Assignment-3: Array

Note:Use javascript function to solve these problems.

Problem 1: Grouping Employees by ID

Scenario:
IQ-India, a corporate company, is facing challenges in managing a growing number of employees. To streamline operations and organize their workforce, the company has decided to divide employees into two groups based on their employee IDs:

1. **Group A (Even IDs):** Employees with even-numbered IDs.
2. **Group B (Odd IDs):** Employees with odd-numbered IDs.

Task:
Your task is to write a program that separates the employee IDs into these two groups.

Problem 2: Categorizing Transactions in Banking

Scenario:
Imagine you're processing bank transactions and want to categorize them based on whether the transaction amount is even or odd. You also want to include the index to track the order of transactions.

Task:
You need to process a list of bank transaction amounts and categorize them based on whether the transaction amount is **even** or **odd**. Additionally, include the **index** of each transaction to track the order in which they occurred.

For each transaction, if the amount is **even**, categorize it as **"Approved"**. If the amount is **odd**, categorize it as **"Requires Manual Review"**.

Expect Output:

"Transaction 1: 1200 - EVEN - Approved",
"Transaction 2: 755 - ODD - Requires Manual Review",

Problem 3: Retail Store Management.

Scenario:
Imagine you work for a grocery store, and you are tasked with creating a system to categorize grocery items based on their prices. The system should allow customers to enter a list of grocery items along with their prices. Then, the system will categorize each item as either "On Sale" or "Regular Price" based on whether the price is below or above a certain threshold (in this case, 200).

Task:

1. **Input Items:**
    - The system will prompt the user (a store employee or customer) to input how many items they want to add to the grocery list.
    - For each item, the user must input the name and price.

Categorize Items:

- If the item price is **below 200**, it will be categorized as "On Sale."
- If the item price is **200 or more**, it will be categorized as "Regular Price."

Problem 4: Weather Monitoring System.

Scenario:
Imagine you're working at a weather monitoring agency that collects daily temperature data from different states. As a part of your job, you are tasked with identifying the state experiencing the highest temperature on a given day.

Problem:

The agency provides temperature readings for various states daily, and you need a quick way to determine which state has the highest temperature and report it.

Use of the Program:

You use this JavaScript program to:

1. Ask users how many cities want to enter
2. Enter the names and temperatures of all the states for that day.
3. The program evaluates the data and determines the state with the highest temperature and lowest.
4. The program outputs the state name along with its temperature for reporting.

Output:
The state with the highest temperature in Delhi, with a temperature of 38°C
The state with the Lowest temperature is Mumbai, with a temperature of 25°C.

Problem 5: Customer Name Filtering.

Create a JavaScript program to filter and display customer names from a list based on a specific condition: you need to find and display customers whose names start with the specific letter.

Task:

1. You are given an array of customer names. The names are stored as strings in the array.
2. Your task is to filter the names and **display only those customers whose names start with the letter 'R' or 'r'**.
3. The program should output each matching name in the console.

Problem 6:Library Catalog Management.

Imagine you are working at a library and you have a collection of books with varying titles. You need to find the book with the longest title to highlight it in a special section or for designing its book cover in a way that holds its long title.

book_Titles = ["The Great Gatsby", "Atomic habits", "The Catcher in the Rye", "War and Peace", "The Old Man and the Sea", "A Brief History of Time"];

Task:
Create a program that helps you find the book with the longest title from a list of books in a library catalog. This information can be useful for designing the book cover, featuring the book in a special section, or highlighting it for marketing purposes.

Problem 8:User Name Validation or Reporting in a System.

In an application Usernames are critical for identifying users in any application. Ensuring

usernames belong to specific length requirements is essential for usability, security, and system consistency. Short usernames are easier to guess, making them a potential security risk, while overly long strings can lead to input abuse and impact database performance. Defined length restrictions help mitigate these risks. Analyzing usernames and reporting their lengths provide valuable insights, ensuring compliance with system rules and maintaining a secure, user-friendly environment

Task:

Write a function to Validate that all usernames meet the defined length requirements (e.g., between 3 and 10 characters) that are stored in an array .

