**Assignment B15**

/\*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

a) The list of available seats is to be displayed

b) The seats are to be booked

c) The booking can be cancelled.\*/

===========================================================================

**#include**<iostream>

**using** **namespace** std;

**struct** node

{

**int** seatc,seatr;

string status;

**struct** node \*next ,\*prev;

}\*head[10],\*last[10];

**class** ticket

{

**public**:

**ticket**()

{

**for**(**int** i=1 ; i<=10 ; i++)

{

head[i]=last[i]=NULL;

**struct** node\* temp;

**for**(**int** j=1 ; j<=7 ; j++)

{

temp=create\_node(i,j);

**if**(head[i]==last[i] && head[i]==NULL)

{

head[i]=last[i]=temp;

head[i]->next=last[i]->next=NULL;

head[i]->prev=last[i]->prev=NULL;

}

**else //Insertion at beginning**

{

temp->next=head[i];

head[i]->prev=temp;

head[i]=temp;

head[i]->prev=last[i];

last[i]->next=head[i];

}

}

}

}

node\* **create\_node**(**int** x,**int** y)

{

**struct** node\*temp;

temp=**new**(**struct** node);

**if**(temp==NULL)

{

cout<<"\nMemory not allocated";

**return** 0;

}

**else**

{

temp->seatr=x;

temp->seatc=y;

temp->status="A";

temp->next=NULL;

temp->prev=NULL;

**return** temp;

}

}

**void** **book**()

{

**int** x,y;

cout<<"\nEnter row and column";

cin>>x>>y;

**struct** node\* temp;

temp=head[x];

**for**(**int** i=1 ; i<=7 ; i++)

{

**if**(temp->seatc==y)

{

**if**(temp->status=="A")

{

temp->status="B";

}

**else**

{

cout<<"\nSORRY !! Already booked!!";

}

}

temp=temp->next;

}

display();

}

**void** **cancel**(){

**int** x,y;

cout<<"\nEnter row and column to cancel booking : ";

cin>>x>>y;

**struct** node\* temp;

temp=head[x];

**for**(**int** i=1 ; i<=7 ; i++)

{

**if**(temp->seatc==y)

{

**if**(temp->status=="B")

{

temp->status="A";

}

**else**

{

cout<<"\nSORRY !! Already unbooked!!";

}

}

temp=temp->next;

}

display();

}

**void** **display**()

{

**struct** node\* temp;

**for**(**int** j=1 ; j<=10 ; j++)

{

temp=head[j];

**for**(**int** i=1 ; i<=7 ; i++)

{

cout<<temp->seatr<<","<<temp->seatc<<temp->status<<"\t";

temp=temp->next;

}

cout<<"\n";

}

}

};

**int** **main**()

{

ticket t;

**int** ch;

t.display();

**do**{

cout<<"\n1.Book Ticket \n2.Cancel Booking \n3.EXIT";

cin>>ch;

**switch**(ch)

{

**case** 1:t.book();**break**;

**case** 2:t.cancel();**break**;

}

}**while**(ch!=3);

**return** 0;

}

===========================================================================

Output:

1,7A 1,6A 1,5A 1,4A 1,3A 1,2A 1,1A

2,7A 2,6A 2,5A 2,4A 2,3A 2,2A 2,1A

3,7A 3,6A 3,5A 3,4A 3,3A 3,2A 3,1A

4,7A 4,6A 4,5A 4,4A 4,3A 4,2A 4,1A

5,7A 5,6A 5,5A 5,4A 5,3A 5,2A 5,1A

6,7A 6,6A 6,5A 6,4A 6,3A 6,2A 6,1A

7,7A 7,6A 7,5A 7,4A 7,3A 7,2A 7,1A

8,7A 8,6A 8,5A 8,4A 8,3A 8,2A 8,1A

9,7A 9,6A 9,5A 9,4A 9,3A 9,2A 9,1A

10,7A 10,6A 10,5A 10,4A 10,3A 10,2A 10,1A

1.Book Ticket

2.Cancel Booking

3.EXIT1

Enter row and column1

6

1,7A 1,6B 1,5A 1,4A 1,3A 1,2A 1,1A

2,7A 2,6A 2,5A 2,4A 2,3A 2,2A 2,1A

3,7A 3,6A 3,5A 3,4A 3,3A 3,2A 3,1A

4,7A 4,6A 4,5A 4,4A 4,3A 4,2A 4,1A

5,7A 5,6A 5,5A 5,4A 5,3A 5,2A 5,1A

6,7A 6,6A 6,5A 6,4A 6,3A 6,2A 6,1A

7,7A 7,6A 7,5A 7,4A 7,3A 7,2A 7,1A

8,7A 8,6A 8,5A 8,4A 8,3A 8,2A 8,1A

9,7A 9,6A 9,5A 9,4A 9,3A 9,2A 9,1A

10,7A 10,6A 10,5A 10,4A 10,3A 10,2A 10,1A

1.Book Ticket

2.Cancel Booking

3.EXIT1

Enter row and column8

5

1,7A 1,6B 1,5A 1,4A 1,3A 1,2A 1,1A

2,7A 2,6A 2,5A 2,4A 2,3A 2,2A 2,1A

3,7A 3,6A 3,5A 3,4A 3,3A 3,2A 3,1A

4,7A 4,6A 4,5A 4,4A 4,3A 4,2A 4,1A

5,7A 5,6A 5,5A 5,4A 5,3A 5,2A 5,1A

6,7A 6,6A 6,5A 6,4A 6,3A 6,2A 6,1A

7,7A 7,6A 7,5A 7,4A 7,3A 7,2A 7,1A

8,7A 8,6A 8,5B 8,4A 8,3A 8,2A 8,1A

9,7A 9,6A 9,5A 9,4A 9,3A 9,2A 9,1A

10,7A 10,6A 10,5A 10,4A 10,3A 10,2A 10,1A

1.Book Ticket

2.Cancel Booking

3.EXIT2

Enter row and column to cancel booking : 8

5

1,7A 1,6B 1,5A 1,4A 1,3A 1,2A 1,1A

2,7A 2,6A 2,5A 2,4A 2,3A 2,2A 2,1A

3,7A 3,6A 3,5A 3,4A 3,3A 3,2A 3,1A

4,7A 4,6A 4,5A 4,4A 4,3A 4,2A 4,1A

5,7A 5,6A 5,5A 5,4A 5,3A 5,2A 5,1A

6,7A 6,6A 6,5A 6,4A 6,3A 6,2A 6,1A

7,7A 7,6A 7,5A 7,4A 7,3A 7,2A 7,1A

8,7A 8,6A 8,5A 8,4A 8,3A 8,2A 8,1A

9,7A 9,6A 9,5A 9,4A 9,3A 9,2A 9,1A

10,7A 10,6A 10,5A 10,4A 10,3A 10,2A 10,1A

1.Book Ticket

2.Cancel Booking

3.EXIT