Question1:

Write a C++ program to manage student information for a university system. The program should perform the following tasks:

1. Define a student class with the following attributes:
    - id (integer): Student ID.
    - name (string): Student's name.
    - dob (string): Student's date of birth.
    - address (string): Student's address.
    - gpa[8] (float array): GPA for up to 8 semesters.
    - cgpa (float): Cumulative GPA (CGPA).
    - count (static int): To keep track of the number of student objects created.
2. Implement the following methods in the student class:
    - **Constructor** to initialize GPA values to 3.0 and increment the object count.
    - **Copy Constructor** to create a duplicate of a student object.
    - setter(): To input student details (id, name, dob, and address).
    - Cgpa(): To calculate and return the CGPA based on entered semester GPAs.
    - getter(): To display the student's details and CGPA.
    - scholarship(): To determine if the student is eligible for a scholarship (CGPA ≥ 3.0).
    - NumberOfObject(): A static method to return the total number of student objects created.
3. In the main() function:
    - Prompt the user to enter the number of students.
    - Dynamically create an array of student objects.
    - For each student, input their details, calculate CGPA, display the information, and check scholarship eligibility.
    - Demonstrate the copy constructor by copying the second student into a new object and displaying its details.
    - Display the total number of student objects created.

Question2:

Write a C++ program to perform various mathematical operations using operator overloading. Implement a class MathOperation that supports arithmetic and comparison operations. The program should fulfill the following requirements:

1. **Class Definition:**
   Define a class MathOperation with the following:
   - A private integer data member x.
   - A default constructor that initializes x to 0.
   - A parameterized constructor to initialize x with a user-provided value.
   - A set() method to input the value of x (prompting with characters 'a', 'b', 'c', etc.).
   - A show() method to display the value of x.
2. **Operator Overloading:**
   Overload the following operators:
   - **Unary Operators:**
     - ++ (Prefix and Postfix): Increment the value of x.
     - (Prefix and Postfix): Decrement the value of x.
   - **Compound Assignment Operators:**
     - +=: Add 5 to the value of x.
     - -=: Subtract 5 from the value of x.
   - **Binary Arithmetic Operators:**
     - +: Add two MathOperation objects.
     - -: Subtract one MathOperation object from another.
     - *: Multiply two MathOperation objects.
     - /: Divide one MathOperation object by another.
   - **Comparison Operators:**
     - >: Return the greater of two MathOperation objects.
     - <: Return the smaller of two MathOperation objects.
3. **Main Function:**
   - Create three MathOperation objects: a, b, and c.
   - Input values for these objects using the set() method.
   - Perform operations and display the result.

Question3:

Write a C++ program to manage a book inventory system with the following features:

1. **Book Class:**
   - Attributes: author, title, publisher, price, and stock.

- Methods:
    - get_data(): Input book details.
    - display_data(): Display book details.
    - search(char[], char[]): Search for a book by author and title.
    - copies(int): Check stock and calculate the price for purchasing copies.
    - transaction(): Display successful and unsuccessful transaction counts.

2. **Menu-Driven Program:**
    - **Create**: Add multiple books to the inventory.
    - **Buy Book**: Search for a book and purchase copies if available.
    - **Transaction**: View successful and unsuccessful transactions.
    - **Display**: Show all books in the inventory.
3. **Loop:**
    - Repeat the menu until the user decides to exit.