Node.js Tasks #1

Note: These tasks are for self-learning purposes. They do not have an sort of grading weight-age. You are allowed to Google syntax. But you must implement all these tasks by your own code. Even if you copy parts of the code, make sure that you first understand that piece of code before pasting it as your source code.
You are required to provide a .zip file of all these tasks with the naming convention of task_n.js Inside each .js file you must first paste the task statement and comment it out, and start your solution then. Each task file must have code for one task only.

Task 1: Write a **function** that takes an **array of integers** as input and **returns** the the **count** of **even** numbers.

Example:

Input: **[1, 2, 3, 4, 5, 6, 7, 8, 9]** → Output: **4**

Task 2: Create a custom **object**. **Sort** that object by its **keys** and return the corresponding **values** of those sorted keys in an **array**.

Hint: Search **Object.keys()** method

Example:

Input:

```
{
    fatima:   23,
    ali:      10,
    zain:     45,
}
```

object sorted by keys:

```
{
    ali:      10,
    fatima:   23,
    zain:     45,
}
```

Output: **[10, 23, 45]**

Task 3: Create a function that takes an **array** of custom objects **Users** and returns an array of all the users that live in a certain country. (Your function must not exceed one line of code)

Task 4: Create a **function** that takes in an **array** of numbers and returns a resultant array after **transforming** the input array into an array of **cubes** of its elements. (Your function must not exceed one line of code)

Input: [1, 2, 3] → Output: [1, 8, 27]

Task 5: Create a **function** that takes in a **string** and a **substring** and returns **true** or **false** based on the **condition** that substring **exists** in the given string.

Input:
string: “Hello World” substring: “llo”
Output: true

Task 6: Given an **array** of numbers. Find the **index** of the **first occurrence of the most frequent element** in it. If all elements have the **same** frequency return **-1.**

Input: [1, 1, 2, 3, 3, 4] → Output: 0

Input: [1, 2, 3, 4, 5] → Output: -1

Input: [1, 2, 3, 3, 3] → Output: 2

Task 7: Given an **array** of numbers. Find the **index** of the **first prime** number. Return **-1** if no prime is found.

Input: [6, 8, 4, 3, 2] → Output: 3

Input: [32, 40, 6, 1, 9, 10, 23] → Output: 6

Task 8: Create a **function** that takes an **array** of numbers as **input** and returns **true** or **false** based on the **condition** that **every** number is **even.**
Hint: Search for **Array.every()** method.

Input: [2, 3, 4, 6, 8, 10] → Output: false

Input: [2, 4, 6, 8, 10, 12] → Output: true

Task 9: Without using a **loop** initialize an array with first 1000 natural numbers.
Hint: Search **Array.fill()**

Output: [1, 2, 3, 4, 5, …, 998, 999, 1000]

Task 10: Create a **function** that takes a **string** and an **array** of **strings** as input. It then returns **true** or **false** based on the condition that the **input string** is present in the **array** of **strings**. The **search** should be **case-insensitive.**

Input String: “Hello” Input Array: [“World, “Apple”, “heLLo”]

Output: true

Input String: “Hello” Input Array: [“World, “Apple”, “Hel”]

Output: false

Task 11: Create a **function** that takes in an **array** of numbers or strings and returns **true** or **false** if the elements in the array form a palindrome when traversed in order.

Input: [1, 0, 0, 1] → Output: true

Input: [‘h’, ‘e’, ‘y’] → Output: false

Input: [‘a’, ‘v’, ‘a’] → Output: true

Task 12: Prove with code that **data types** in **Javascript** have **Truthy** and **Falsey** values.

Task 13: Create a **function** to generate a **random** integer. And build a small **dice** game.
Hint: Explore **Math** library and how to take **user input** in Javascript

Task 14: Create a function that accepts a **string** as **input** and returns a **resultant** string with the **first letter** of **each word** of the string in **upper case.**

Input: “the quick brown fox jumps over the lazy fox”

Output: “The Quick Brown Fox Jumps Over The Lazy Fox”

Task 15: Create a **function** that takes an **array** of **country names** and returns the **country** that has the **longest name**. If two or more countries have the same largest length then return the first occurrence.

Input: [‘Canada’, ‘Belgium’, ‘United Arab Emirates’, ‘Pakistan’]

Output: United Arab Emirates

Input: [‘Italy’, ‘Spain’, ‘Iran’]

Output: Italy

Task 16: Create a function that generates a **string id** of **input length n** of **random characters.**

Input: 6 → Output: Rt8Ad4

Input: 8 → Output: 1Yq88bzS