Class assignment
Queue

In this assignment, you will implement a simple Movie Theater Ticket Booking System using C++. The system will allow users to book tickets, cancel reservations, check ticket availability, display the current reservations, and perform other operations using a **queue-based data structure**.

Requirements:

- **Global Variables:** Initialize the `data`, `front` and `rear` as global variables as follows:

```
const int MAX_SIZE = 100;
string data[MAX_SIZE];
int front = 0;
int rear = -1;
```

- **Function: `enqueue`**
  - Description: Implement the `enqueue` function that allows users to book a ticket by adding a customer's name to the queue.
  - Function Signature: `void enqueue(string customerName);`

- **Function: `dequeue`**
  - Description: Implement the `dequeue` function that processes the reservation at the front of the queue.
  - Function Signature: `void dequeue();`

- **Function: `display`**
  - Description: Implement the `display` function that displays the current reservations in the queue.
  - Function Signature: `void display();`

- **Function: `cancelReservation`**
  - Description: Implement the `cancelReservation` function that allows users to cancel a reservation by providing the customer's name.
  - Function Signature: `void cancelReservation(string customerName);`

- **Function: `checkAvailability`**
  - Description: Implement the `checkAvailability` function that checks if a reservation exists for a customer by providing the customer's name. It should return `true` if the reservation exists, otherwise `false`.

- Function Signature: `bool checkAvailability(string customerName);`

- **Function: `clearQueue`**
  - Description: Implement the `clearQueue` function that clears all reservations from the queue.
  - Function Signature: `void clearQueue();`

- **Function: `getQueueSize`**
  - Description: Implement the `getQueueSize` function that returns the current number of reservations in the queue.
  - Function Signature: `int getQueueSize();`

- **Function: `peekFront`**
  - Description: Implement the `peekFront` function that allows users to see the name of the customer at the front of the queue without removing the reservation.
  - Function Signature: `void peekFront();`

- **Function: `peekRear`**
  - Description: Implement the `peekRear` function that allows users to see the name of the customer at the rear of the queue without removing the reservation.
  - Function Signature: `void peekRear();`

Assignment Tasks:

1. Implement all the functions listed above based on the provided code template.
2. In the `main` function, create a menu-driven interface that allows users to interact with the Movie Theater Ticket Booking System. The menu should include options for booking tickets, canceling reservations, checking availability, displaying reservations, clearing all reservations, getting the queue size, and peeking at the front and rear of the queue.
3. Test your program by simulating various scenarios to ensure that the ticket booking system functions correctly.

N.B: Once you are done with the array-based queue, you need to implement the queue using a linked list.

Sample output:

```
Welcome to the Movie Theater Ticket Booking System!

1. Book a ticket
2. Cancel a reservation
3. Display reservations
4. Check availability
5. Clear all reservations
6. Get queue size
7. Peek at front
8. Peek at rear
9. Exit
Enter your choice: 1
Enter customer name: Alice
Reservation for Alice added to the queue.

1. Book a ticket
2. Cancel a reservation
3. Display reservations
4. Check availability
5. Clear all reservations
6. Get queue size
```

```
7. Peek at front
8. Peek at rear
9. Exit
Enter your choice: 1
Enter customer name: Bob
Reservation for Bob added to the queue.

1. Book a ticket
2. Cancel a reservation
3. Display reservations
4. Check availability
5. Clear all reservations
6. Get queue size
7. Peek at front
8. Peek at rear
9. Exit
Enter your choice: 3
Current reservations:
1. Alice
2. Bob

1. Book a ticket
2. Cancel a reservation
```

```
3. Display reservations
4. Check availability
5. Clear all reservations
6. Get queue size
7. Peek at front
8. Peek at rear
9. Exit
Enter your choice: 4
Enter customer name to check availability: Alice
Reservation for Alice exists.

1. Book a ticket
2. Cancel a reservation
3. Display reservations
4. Check availability
5. Clear all reservations
6. Get queue size
7. Peek at front
8. Peek at rear
9. Exit
Enter your choice: 2
Enter customer name to cancel: Bob
Reservation for Bob canceled.
```

```
1. Book a ticket
2. Cancel a reservation
3. Display reservations
4. Check availability
5. Clear all reservations
6. Get queue size
7. Peek at front
8. Peek at rear
9. Exit
Enter your choice: 5
All reservations cleared.

1. Book a ticket
2. Cancel a reservation
3. Display reservations
4. Check availability
5. Clear all reservations
6. Get queue size
7. Peek at front
8. Peek at rear
9. Exit
Enter your choice: 6
```

```
Queue size: 0 reservations.

1. Book a ticket
2. Cancel a reservation
3. Display reservations
4. Check availability
5. Clear all reservations
6. Get queue size
7. Peek at front
8. Peek at rear
9. Exit
Enter your choice: 7
Queue is empty. No reservations at the front.

1. Book a ticket
2. Cancel a reservation
3. Display reservations
4. Check availability
5. Clear all reservations
6. Get queue size
7. Peek at front
8. Peek at rear
9. Exit
```

```
Enter your choice: 8
Queue is empty. No reservations at the rear.

1. Book a ticket
2. Cancel a reservation
3. Display reservations
4. Check availability
5. Clear all reservations
6. Get queue size
7. Peek at front
8. Peek at rear
9. Exit
Enter your choice: 9
Exiting the program. Thank you!
```