the SoftUni judge system at
<https://judge.softuni.bg/Contests/Practice/Index/1422>

1. I like JavaScript!

Write a JS function that **can receive a name** as input and print a greeting to the console.

The **input** comes as a single string that is the name of the person.

The **output** should be printed to the console.

Example

Input	Output
'George'	Hello George, do you like JavaScript?
'Maria'	Hello Maria, do you like JavaScript?

2. Even Numbers 1 to N

Write a JS function that reads an integer **n** and prints all **even numbers** from **1** to **n**.

The **input** comes as a **single number n**. The number **n** will be an integer in the range [1 ... 100 000].

The **output** should hold the expected even numbers, each at a separate line.

Example

Input	Output
5	2 4

Input	Output
4	2 4

Input	Output
7	2 4 6

3. Fruit

Write a JS function that calculates how much money you need to buy a fruit. You will receive a **string** for the type of fruit you want to buy, a **number** for weight in grams and another **number** for a price per kilogram.

Print the following text on the console: **'I need {money} leva to buy {weight} kilograms {fruit}.'** . Print the weight and the money **rounded** to two decimal places.

The **input** comes as **three arguments** passed to your function.

The **output** should be printed to the console.

Example

Input	Output
'orange', 2500, 1.80	I need 4.50 leva to buy 2.50 kilograms orange.

Input	Output
'apple', 1563, 2.35	I need 3.67 leva to buy 1.56 kilograms apple.

4. Fitness Rates

Write a JS function that calculates how much money you need to visit your favorite gym. You will receive **two strings** for a day of week and a service which you want to use and a **number** for the time in which you can go to the gym.

In the table below you can find information about the prices and services offered at the gym.

Service	Fitness		Sauna		Instructor	
Hours	8.00-15.00	15.00-22.00	8.00-15.00	15.00-22.00	8.00-15.00	15.00-22.00
Monday	5.00	7.50	4.00	6.50	10.00	12.50
Tuesday	5.00	7.50	4.00	6.50	10.00	12.50
Wednesday	5.00	7.50	4.00	6.50	10.00	12.50
Thursday	5.00	7.50	4.00	6.50	10.00	12.50
Friday	5.00	7.50	4.00	6.50	10.00	12.50
Saturday	8.00	8.00	7.00	7.00	15.00	15.00
Sunday	8.00	8.00	7.00	7.00	15.00	15.00

Example: If you want to go to the gym on Monday at 15 o'clock and use the sauna you have to pay 6.50 leva.

The **input** comes as **three arguments** passed to your function.

The **output** should be printed to the console.

Example

Input	Output
'Monday', 'Sauna', 15.30	6.5

Input	Output
'Sunday', 'Fitness', 22.00	8

5. Greatest Common Divisor – GCD

Write a JS function that takes **two positive numbers** as input and compute the greatest common divisor.

Print on the console the result.

The **input** comes as **two positive integer numbers**.

The **output** should be printed to the console.

Example

Input	Output
15, 5	5

Input	Output
2154, 458	2

6. Same Numbers

Write a JS function that takes **an integer number** as input and check if all the digits in a given number are the same or not.

Print on the console **true** if all numbers are same or **false** if not. On the next line print the **sum of all the digits**.

The **input** comes as an integer number.

The **output** should be printed to the console.

Examples

Input	Output
2222222	true 14

Input	Output
1234	false 10

7. Time to Walk

Write a JS function that **calculates** how long it takes a student to get to the university.

The function takes **three numbers**:

- The **first** is the number of **steps** the student makes from his home to the university
- The **second** number is the length of the student's footprint in **meters**
- The **third** number is the student speed in **km/h**

Sometimes the student needs a rest. Every 500 meters, the person makes a **1 minute break**.

Calculate how long the student goes from home to university and print on the console the result as follows: **'hours:minutes:seconds'**.

The **input** comes as **three numbers**.

The **output** should be printed to the console.

Example

Input	Output
4000, 0.60, 5	00:32:48

Input	Output
2564, 0.70, 5.5	00:22:35

8. Flight Timetable

Write a JS function that displays flight information.

