**Practical No: 8(C)**

**Practical Title :** The ticket booking system of Cinemax theater has to be implemented using C++ program. There are 10 rows and 7 seats in each row. Doubly circular linked list has to be maintained to keep track of free seats at rows. Assume some random booking to start with. Use array to store pointers (Head pointer) to each row. On demand

* 1. The list of available seats is to be displayed
  2. The seats are to be booked
  3. The booking can be cancelled.

**Pre-requisite:**

* Knowledge of Doubly Circular Linked List
* Representation of Circular Linked list
* Knowledge of ticket booking

**Objective:**

* To perform Doubly Circular linked list for Cinemax ticket booking.
* To display available seats.
* To book and cancel seats

**Input:**

Row no and seat no to book seat

Outcome:

* Display available seats to book movie ticket.
* Display status of Booked seat/ cancel seat.

**Theory:**

Theory :

The linked lists that we have seen so far are known as linear linked lists. All elements of such a linked list can be accessed by fi rst setting up a pointer pointing to the first node in the list and then traversing the entire list. Although a linear linked list is a useful data structure, it has some drawbacks. For example, consider an SLL. Given a pointer Current to a node in an SLL,we cannot reach any of the nodes that precede the Current node

(+++this is not the case with DLL as DLL has two, one backward and one forward, links). This drawback can be overcome by making a small change, and this change is without any additional data structure. In a singly linear list, the last nodes link field is set to Null.Instead of that, store the address of the first node of the list in that link field. This change will make the last node point to the first node of the list. Such a linked list is called circular linked list , shown in Fig.

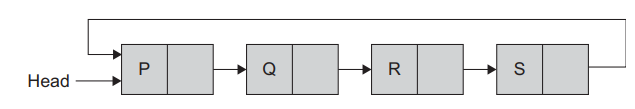


Fig. Circular linked list

From any node in such a list, it is possible to reach to any other node in the list. We need not traverse the list again right from the first node. Circular linked list is used in many applications. Circular linked list is used to keep track of free space (unused nodes) in memory. In a circular list, traversal can be continued from current node. It helps us to keep the traversal procedure an unending one. The two primary applications of circular list is time slicing and memory management. We can have a circular SLL or DLL. Both alternatives are possible. Similarly, circular linked lists could be with or without header nodes.

**Doubly Circular Link List:**

In doubly circular linked list, the last node’s next link is set to the first node of the list and the first node’s previous link is set to the last node of the list. This gives access to the last node directly from the first node

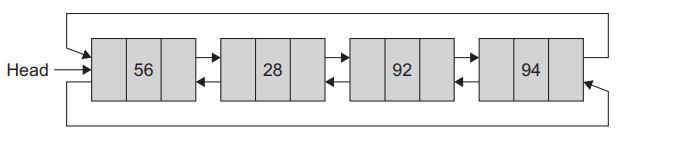


Fig. Doubly circular list

Figure represents the doubly circular linked list without a header node. Figure is the doubly circular linked list with header node. Header node may store some relevant information of the list.

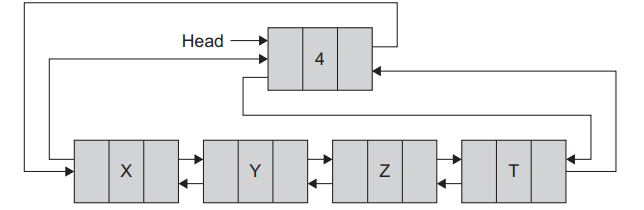
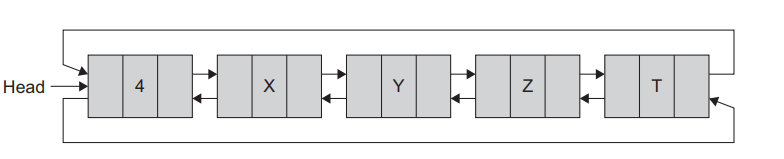


Fig. Headed doubly circular list

The operations on circular linked list—insert, delete, create and traverse—follow the same method as that of linear list except for a few changes. We can redraw the circular list with header node as in



* Doubly Circular linked list as an ADT : (write pseudo code for each operation)
* Algorithms :(Write your own algorithms for your program)
* Flowchart :(draw flowchart for above algorithms)

**Conclusion:**

By this way, we can book or cancel movie ticket using doubly Circular linked lists.

**Question Bank:**

1. What is Doubly Circular Linked List?

2. How to represent doubly circular linked list?

3. How to book or cancel seat?

3. What is doubly linked list?

4. How to insert and delete elements from doubly circulars linked list?