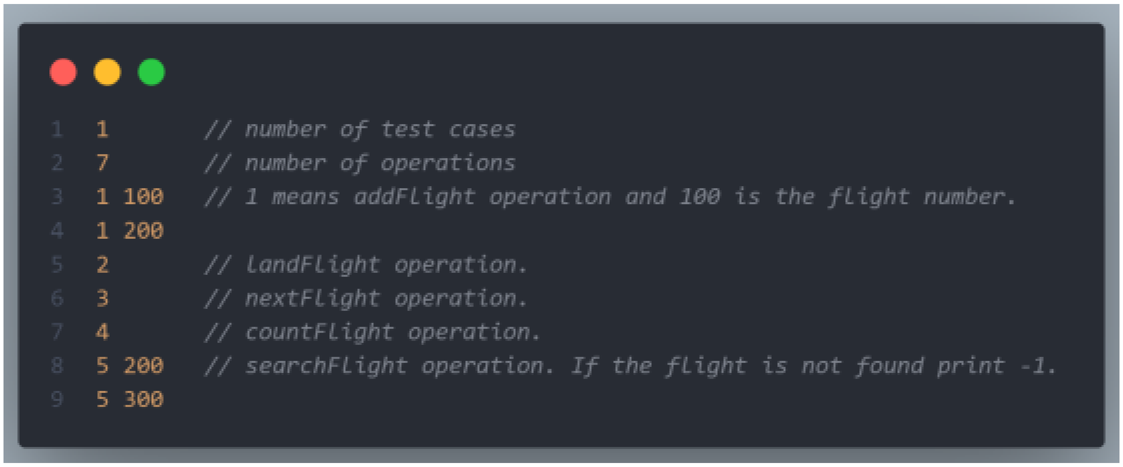
Question 1:

You are working on a simulation project for an air traffic control system. The system needs to keep track of flights and their landing requests. Design a linked list data structure to implement the landing queue for incoming flights. Your linked list should have the following capabilities:

1. Add a flight's landing request at the end of the queue. (addFlight)
2. Grant permission for landing to the next flight and remove it from the queue. (landFlight)
3. Display the fight details at the front of the queue without removing it. (nextFlight)
4. Display the number of flights waiting in the queue. (countFlights)
5. Search for a specific flight in the queue and display its position. (searchFlight)

Example Input:



```
1       // number of test cases
7       // number of operations
1 100   // 1 means addFlight operation and 100 is the flight number.
1 200
2       // landFlight operation.
3       // nextFlight operation.
4       // countFlight operation.
5 200   // searchFlight operation. If the flight is not found print -1.
5 300
```

Output:
200 // 100 & 200 were added. 100 landed so the next flight is 200.
1 // The total number of flights in the queue at the moment is 1.
1 // The position of flight 200 is 1.
-1 // Flight 300 is not in the queue.

The operation 1, & 2 do not generate any output.

Question 2:

Design a doubly linked list and demonstrate the following operations:

1. Insertion at any given X position.
2. Given two numbers A and B, determine the number of elements existing between the two of them.
3. Deletion of any given element X.

Question 3:

Convert the above doubly linked list into a circular linked list (where the last element links to the first element in the list), then change your insertion function to be able to handle insertion at any given position in a circular linked list. By default, the newly added element should be assigned as the “first” element of the circular linked list.

Question 4 (bonus):

Implement Floyd’s algorithm to determine and confirm whether the above linked list is indeed circular.