

Here's a complex nested object literal representing a company with multiple departments, employees, and projects they're working on:

```
const company = {
  name: "Tech Innovators Inc.",
  founded: 2010,
  location: "San Francisco",
  departments: [
    {
      name: "Engineering",
      budget: 500000,
      employees: [
        { id: 1, name: "John Doe", position: "Software Engineer", salary: 80000, projects: ["Project A", "Project B"] },
        { id: 2, name: "Michael Johnson", position: "Lead Developer", salary: 110000, projects: ["Project A"] },
        { id: 3, name: "Sarah Brown", position: "DevOps Engineer", salary: 95000, projects: ["Project C"] }
      ],
      projects: [
        { name: "Project A", budget: 200000, status: "active" },
        { name: "Project B", budget: 150000, status: "completed" },
        { name: "Project C", budget: 100000, status: "active" }
      ]
    },
    {
      name: "Marketing",
      budget: 300000,
      employees: [
        { id: 4, name: "Anna White", position: "Marketing Manager", salary: 85000, projects: ["Project D"] },
        { id: 5, name: "Chris Lee", position: "SEO Specialist", salary: 60000, projects: ["Project D", "Project E"] }
      ],
      projects: [
        { name: "Project D", budget: 100000, status: "active" },
        { name: "Project E", budget: 120000, status: "inactive" }
      ]
    }
  ],
}
```

```

    {
      name: "Human Resources",
      budget: 100000,
      employees: [
        { id: 6, name: "Emily Davis", position: "HR Manager", salary:
70000, projects: ["Project F"] },
        { id: 7, name: "David Wilson", position: "Recruiter", salary:
50000, projects: [] }
      ],
      projects: [
        { name: "Project F", budget: 50000, status: "active" }
      ]
    }
  ]
};

```

Assignment Questions:

Beginner Level (Basic Object and Array Manipulation)

1. **Access Department Data:** Write a function that takes the name of a department as input and returns the budget of that department. If the department doesn't exist, return "Department not found".
2. **List Employee Names:** Write a function that takes a department name and returns a list of all employee names working in that department.
3. **Employee Count in Department:** Write a function that returns the number of employees working in a specified department.
4. **Get Project Names for Department:** Write a function that returns the names of all the projects in a given department.
5. **Access the Company Name:**
Retrieve the `name` of the company from the `company` object.
6. **Get the Budget of the First Department:**
Access and retrieve the `budget` of the first department in the `departments` array.
7. **Get the Position of Employee with ID 2:**
Retrieve the `position` of the employee with `id: 2` from the `employees` array of the "Engineering" department.
8. **Find the Department of Employee "Anna White":**
Access the department name for the employee named "Anna White".
9. **Get the Status of Project "Project E":**
Retrieve the `status` of the project named "Project E" in the `projects` array of the "Marketing" department.
10. **Get the Salary of Employee with ID 6:**
Access the `salary` of the employee with `id: 6` from the "Human Resources" department.

11. **Access the List of Employees in the "Engineering" Department:**
Retrieve the `employees` array from the "Engineering" department and access all employees in that department.
 12. **Get the Name of the Last Employee in the "Marketing" Department:**
Access and retrieve the `name` of the last employee in the "Marketing" department.
 13. **Find the Budget of "Project C":**
Retrieve the `budget` of the project named "Project C" from the "Engineering" department.
 14. **Get the Number of Projects in the "Marketing" Department:**
Retrieve the length of the `projects` array for the "Marketing" department.
 15. **Access the Department Budget for "Human Resources":**
Retrieve the `budget` for the "Human Resources" department.
 16. **Find the Name of the First Project in the "Marketing" Department:**
Access and retrieve the `name` of the first project in the "Marketing" department.
 17. **Get the List of All Project Names in the "Engineering" Department:**
Retrieve the `name` of all projects in the "Engineering" department.
 18. **Access the Location of the Company:**
Retrieve the `location` of the company from the `company` object.
 19. **Get the Status of All Projects in the "Human Resources" Department:**
Retrieve and list the `status` of all projects in the "Human Resources" department.
 20. **Get the Salary of the Employee Named "Chris Lee":**
Retrieve the `salary` of the employee named "Chris Lee".
 21. **Find the Name of the Department with the Highest Budget:**
Access the department that has the highest `budget` and retrieve its `name`.
 22. **Access the Employee with the Highest Salary in the "Engineering" Department:**
Retrieve the `name` and `salary` of the employee with the highest salary in the "Engineering" department.
-

These questions focus on **directly accessing** properties from objects and arrays without involving functions, making them great practice for understanding how to navigate through nested data structures.

Let me know if you need clarification or more examples!

Intermediate Level (Filtering, Searching, and Updating)

23. **List Employees on Active Projects:** Write a function that returns the names of employees who are working on at least one active project.
24. **Total Department Budget:** Write a function that takes the name of a department and returns the total budget of the department (including all its projects).
25. **Update Employee Salary:** Write a function that updates the salary of an employee by ID and returns the updated employee data. If the employee doesn't exist, return "Employee not found".
26. **Find Project with Highest Budget:** Write a function that returns the project with the highest budget across all departments.

27. **Add a New Employee:** Write a function that adds a new employee to a specified department. The employee should have an ID, name, position, salary, and a list of projects.
28. **Filter Employees by Salary:** Write a function that takes a salary threshold as input and returns the names of all employees whose salary is greater than the specified threshold.

Advanced Level (Complex Data Manipulation)

11. **Find Employees with Multiple Projects:** Write a function that returns the names of employees who are working on more than one project.
12. **Total Salary Expense by Department:** Write a function that calculates and returns the total salary expense for each department. The result should be an object where the keys are department names and the values are total salaries.
13. **Count Active Projects in Each Department:** Write a function that returns an object where the keys are department names and the values are the number of active projects in that department.
14. **Employee with the Highest Salary in Each Department:** Write a function that returns the employee with the highest salary in each department. The result should be an object where the keys are department names, and the values are the employee with the highest salary.
15. **Promote Employee:** Write a function that promotes an employee by increasing their salary by a given percentage. If the employee doesn't exist, return "Employee not found". After promotion, return the updated employee data.
16. **Department with Most Projects:** Write a function that returns the name of the department with the most projects. If there's a tie, return the departments with the most projects in an array.
17. **Project Status Report:** Write a function that returns a report with the names of all projects and their statuses across all departments in the following format: `Project Name: [name], Status: [status]`.
18. **Find All Employees Involved in Active Projects:** Write a function that returns the names of all employees who are working on projects that are marked as "active."