21143 Tanmay Karmarkar

Doubly Circular Linked List

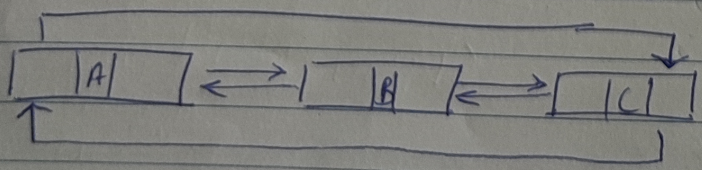
Problem statement:
The ticket booking system of a ~~stt~~ cinemax has to be implemented using C++ program. There are 10 rows & 7 columns.
On demand:
1. List of available seats is to be displayed.
2. Seats are to be sorted.
3. Booking can be cancelled.

Objective: To
implement doubly circular linked list.

Outcomes: Students will be able to
Write & execute C++ program to book tickets using concept of DLL.

SW & HW:
VS CODE, mingw compiler, Windows 10, 8 GB RAM, 512 GB SSD.

Theory: Linked list is an example of dynamic data structure.
They can grow & shrink during execution.
Insertion & deletion are easier & faster.



Each node has 3 parts: 1) Head 2) Tail 3) Data.

Pseudocode

```
class node
{ node * next, * prev ;
  int seat, status;
  string id ;}
class cinemax
{ node* head, *tail, *temp ;
void creat ()
{
 int i = 1 ;
temp = new Node ;
temp → seat = 1
 temp → status = 0
temp → id = "null"
tail = head = temp ;
for loop from 2 to 70 to create node.
node* p ;
p → status = 0 ;
p → id = "null" ;
tail → next = p ;
p → prev = tail ;
tail → next = head ;
head → prev = tail ;
```

void display ()
{ Similar to create
Iterate from 1 to 70 till
temp → next != head.
print -E- if status = 0
print -B- if status = 1
}
Void Book ()
{ cout << "Enter seat to be booked";
cin >> x;
traverse in linked list
until temp → seat = x;
change temp → status = 1;
print (seat booked)

void cancel ()
{ cout << "Enter seat no to be cancelled";
cin >> x
Similar to book.
change temp → status = 0;
if tem → status = 0; cout << "Seat is not booked";
else : print (cancelled)

```
main ()
{ cinemax obj;
  obj.create();
}
```

Menu Code:

```
do{ 
 cout << "*** Menu *** ";
 cout << "\n1. current status \n2. Books eat 3. cancel 4. exit.
 cout << "Enter choice";
 cin >> ch;
 switch (ch)
{ case 1: obj.display();
   case 2; obj.book();
    case 3; obj.cancel;
}while ch != 4)
 ret 0;
}.
```