Basic Level Tasks

1. **Print Array Elements (using for loop)**
   - **Given: let arr = [10, 20, 30, 40, 50];**
   - **Task: Print all elements using a for loop.**
2. **Sum of Elements using forEach**
   - **Given: let nums = [1, 2, 3, 4];**
   - **Task: Use forEach() to calculate and print the sum.**
3. **Loop Over Object with for...in**
   - **Given:**

     ```
     let person = { name: "Ramesh", age: 30, city:
     "Mumbai" };
     ```

   - **Task: Print all keys and values using for...in.**
4. **Print Characters in a String using for...of**
   - **Given: let str = "Hello";**
   - **Task: Print each character using for...of.**

Medium Level Tasks

5. **Reverse an Array using a for loop**
   - **Task: Write a function that reverses an array manually.**
6. **Find Maximum Value in Array using for...of**
   - **Given: let nums = [23, 45, 12, 78];**
   - **Task: Use for...of to find the maximum number.**
7. **Convert Array to Object using forEach**
   - **Given: let keys = ["id", "name", "email"]; let values = [101, "Ganesh", "ganesh@example.com"];**
   - **Task: Create an object { id: 101, name: "Ganesh", email: "ganesh @example.com" }.**
8. **Loop Over Nested Object using for...in**
   - **Given:**

     ```
     let user = {
       name: "Banny",
       address: {
         city: "Delhi",
         zip: "110001"
       }
     ```

};

- **Task: Loop through both outer and inner keys and print.**

▲ Hard Level Tasks

9. **Custom Implementation of forEach()**
   - **Task: Create your own myForEach() function that behaves like the built-in forEach().**
10. **Flatten a Nested Array using Loops**

- **Given: let nested = [1, [2, 3], [4, [5, 6]]];**
- **Task: Flatten this array to one level [1, 2, 3, 4, 5, 6].**

11. **Group Array Items by Data Type**

- **Given: let mix = [1, "two", 3, true, "four", null];**
- **Task: Output an object:**

```
{
 number: [1, 3],
 string: ["two", "four"],
 boolean: [true],
 object: [null]
}
```

12. **Deep Comparison of Two Arrays**

- **Task: Write a function deepEqual(arr1, arr2) that checks if both arrays are deeply equal in structure and values.**

Basic Level Tasks

1. **Create and Access**
   - **Create an array of 5 city names.**
   - **Print the first and last elements using index access.**
2. **Loop Through an Array (for loop)**
   - **Create an array of 10 numbers.**

    - Use a for loop to print all numbers.
3. Using forEach()
    - Given: let fruits = ["apple", "banana", "mango"];
    - Print: Fruit: apple, Fruit: banana, etc. using forEach.
4. Using push() and pop()
    - Start with an array [1, 2, 3].
    - Add 4 and 5 to the end using push().
    - Remove the last item using pop().
5. Check Existence
    - Use includes() to check if "banana" is in your fruit array.

Medium Level Tasks

6. Transform with map()
    - Given: let nums = [1, 2, 3, 4];
    - Return a new array with each element doubled.
7. Use filter()
    - Given: let ages = [12, 25, 17, 30, 15];
    - Filter and print only ages greater than 18.
8. Sort with Numbers
    - Given: let points = [40, 100, 1, 5, 25];
    - Sort numerically in ascending order.
9. Use splice()
    - Remove the second and third items from an array of 5 elements.
    - Insert "hello" and "world" at index 2.
10. Reduce Array to Sum

- Sum all values in [5, 10, 15] using reduce().

Hard Level Tasks

11. Nested Array Access

- Given:

  let matrix = [

```
  [1, 2, 3],
  [4, 5, 6],
  [7, 8, 9]
];
```

- Print the middle value (5) using nested indexing.

12. Find Unique Elements

- Given: let nums = [1, 2, 2, 3, 4, 4, 5];
- Write code to return a new array with only unique values.

13. Chain map, filter, reduce

- Given: let nums = [1, 2, 3, 4, 5];
- Task: Square the numbers, filter even results, and sum them.

14. Sort Array of Objects

- Given:

```
let users = [
  { name: "Mahesh", age: 25 },
  { name: "Shubham", age: 20 },
  { name: "Rahul", age: 30 }
];
```

- Sort users by age in ascending order.

15. Manual Implementation of map()

- Create a custom myMap() function that mimics map().

16. Matrix Sum

- Write a function that accepts a 2D array and returns the total sum of all values inside it.

17. Simulate Stack

- Use only push and pop to simulate stack operations:
    - push(10)

- **push(20)**
- **pop()**
- **Print final stack.**