

Основна част

1. Сумата на 3 числа

Напишете JS функция, която да приема три числа като параметри и да извежда тяхната сума. Входът идва като масив от низови елементи, които трябва да бъдат обърнати към числа. Изходът трябва да бъде отпечатан на конзолата.

Вход	Изход
['2', '3', '4']	9

2. Сума и ДДС

Напишете JS функция, която сумира променлив брой цени и изчислява тяхното ДДС (данък върху добавената стойност, 20%).

Входът идва като масив от низови елементи, които трябва да бъдат обърнати към числа.

Броят на елементите ще бъде различен.

Изходът трябва да съдържа сумата, ддс-то и крайното салдо.

3. Буква в низ

Напишете JS функция, която отчита колко пъти се появява определена буква в даден низ.

Входът идва като масив. Първият елемент е низът за проверка, а вторият елемент е буквата за броене.

Изходът трябва да бъде върнат в резултат на функцията.

Вход	Изход
['hello', 'l']	2

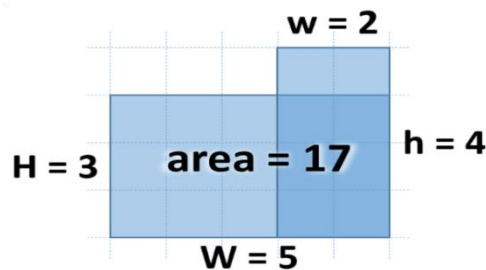
4. Площ на фигурата

Напишете JS функция, която изчислява площта на фигурата вдясно по зададени стойности за w, h, W и H.

Долният десен ъгъл винаги е общ за двата правоъгълника.

Входът идва като масив от низови елементи, които трябва да бъдат обърнати към числа.

Изходът трябва да бъде върнат в резултат на вашата функция.



5. Високосна година

Напишете JS функция, за да проверите дали една година е високосна. Високосните години са или делими на 4, но не и на 100, или са делими на 400.

Изходът на конзолата да показва Yes/No.

6. Низ от числа 1 ... N

Напишете JS функция, която чете число n като вход и отпечатва всички числа от 1 до n, слепени едно до друго.

Изходът трябва да бъде отпечатан на конзолата.

7. Разстояние между точките

Напишете JS функция, която изчислява разстоянието между две точки по зададени x и y координати. Използвайте обекти, за да съхраните двете точки.

Входът идва като масив от четири елемента, във формат [x1, y1, x2, y2]. (Не забравяйте да прехвърлите елементите на масива към число) . Всяка двойка елементи са координатите на точка в 2D пространство.

Изходът трябва да бъде отпечатан на конзолата.

Формулата за разстоянието между две точки

$$\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

8. Кутии и бутилки

Напишете JS функция за да изчислите колко кутии ще са необходими, за да се поберат n на брой бутилки, ако всяка кутия отговаря на k на брой бутилки.

Първият елемент е броя на бутилките, а вторият е вместимостта на една кутия.

Изходът трябва да бъде отпечатан на конзолата.

Вход	Изход
20 5	4

Вход	Изход
15 7	3

Вход	Изход
5 10	1

9. Площ на триъгълник

Напишете JS функция, която изчислява площта на триъгълник по неговите 3 страни.

<https://www.mathopenref.com/heronsformula.html>

10. Обем и повърхност на конус

Напишете JS функция за да изчислите обема и повърхността на конус по зададена височина и радиус на основата.

Входът идва като масив от два елемента. Първият елемент е радиусът на конуса, а вторият е неговата височина.

Изходът трябва да бъде отпечатан на конзолата на нов ред за всеки резултат.

<https://www.calculatorsoup.com/calculators/geometry-solids/cone.php>

11. Четно или Нечетно

Напишете JS функция, която проверява дали число е нечетно,четно или невалидно (дробите не са нито нечетни, нито четни).

Изходът трябва да бъде отпечатан на конзолата.

12. Прости числа

Напишете JS функция, за да проверите дали число е просто .

13. Разстояние

Два обекта започват от точка А и пътуват в една и съща посока с постоянни скорости V1 и V2 за период Т. Напишете JS функция, която изчислява разстоянието между двата обекта в края на периода.

Входът идва като масив. Първите два елемента са скоростите в км / час, а третият елемент е времето в секунди.

Изходът трябва да бъде отпечатан на конзолата.

Изчислете разстоянието в метри.

Малко помощ:

Скоростта, времето и разстоянието са свързани помежду си със следната формула:

$$S = V * T$$

Първо трябва да се изравнят входящите. Има 3600 секунди за час и 1000 метра на километър.

Не знаем кой обект е изминал по-голямо разстояние!

14. Присвояване на свойства

Напишете JS функция, която съставя обект по зададени свойства.

Вход
['name', 'Pesho', 'age', '23', 'gender', 'male']
Изход
{ name: 'Pesho', age: '23', gender: 'male' }

15. Най – голямото число сред 3

Напишете JS функция, която намира най-голямото от 3 числа.

Входът идва като масив.

Изходът е най-големият от входните числа.

16. Функционален калкулатор

Напишете JS програма, която получава две променливи и оператор и извършва изчисление.

Вход	Изход
['18', '-1', '*']	-18

17. Големи думи

Напишете JS програма, която извлича всички думи от низ и ги преобразува, така че всички букви да са главни.

Вход	Изход
'Hi, how are you?'	HI, HOW, ARE, YOU

Допълнителна част

Ако се интересуваш от нови и яки технологии, свързани с програмирането, не може да не си чувал за [Докер](#).

Накратко казано, това е набор от продукти тип „платформа като услуга“ (PaaS), които използват виртуализация на ниво операционна система, за да доставят някакъв вид софтуер в пакети, наречени контейнери. Контейнерите са изолирани един от друг и обединяват собствения си софтуер, библиотеки и конфигурационни файлове; те могат да комуникират помежду си по добре дефинирани канали. Всички контейнери се управляват от едно ядро на операционната система и поради това използват по-малко ресурси от виртуалните машини.

Docker може да пакетира приложение и неговите зависимости във виртуален контейнер, който може да работи на всеки компютър с Linux, Windows или macOS. Това дава възможност на приложението да се изпълнява на различни места, като локално, в публичен облак и / или в частен облак.

Тъй като контейнерите на Docker са леки, един сървър или виртуална машина може да изпълнява няколко контейнера едновременно.

Основните класове на Docker обекти са изображения, контейнери и услуги.

Контейнерът на Docker е стандартизирана, капсулирана среда, която изпълнява приложения. Той се управлява с помощта на Docker API или CLI.

Изображението на Docker е шаблон само за четене, използван за изграждане на контейнери. Изображенията се използват за съхранение и изпращане на приложения.

Регистърът на Docker е хранилище за изображения на Docker.

Задача:

Събери кода на всички методи от задачите по горе в един "index.js" файл.
Напиши Dockerfile с инструкции за изпълнение на index.js.

```
FROM node:current-slim

LABEL MAINTAINER=YourName

# Copy source code to /src in container

# Document the port the app listens on

# Run this command (starts the app) when the container
starts

<- something in here -> node ./index.js
```

С коя команда се изгражда изображение от Dockerfile?

Коя команда трябва да изпълните за стартиране на контейнер въз основа на новото ви изображение?

Може ли да създадете "Конвейер за непрекъсната интеграция" (pipeline for continuous integration), който да ползва вашият докер файл?