Problem 1: Fibonacci Number Series

A company is developing a tool to generate terms of the Fibonacci Number Series, defined as:

- F(0) = 0
- F(1) = 1
- F(n) = F(n-1) + F(n-2) for n > 1

Task: Write a Java program that:

1. Accepts a positive integer `num` as input from the user.
2. Displays the `num`th Fibonacci number.
3. Displays all Fibonacci numbers up to the `num`th term.
4. Handles invalid inputs by prompting the user for a valid positive number.

Problem 2: Factorial Sequence

A company is creating a tool for calculating **factorial values** defined as:

- F(0) = 1
- F(n) = n * F(n-1) for n > 0

Task: Write a Java program that:

1. Accepts a positive integer `num` as input from the user.
2. Displays the factorial of `num`.
3. Displays all factorial values from F(0) to F(num).
4. Handles invalid inputs appropriately.

Problem 3: Sum of Multiples of 6 or 10

Given a positive number y, find the sum of all multiples of 6 or 10 that are less than or equal to y. For example, if y=30, the multiples of 6 or 10 that are less than or equal to 30 are 6, 10, 12, 18, 20, 24, and 30. The sum of these multiples is 120.

Problem 4: Subtract One from Each Digit of a 4-Digit Number

Write a program to subtract one from each digit of a 4-digit number. If the digit is 0, it should wrap around to` 9.

Example:

- **Input:** `3456`
- **Output:** `2345`

Additional Requirements:

- The input must always be a valid 4-digit number (i.e., between 1000 and 9999).
- The program should handle the wrap-around for 0 correctly. For instance, if the input is `1000`, the output should be `9009`

Problem 5: Maximum Product Subarray

Given an integer array `nums`, find the contiguous subarray (containing at least one number) which has the largest product and return that product.

Task: Write a program that:

1. Accepts an integer array as input.
2. Calculates and returns the maximum product of any contiguous subarray.
3. Handles edge cases, such as arrays with negative numbers and zeros.

Problem 6: Maximum Sum of K Consecutive Elements

Given an integer array `nums` and an integer `k`, find the maximum sum of any contiguous subarray of size `k`.

Task: Write a program that:

1. Accepts an integer array and an integer `k` as input.
2. Calculates and returns the maximum sum of any contiguous subarray of size `k`.
3. Handles edge cases, such as when `k` is larger than the size of the array.

Here are three different problems related to arrays and sums:

Problem 7: Largest Sum of Non-Adjacent Elements

Given an integer array `nums`, find the maximum sum of non-adjacent elements (elements that are not next to each other in the array).

Task: Write a program that:

1. Accepts an integer array as input.
2. Calculates the maximum sum of non-adjacent elements.
3. Outputs the maximum sum and the elements contributing to this sum.

Problem 8: Find the Subarray with Given Sum

Given an integer array `nums` and an integer `target`, find the contiguous subarray that sums up to `target`. If such a subarray exists, return its start and end indices.

Task: Write a program that:

1. Accepts an integer array and an integer `target` as input.
2. Searches for a contiguous subarray whose sum equals `target`.
3. Returns the starting and ending indices of the subarray if found; otherwise, indicate that no such subarray exists.