const Usernames = ['aarav123', 'vihaan007', 'vivaan_k', 'ananya.star', 'ishita_01', 'aanya.sharma', 'rohan_raj', 'aryan_singh', 'sid.cool', 'priya_1995],

Output: ['aarav123:len-8', 'vihaan007:len-9', 'vivaan_k:len-8', 'ananya.star:len-11', 'ishita_01:len-9', 'aanya.sharma:len-12', 'rohan_raj:len-9', 'aryan_singh:len-11', 'sid.cool:len-8', 'priya_1995:len-10']

Problem 9:Employee management System

In an Employee Management System, it's often necessary to filter employee names based on a specific character, whether it's for search functionality or organizing data for reports. By implementing a feature that filters employee names by their first character, we can improve user experience, making it easier to access and manage employee records.

Task: Filter Employee Names by Character

You are tasked with writing a function that filters employee names from a list based on the first character of the name. For example, if the user wants to filter all employees whose names start with the letter "A/a", the system should return only those employee names that start with "A/a".

Problem 10:E-commerce Product Descriptions.

In an e-commerce application, product descriptions can be long. To improve user

experience and keep the page clean, you can show only the first few characters of each product description with a "view more" . This approach lets users quickly scan products without being overwhelmed by large blocks of text.

Task:
You have a list of products with detailed descriptions. Displaying the first 20 characters of each description with a "view more" allows users to decide if they want to read further.

let product_Descriptions = [

'This new smartwatch is packed with features including heart rate monitoring, sleep tracking, and a long battery life.',

'Our latest smartphone has an ultra-fast processor, an incredible camera, and long-lasting battery life.',

'This ergonomic office chair is designed to provide comfort for long hours of sitting, with adjustable features.',

'The new gaming laptop features a high-resolution display, powerful graphics card, and fast processing power.',

'Eco-friendly water bottles made from recycled materials that keep your beverages hot or cold for hours.'

];

Output:

```
[
  'This new smartwatch... view more',
  'Our latest smartphone... view more',
  'This ergonomic offi... view more',
  'The new gaming lap... view more',
  'Eco-friendly water ... view more'
]
```

Problem :11 Menu-Driven Program for Employee Management (CRUD Operations with Arrays)

Problem Description:

Create a menu-driven program that allows you to manage employee data using basic CRUD (Create, Read, Update, Delete) operations. You will use an array to store employee details. The program should present the user with options to perform various operations on the employee data.

The operations you need to implement are as follows:

1. Add Employee at Last Position
   Add a new employee at the end of the employee list.

2. Add Employee at First Position
   Add a new employee at the beginning of the employee list.

3. Add Employee at Specific Position
   Add a new employee at a specific position in the list.

4. Read Employee at First Position
   Display the employee details at the first position in the list.

5. Read Employee at Last Position
   Display the employee details at the last position in the list.

6. Read Employee at Specific Position
   Display the employee details at a specific position in the list.

7. Delete Employee at First Position
   Remove the employee from the first position in the list.

8. Delete Employee at Last Position
   Remove the employee from the last position in the list.

9. Delete Employee at Specific Position
   Remove the employee at a specific position in the list.

10. Update Employee at First Position
    Update the employee details at the first position in the list.

11. Update Employee at Last Position
    Update the employee details at the last position in the list.

12. Update Employee at Specific Position
    Update the employee details at a specific position in the list.

13. Show All Employees
    Display the entire list of employees.

14. Exit
    Exit the program.

Task Requirements:

1. Input: User input will determine the action to be taken, such as adding, reading, updating, or deleting employees.
2. Output: The program should display the employee details or a message indicating the success or failure of an operation.
3. Array Operations: Use JavaScript array methods to implement the operations .

Example Output:

1. Add Employee at Last Position

2. Add Employee at First Position

3. Add Employee at Specific Position

4. Read Employee at First Position

5. Read Employee at Last Position

6. Read Employee at Specific Position

7. Delete Employee at First Position

8. Delete Employee at Last Position

9. Delete Employee at Specific Position

10. Update Employee at First Position

11. Update Employee at Last Position

12. Update Employee at Specific Position

13. Show All Employees

14. Exit

Enter your choice: 1

Enter Employee Name: Dev

Employee added at last position: Dev

Enter your choice: 13

Employee List: ['J', 'k', 'Dev']