Project Overview:

This C++ program simulates a movie ticket booking system at SM Cinema Legazpi, where users can interact with various features like booking seats, viewing movies by showtimes, and processing priority requests. The system uses heaps (Max-Heap and Min-Heap) and a binary tree to manage seat bookings and movie scheduling efficiently. Users can also interact with a priority system where senior citizens or other priority customers are processed first based on their age.

How to Run the Code:

1. **Compilation:**
    - Save the code in a file (e.g., cinema_booking.cpp).
    - Compile the program using a C++ compiler, for example:
    - g++ cinema_booking.cpp -o cinema_booking
2. **Execution:**
    - Run the compiled program:
    - ./cinema_booking
3. **Interacting with the Program:**
    - Once the program starts, you will be greeted with the main menu. You can interact with the menu by selecting options, such as viewing available movies, booking seats, and processing priority requests.

Description of Each Functionality:

1. **Movies by Showtimes:**
    - This option displays the list of movies available for booking, sorted by their showtimes. Movies are stored in a Min-Heap, ensuring that the earliest showtime is prioritized.
2. **Choose Movie:**
    - Allows users to choose a movie by its name. The movie's showtime is displayed in the format of hours and minutes (AM/PM).
3. **View Booked Seats:**
    - Displays the seats that have been booked, using three types of traversals (Pre-order, In-order, and Post-order) on a binary tree structure. The tree is used to manage seat bookings.
4. **Book Seat:**
    - Users can book a seat by entering a seat number between 1 and 10. The seat is inserted into the binary tree to maintain a balanced booking structure.
5. **Cancel Booking:**
    - This functionality could be implemented for canceling a seat booking (though it’s not currently available in the provided code).
6. **Add Priority Request:**
    - Adds a customer to a Max-Heap with their name and age. This heap structure ensures that priority requests are handled based on age, with senior citizens or older customers being processed first.

7. **Process Next Priority Request:**
    - Processes the next priority request from the heap. The customer with the highest priority (oldest age) is processed first.
8. **View Priority Requests:**
    - Displays the list of pending priority requests, showing customer names and ages in the order they will be processed based on their priority.
9. **Exit:**
    - Exits the program, thanking the user for visiting the cinema.

Key Data Structures:

1. **Binary Tree (SeatTree):**
    - This tree is used to manage seat bookings. It ensures that each seat booking is placed in the correct position based on seat number.
2. **Max-Heap (PriorityHeap):**
    - A Max-Heap is used to manage priority requests based on the customer's age. This ensures that older customers are processed first.
3. **Min-Heap (MovieHeap):**
    - A Min-Heap is used to manage the movies by their showtimes. This structure allows for efficient retrieval of the movie with the earliest showtime.

Program Execution Example:

When you run the program, it will display the main menu, and you can interact with it. For example, you could select option 1 to view movies by their showtimes, which will display:

```
Movies by Showtimes:

 Hello, Love, Again - 1:00 PM

 Moana 2 - 2:30 PM

 The Wicked - 5:00 PM
```

You can then select option 2 to choose a movie, and if you select a valid movie, you'll be shown the details of the movie you selected. Additionally, you can view and book seats, add priority requests, and process them as needed.

This system provides a simple but functional simulation of a cinema booking system with priority management and movie scheduling, using heaps and binary trees to ensure efficiency and clarity.