**Scenario 1: Styling a Blog Post**

You are tasked with styling a blog post on a website. The HTML structure for the blog post includes a title, author name, publication date, and the content of the post. Your goal is to make the blog post visually appealing.

Task 1: Create a CSS rule that styles the blog post title with a large and bold font, a different font color, and adds a subtle text shadow for emphasis.

Task 2: Style the author's name and publication date so that they are displayed in a smaller font size and italicized below the title. Ensure that there is enough spacing between these elements.

Task 3: Apply a background color to the entire blog post container, add some padding around the content, and set a border around the container to separate it from the rest of the page.

**Scenario 2: Responsive Navigation Menu**

You are working on a website's navigation menu, and you need to make it responsive for different screen sizes.

Task 1: Create a CSS rule that hides the navigation menu items when the screen width is less than 768 pixels. You should use media queries for this task.

Task 2: Add a "hamburger" icon (three horizontal lines) as a button that, when clicked on small screens, toggles the display of the navigation menu.

Task 3: Style the navigation menu items so that they are displayed in a horizontal line on larger screens and in a vertical list on smaller screens. Adjust padding and margins for proper spacing.

**Scenario 3: Form Validation**

You are building a registration form for a website and want to implement CSS styles to provide feedback to users when they enter incorrect information.

Task 1: Create a CSS rule that changes the border color of form input fields to red when a user enters incorrect data (e.g., an invalid email address).

Task 2: Add a CSS rule that displays a green checkmark icon next to input fields with correct data (e.g., a valid password).

Task 3: Style error messages below the input fields so that they are displayed in red text, with a red exclamation icon, when users submit incorrect data. These messages should be hidden by default and only appear when there's an error.