Array Assignments

Learning Objectives: In this assignment, you'll get hands-on experience with TypeScript arrays. By working with real-world scenarios involving employee data, car details, and product inventory, you'll reinforce your understanding of:

- Basic arrays and their operations
- Typed arrays and index access
- Multi-dimensional arrays and tuples

Part 1: Basic Arrays - Product Inventory

Challenge:

You are tasked with creating a Product Inventory system. Implement functions and logic to manage object manipulation effectively.

1. Define a type alias named **Product** to represent a product with the following properties:

- **name (string):** The name of the product.
- **price (number):** The price of the product.
- **inventory (object):** An object containing inventory details including:
- **stock (number):** The number of products available.
- **colorOptions (string[ ]):** An array listing available colors.

2. Create an array named **products** containing at least three product objects. Each product object should include **a name, price, and inventory details**.
3. Implement a function named **changeColor** that takes a **product object** and a **new color** as input. This function should update the color property of the product and adjust the price based on the new color (implement your own logic, e.g., increase by 10% for red, decrease by 5% for blue).
4. Display each **product's name, price, stock, and available colors**. Iterate through the products array and print each product's details.

Part 2: Multi-Dimensional Arrays and Tuples - Student Grades

Challenge:

You are tasked with creating a student grading system. Implement functions and logic to manage student grades effectively.

1. Define a TypeScript type alias named **Student** to represent a student with the following properties:
   - **name (string):** The name of the student.
   - **grades (number[ ]):** An array of grades for different subjects.

2. Create an array named **students** containing at least three student objects. Each student object should include a name and an array of grades.
3. Implement a function named **calculateAverageGrade** that takes a student's grades as input and returns the average grade for that student.
4. Display each **student's name** and **average grade.** Iterate through the **students array**, calculate the average grade for each student using the **calculateAverageGrade** function, and print their name and average grade.

Part 3: Array with Types and Indexing - Employee Salaries

Challenge:

You are managing employee salaries for a company. Implement logic to calculate salaries and handle bonuses.

1. Define a type alias named **Employee** with the following properties:

- **name (string):** The name of the employee.
- **hoursWorked (number):** The number of hours the employee worked.
- **hourlyRate (number):** The hourly rate of the employee.
- **salary (number):** The base salary of the employee.

2. Define an array named **employees** containing employee objects. Each employee object should include a name, **hoursWorked, hourlyRate,** and **salary**.
3. Implement a function named **calculateSalary** that calculates the salary for each employee based on the **hours worked** and **hourly rate**.
4. If an employee has worked **20 hours or more**, apply a **10% bonus** to their salary.