1. **Array Assignments:**
   Create a Java program that performs various operations on arrays, such as finding the maximum element, calculating the sum and average, and sorting the elements in ascending order.
   Tasks:
   a. Implement a method to find the maximum element in an array.
   b. Implement a method to calculate the sum of all elements in an array.
   c. Implement a method to calculate the average of all elements in an array.
   d. Implement a method to sort the elements of an array in ascending order using any sorting algorithm of your choice (e.g., bubble sort, selection sort).
   e. Write a main program to demonstrate these operations:
   f. Create an array of integers and initialize it with some values.
   g. Call the methods implemented in steps 1 to 4 to perform the respective operations.
   h. Display the results of each operation.

2. **Matrix Operations Assignment:**
   Create a Java program that performs various operations on matrices using 2D arrays.
   Tasks:
   a. Implement methods to add, subtract, and multiply two matrices.
   b. Write a method to transpose a matrix.
   c. Implement scalar multiplication of a matrix.
   d. Create a method to check if a matrix is symmetric.
   e. Write a main program to demonstrate these operations by taking input matrices from the user.
   f. Display the results of each operation.

3. **Bank Account Management System**
   Create a simple Bank Account Management System that allows users to perform basic operations such as depositing money, withdrawing money, and checking their account balance.
   Tasks:
   a. Define a class named BankAccount with the following attributes:
      i. accountNumber: Unique identifier for each bank account.
      ii. accountHolderName: Name of the account holder.
      iii. balance: Current balance in the account.
   b. Implement appropriate constructors and methods for the BankAccount class:
      i. Include methods to deposit money into the account, withdraw money from the account, and check the account balance.
      ii. Ensure that the balance cannot be negative after a withdrawal.
   c. Create a main program to interact with the BankAccount class:
      i. Allow users to perform operations such as depositing money, withdrawing money, and checking their account balance using a simple menu-driven interface.
      ii. Prompt users to enter the necessary information for each operation (e.g., account number, account holder name, amount to deposit/withdraw).
      iii. Display appropriate messages to indicate the success or failure of each operation.