The **input** comes as an **array of string elements**.

- The first string can be **'Arrivals'** or **'Departures'**
- The second string is the **name** of the town
- The third is the **time** when the plain departures or arrives
- The fourth is the **flight number**
- The last one is the **gate number**

The **output** should be printed to the console in the following format:

"Departures/Arrivals: Destination - {town}, Flight - {flight number}, Time - {departure/arrival time}, Gate - {gate number}"

Example

Input	Output
['Departures', 'London', '22:45', 'BR117', '42']	Departures: Destination - London, Flight - BR117, Time - 22:45, Gate - 42
['Arrivals', 'Paris', '02:22', 'VD17', '3']	Arrivals: Destination - Paris, Flight - VD17, Time - 02:22, Gate - 3

9. Calorie Object

Write a JS function that composes an object by given properties. Every even index of the array is the string and the name of the food. Every odd index is a number that is equal to the calories in 100 grams of product. Assign each value to its respective property of an object and print it on the console.

The **input** comes as an **array of string elements**.

The **output** should be printed to the console.

Examples

Input	Output
['Yoghurt', 48, 'Rise', 138, 'Apple', 52]	{ Yoghurt: 48, Rise: 138, Apple: 52 }
['Potato', 93, 'Skyr', 63, 'Cucumber', 18, 'Milk', 42]	{ Potato: 93, Skyr: 63, Cucumber: 18, Milk: 42 }

10. *Coffee Machine

Write a program for a coffee machine. Calculate whether the money inserted in the machine is enough to make the order and print the corresponding output.

Input

The input is an **array of strings**. Each string represents one order with different parts, separated by a single space ' '.

- The **first part** is the **coins inserted**.
- The **second** is the **type of drink (coffee or tea)**.
- Next, if the drink type is **coffee**, you will receive **'caffeine'** or **'decaf'**.
- Next, you may receive **'milk'**, if the ordered drink is with milk. **It costs 10% of the drink price, rounded to first decimal point**
- And last you receive the **quantity of sugar, between 0 and 5. No matter the quantity (except from 0) it costs 0.10. Add the sugar at the end!**

The **prices of drinks** are:

Type	Price
coffee caffeine	0.80
coffee decaf	0.90
tea	0.80

Constraints

- The input will always be **valid**.

Output

For each order there are **two possible** outputs:

- If the money inserted is enough, calculate the change of the order:
'You ordered {drink}. Price: {price}\$ Change: {change}\$'
- If the money is not enough:
'Not enough money for {drink}. Need {moneyNeeded}\$ more'

After proceeding all orders, print the **total money earned** from the **successful** orders in the format: **'Income Report: {totalMoney}\$'**

All of the numbers should be **formatted to the second decimal point**.

Example

Input	Output
['1.00, coffee, caffeine, milk, 4', '0.40, tea, milk, 2', '1.00, coffee, decaf, 0']	You ordered coffee. Price: 1.00\$ Change: 0.00\$ Not enough money for tea. Need 0.60\$ more. You ordered coffee. Price: 0.90\$ Change: 0.10\$ Income Report: 1.90\$
Comments	
<p>The first order is coffee with caffeine, milk and sugar. The price of the drink is 0.80\$, we calculate the milk, 10% of the price, rounded to the first decimal point - 0.1\$, and we add the sugar => $0.80 + 0.10 + 0.10 = 1.00$.</p> <p>The second order is tea with milk and sugar ($0.80 + 0.10 + 0.10 = 1.00$), but the money inserted is not enough.</p> <p>Next, we receive order for coffee decaf with no milk and 0 sugar => 0.90\$. The change is 0.10\$.</p> <p>Total income = 1.90</p>	
Input	Output
['8.00, coffee, decaf, 4', '1.00, tea, 2']	You ordered coffee. Price: 1.00\$ Change: 7.00\$ You ordered tea. Price: 0.90\$ Change: 0.10\$ Income Report: 1.90\$