**E-commerce Problem Statement on arrays**

In an e-commerce application, the absence of arrays poses a significant challenge in managing product data. Without arrays, it becomes challenging to efficiently store, organize, and manipulate information about the numerous products available on the platform. To overcome this, arrays are vital.

Task:

1**. Creating an Array**: Define an array that stores product information for an e-commerce application. Each element in the array should be a string containing product details, such as "Product Name - Price - Category - Availability."

2. **Access and Display**: Write code to access and display the details of a specific product from the array. This demonstrates how arrays allow you to retrieve and display information efficiently.

3. **Adding Products**: Add new products to the array by pushing new strings. Show how arrays are dynamic and how products can be inserted into the array as new items become available.

4. **Removing Products**: Remove discontinued products from the array using the `splice()` method. This illustrates how arrays help in managing changes in the product catalog.

5. **Price Calculation**: Calculate the total price of products in a specific category by iterating through the array. This demonstrates how arrays facilitate bulk operations on data.

6. **Sorting**: Implement a function to sort the products based on their prices. This showcases the sorting capabilities of arrays.

8. **Reporting**: Generate a report that summarizes product information, such as the number of products in each category, by iterating through the array. This showcases the use of arrays for reporting and analysis.

9**. Convert Product Prices to a Different Currency**:- You have an array of products with their prices in USD. convert all prices to a different currency (e.g., Euros or Yen).

10. **Calculate Total Cart Value**: Given an array of products and their quantities in a shopping cart, calculate the total value of the items in the cart.

11. **Filter Out-of-Stock Products**:- You have an array of products, some of which are out of stock (quantity is 0). create a new array containing only the in-stock products.

12. **Display Product Names**:- display the names of products on the product listing page.

13. **Sort Products by Price**:- Implement a feature to sort products on a category page by their prices in ascending order

14. **Add Discounts to Products**: - Given an array of products, apply a discount (e.g., 10% off) to each product's price.

15. **Check for Expensive Products**:- check if there's at least one product in the array with a price greater than a certain threshold (Rs 1000).

16. **Check if All Products Are on Sale**:- check if all products in the array are currently on sale (e.g., have a discount).

17. **Display Nested Product Categories**: - You have an array of products with nested categories. Use the `flat` method to flatten the array and display a single-level list of categories.

18. **List Product Tags**:- Each product has an array of tags associated with it. Use the `flatMap` method to extract and display a single-level list of unique tags.

19. **Search for a Specific Product**:- Implement a search to locate and display a specific product by its name.

20. **Find the Index of a Product:-** Create a search feature that uses the `findIndex` method to locate and display the index of a specific product by its name.

21. **Calculate Total Savings**:- Given an array of products with discounts, calculate the total savings when each product is purchased.

22**. Extract Usernames from Email Addresses**:- You have an array of user data with email addresses. extract and display a list of usernames.

23. **Calculate Total Shipping Cost**:- Given an array of products and their weights, calculate the total shipping cost based on weight.

24**. List Unique Product Categories**:- create a list of unique product categories from an array of products.

25**. Filter Top-Rated Products**:- You have an array of products with ratings. create a new array containing only the top-rated products (e.g., ratings greater than 4 out of 5).

26. **Calculate Average Product Ratings:-** calculate and display the average rating for a set of products.

27. **Display Products with Odd Prices**:- display the names of products with odd prices.

28. **Sort Products by Name**:- Implement a feature to sort products on a category page alphabetically by their names.