1. **Node Structure:**
   * Defines a structure **node** to represent a seat with attributes **seatr** (row), **seatc** (column), **status** (booking status), **next**, and **prev** pointers for doubly linked list.
2. **ticket Class:**
   * Represents a ticket management system.
   * Initializes the seat matrix using a constructor:
     + Initializes **head** and **last** arrays to maintain the beginning and end of each row.
     + Creates nodes for each seat and links them in a doubly linked list.
3. **create\_node Function:**
   * Creates a new node with the specified row, column, and initial status ("A" for available).
   * Returns a pointer to the newly created node.
4. **book Function:**
   * Allows the user to book a seat by specifying the row and column.
   * Checks if the seat is available (status is "A") and updates the status to "B" (booked).
   * Displays the updated seat matrix.
5. **cancel Function:**
   * Allows the user to cancel a booked seat by specifying the row and column.
   * Checks if the seat is booked (status is "B") and updates the status to "A" (available).
   * Displays the updated seat matrix.
6. **display Function:**
   * Displays the current status of all seats in the theater.
   * Iterates through the doubly linked list and prints the row, column, and status of each seat.
7. **main Function:**
   * Creates an instance of the **ticket** class.
   * Displays the initial seat matrix.
   * Provides a menu to the user with options to book a ticket, cancel booking, or exit.
   * Loops until the user chooses to exit.
8. **User Interaction:**
   * The user can choose to book a ticket by entering the row and column, cancel a booking, or exit the program.
9. **Note:**
   * The code uses a doubly linked list to represent the seats in each row.
   * Each row is a circular doubly linked list, where the **next** pointer of the last node points to the first node, and the **prev** pointer of the first node points to the last node.
   * The seat status is represented by the string "A" for available and "B" for booked.

Algorithm:

1. \*Include necessary libraries and define a structure for the doubly linked list node.\*

2. \*\*Create a class ticket to encapsulate the functionality of the ticket booking system.\*\*

3. \*\*Implement the member functions create\_node, book, cancel, and display within the ticket class.\*\*

4. \*\*In the main function, create an instance of the ticket class and display the initial seating arrangement.\*\*