Nested array, object combination crud operations

1. Access Deeply Nested Properties

Question: Retrieve the `city` where the company `TechCorp` is located.

```
const companies = [
  {
    id: 1,
    name: 'TechCorp',
    details: {
      address: {
        street: '123 Elm St',
        city: 'New York',
      },
    },
  },
  {
    id: 2,
    name: 'SoftSystems',
    details: {
      address: {
        street: '456 Pine St',
        city: 'San Francisco',
      },
    },
  },
];

Expected Output: 'New York'
```

2. Update a Nested Property

Question: Update the `salary` of the employee `Bob` to `75000`.

```
const employees = [
  {
    id: 1,
    name: 'Alice',
    details: {
      salary: 50000,
      role: 'Developer',
    },
  },
  {
    id: 2,
    name: 'Bob',
    details: {
      salary: 60000,
      role: 'Manager',
    },
  },
];

Expected Output:

[
  { id: 1, name: 'Alice', details: { salary: 50000, role: 'Developer' } },
  { id: 2, name: 'Bob', details: { salary: 75000, role: 'Manager' } },
]
```

3. Add a New Item to a Nested Array

Question: Add `'Node.js'` to the `skills` array of the developer `Alice`.

```
const team = [
  {
    name: 'Alice',
    skills: ['HTML', 'CSS', 'JavaScript'],
  },
  {
    name: 'Bob',
    skills: ['Java', 'Spring Boot'],
  },
];
```

```
Expected Output:
[
  { name: 'Alice', skills: ['HTML', 'CSS', 'JavaScript', 'Node.js'] },
  { name: 'Bob', skills: ['Java', 'Spring Boot'] },
]
```

4. Filter by Nested Property

Question: Retrieve all tasks assigned to the employee `Alice`.

```
const projects = [
  {
    project: 'Website',
    tasks: [
      { task: 'Design', assignedTo: 'Alice' },
      { task: 'Code', assignedTo: 'Bob' },
    ],
  },
  {
    project: 'App',
    tasks: [
      { task: 'Develop', assignedTo: 'Alice' },
      { task: 'Test', assignedTo: 'Charlie' },
    ],
  },
];

Expected Output:
[
  { task: 'Design', assignedTo: 'Alice' },
  { task: 'Develop', assignedTo: 'Alice' },
]
```

5. Access Dynamic Properties

Question: Access the `role` of `Charlie` dynamically using the `key` variable.

```
const team = {
  Alice: { role: 'Developer', age: 25 },
  Bob: { role: 'Manager', age: 30 },
  Charlie: { role: 'Tester', age: 28 },
};
const key = 'Charlie';

Expected Output: 'Tester'
```

6. Find an Item by Nested Property

Question: Write a function to find the product with the `price = 500`.

```
const catalog = [
  {
    id: 101,
    name: 'Laptop',
    details: { price: 1000, stock: 5 },
  },
  {
    id: 102,
    name: 'Phone',
    details: { price: 500, stock: 10 },
  },
];

Expected Output:
{ id: 102, name: 'Phone', details: { price: 500, stock: 10 } }
```

7. Count Items in Nested Arrays

Question: Count the total number of tasks in all projects.

```
const projects = [
  {
    name: 'Website',
    tasks: ['Design', 'Develop', 'Test'],
  },
  {
    name: 'App',
    tasks: ['Plan', 'Build'],
  },
];

Expected Output: 5
```

8. Delete an Item from a Nested Array

Question: Write a function to remove the `Tablet` from the `inventory`.

```
const inventory = [
  {
    category: 'Electronics',
    items: ['Laptop', 'Phone', 'Tablet'],
  },
  {
    category: 'Furniture',
    items: ['Table', 'Chair'],
  },
];

Expected Output:
[
  { category: 'Electronics', items: ['Laptop', 'Phone'] },
  { category: 'Furniture', items: ['Table', 'Chair'] },
]
```

9. Find the Highest Salary

Question: Write a function to find the employee with the highest salary.

```
const employees = [
  {
    name: 'Alice',
    details: { salary: 50000, role: 'Developer' },
  },
  {
    name: 'Bob',
    details: { salary: 75000, role: 'Manager' },
  },
];

Expected Output:
{ name: 'Bob', details: { salary: 75000, role: 'Manager' } }
```

10. Group by Nested Property

Question: Group the employees by their `role`.

```
const employees = [
  { name: 'Alice', role: 'Developer' },
  { name: 'Bob', role: 'Manager' },
  { name: 'Charlie', role: 'Developer' },
];

Expected Output:
{
  Developer: [
    { name: 'Alice', role: 'Developer' },
    { name: 'Charlie', role: 'Developer' },
  ],
  Manager: [{ name: 'Bob', role: 'Manager' }],
}
```

11. Update a Nested Array

Question: Update the stock of the Phone to 15.

```
const catalog = [
  {
    id: 101,
    name: 'Laptop',
    details: { price: 1000, stock: 5 },
  },
  {
    id: 102,
    name: 'Phone',
    details: { price: 500, stock: 10 },
  },
];

Expected Output:
[
  { id: 101, name: 'Laptop', details: { price: 1000, stock: 5 } },
  { id: 102, name: 'Phone', details: { price: 500, stock: 15 } },
]
```

12. Extract Nested Values

Question: Extract the names of all employees from the `departments`.

```
const departments = [
  {
    name: 'HR',
    employees: ['Alice', 'Bob'],
  },
  {
    name: 'IT',
    employees: ['Charlie', 'David'],
  },
];

Expected Output: ['Alice', 'Bob', 'Charlie', 'David']
```

13. Flatten a Nested Array

Question: Write a function to flatten the `nestedArray`.

```
const nestedArray = [[1, 2], [3, 4], [5, 6]];

Expected Output: [1, 2, 3, 4, 5, 6]
```

14. Filter Nested Array by Condition

Question: Retrieve all employees older than `30`.

```
const employees = [
  { name: 'Alice', details: { age: 25, role: 'Developer' } },
  { name: 'Bob', details: { age: 35, role: 'Manager' } },
];

Expected Output:
[{ name: 'Bob', details: { age: 35, role: 'Manager' } }]
```

15 . Remove a Product by Name

Write a function to remove a product from the `products` array by its `name`.

Input:

```
const products = [
  { name: 'Laptop', price: 1000 },
  { name: 'Phone', price: 500 },
  { name: 'Tablet', price: 300 }
];
```

Expected Output:

```
[
  { name: 'Laptop', price: 1000 },
  { name: 'Tablet', price: 300 }
]
```