C PROGRAMMING
ASSIGNMENT 03.04

Problem 1: Write a C program that reads a day number (1-7) as an integer and displays the corresponding day name in word format.

Solution Approach:

1. Prompt the user to input a day number between 1 and 7.
2. Use a switch-case or if-else statements to map the number to the corresponding day name:
    - 1: Sunday
    - 2: Monday
    - 3: Tuesday
    - 4: Wednesday
    - 5: Thursday
    - 6: Friday
    - 7: Saturday
3. Print the day name based on the user's input.
4. If the input is not within the valid range, display an error message.

- **Test Case 1:**
  Input: `3`
  Expected Output:
  `Tuesday`
- **Test Case 2:**
  Input: `7`
  Expected Output:
  `Saturday`
- **Test Case 3:**
  Input: `8`
  Expected Output:
  `Invalid day number.`

Problem 2: Write a C program that reads a digit (0-9) and displays it in word format.

Solution Approach:

1. Prompt the user to input a digit between 0 and 9.
2. Use a switch-case or if-else statements to map the digit to its corresponding word:
    - 0: Zero
    - 1: One
    - 2: Two
    - 3: Three
    - 4: Four
    - 5: Five
    - 6: Six
    - 7: Seven
    - 8: Eight
    - 9: Nine
3. Print the word corresponding to the digit.
4. If the input is not a digit between 0 and 9, display an error message.

- **Test Case 1:**
  Input: `5`
  Expected Output:
  `Five`
- **Test Case 2:**
  Input: `0`
  Expected Output:
  `Zero`
- **Test Case 3:**
  Input: `12`
  Expected Output:
  `Invalid digit.`

Problem 3: Write a C program to read any month number (1-12) and display the number of days in that month.

Solution Approach:

1. Prompt the user to input a month number between 1 and 12.
2. Use a switch-case or if-else statements to determine the number of days:
    - 1, 3, 5, 7, 8, 10, 12: 31 days
    - 4, 6, 9, 11: 30 days
    - 2: 28 days (29 days for leap years; this can be extended for a leap year check)
3. Print the number of days for the corresponding month.
4. If the input is not within the valid range, display an error message.

Test Cases:

- **Test Case 1:**
  Input: `2`
  Expected Output:
  `February has 28 or 29 days.`
- **Test Case 2:**
  Input: `11`
  Expected Output:
  `November has 30 days.`
- **Test Case 3:**
  Input: `13`
  Expected Output:
  `Invalid month number.`

Problem 4: Write a C program that computes the area of various geometrical shapes (circle, rectangle, triangle) using a menu-driven approach.

Solution Approach:

1. Display a menu with options for different shapes (e.g., circle, rectangle, triangle).
2. Prompt the user to select an option from the menu.
3. Based on the selection:
   - For circle: Prompt for the radius and calculate the area using Area=π×r2.
   - For rectangle: Prompt for the length and width and calculate the area using Area=length×width.
   - For triangle: Prompt for the base and height and calculate the area using Area=1/2×base×height.
4. Print the computed area.
5. If the user selects an invalid option, display an error message.

Test Cases:

- **Test Case 1:**
  Input: `1 (Circle), Radius = 5`
  Expected Output:
  `Area of the circle = 78.54`
- **Test Case 2:**
  Input: `2 (Rectangle), Length = 4, Width = 6`
  Expected Output:
  `Area of the rectangle = 24.00`
- **Test Case 3:**
  Input: `3 (Triangle), Base = 8, Height = 5`
  Expected Output:
  `Area of the triangle = 20.00`
- **Test Case 4:**
  Input: `4`
  Expected Output:
  `Invalid option selected.`

Problem 5: Write a C program that is a menu-driven program to perform simple calculations such as addition, subtraction, multiplication, and division.

Solution Approach:

1. Display a menu with options for addition, subtraction, multiplication, and division.
2. Prompt the user to select an option from the menu.
3. Based on the selection:
   - For addition: Prompt for two numbers and calculate the sum.
   - For subtraction: Prompt for two numbers and calculate the difference.
   - For multiplication: Prompt for two numbers and calculate the product.
   - For division: Prompt for two numbers and calculate the quotient.
4. Print the result of the selected operation.
5. If the user selects an invalid option, display an error message.
6. Handle division by zero gracefully by displaying an error message.

Test Cases:

- **Test Case 1:**
  Input: `1 (Addition), Number1 = 10, Number2 = 5`
  Expected Output:
  `Sum = 15`
- **Test Case 2:**
  Input: `2 (Subtraction), Number1 = 10, Number2 = 5`
  Expected Output:
  `Difference = 5`
- **Test Case 3:**
  Input: `3 (Multiplication), Number1 = 10, Number2 = 5`
  Expected Output:
  `Product = 50`
- **Test Case 4:**
  Input: `4 (Division), Number1 = 10, Number2 = 2`
  Expected Output:
  `Quotient = 5`
- **Test Case 5:**
  Input: `4 (Division), Number1 = 10, Number2 = 0`
  Expected Output:
  `Division by zero is not allowed.`