SYCOA124

Assignment No :- 11

Aim :-
The ticket booking system of Cinemax theater has to be implemented using C++ program. There are 15 rows & 10 seats ~~ee~~ in each row. Doubly linked lists have to be maintained to keep track of free seats in rows. Assume some random booking to start with. Use an array to store pointers (Head pointer) to each row. On demand.
a) the list of ~~avaib~~ available seats is to be displayed.
b) The seats are to be booked.
c) The booking can be cancelled.

Theory :-
1.) Explain the working of doubly link list?
→ Doubly linked list contains a link element called first and last.
Each link carries a data field (s) & two link fields called next & prev.
Each link is linked with its next link using its next link.
Each link is linked with its previous link using its previous link.
The last link carries a link as null to mark the end of the list.

2) How doubly link list helps to fast execution than single link list?

→ If we need better performance while searching & memory is not a limitation in this case doubly linked list is more preferred. As singly linked list store pointer of only one node so consumes lesser memory. On other hand Doubly linked list uses more memory per node (two pointers).

Conclusion :-

We have implemented the concept of doubly link list along with the concept of object oriented programming.

CODE :

```
#include <iostream>
#include <iomanip>
using namespace std;

struct node
{
    int data;
    node *next, *prev;
};
class cinema_hall
{
public:
    node *h[15][11];
    cinema_hall()
    {
        int i = 0;
        for (i = 0; i < 15; i++)
        {
            h[i][0] = new node;
            h[i][0]->data = i;
            h[i][0]->next = h[i][1];
            h[i][0]->prev = NULL;
        }
        for (i = 0; i < 15; i++)
        {
            for (int j = 1; j < 11; j++)
            {
                h[i][j] = new node;
                h[i][j]->data = 0;
                if (j < 10)
                    h[i][j]->next = h[i][j + 1];
                else
                    h[i][j]->next = NULL;
                h[i][j]->prev = h[i][j - 1];
            }
        }
    }
    int check(int row, int seat)
    {
        return (h[row][seat]->data);
    }
    void book()
    {
        int row, seat, check_seat, ch;
    xyz:
        cout << "enter the row number (rows are from 0 to 14)" << endl;
        cin >> row;
```

```
        cout << "enter seat number(seat are from 1 to 10)" << endl;
        cin >> seat;
        check_seat = check(row, seat);
        if (check_seat == 0)
        {
            h[row][seat]->data = 1;
            cout << "succesfully booked" << endl;
            cout << "want to book more 1)yes" << endl;
            cin >> ch;
            if (ch == 1)
            {
                goto xyz;
            }
        }
        else
        {
            cout << "seat not available try different seat"
                 << endl;
            goto xyz;
        }
    }
    void cancel()
    {
        int row, seat, check_seat;
        cout << "enter the row number (rows are from 0 to 14)" << endl;
        cin >> row;
        cout << "enter seat number(seat are from 1 to 10)" << endl;
        cin >> seat;
        check_seat = check(row, seat);
        if (check_seat == 1)
        {
            h[row][seat]->data = 0;
            cout << "succesfully cancelled"
                 << endl;
        }
        else
        {
            cout << "seat not booked,cancelation is not possible"
                 << endl;
        }
    }
    void display()
    {
        cout << endl
             << "number 1=seat not available (already booked)" << endl
             << "       0=seat available" << endl
             << endl;
        cout << "         ";
```

```
        for (int j = 1; j < 11; j++)
        {
            cout << j << " ";
        }
        cout << endl;
        for (int i = 0; i < 15; i++)
        {
            cout << "row " << setw(2) << i << "|";
            for (int j = 1; j < 11; j++)
            {
                cout << " " << h[i][j]->data;
            }
            cout << endl;
        }
        cout << "          all eyes here" << endl;
        cout << "     ----------------------" << endl
             << "     |       PUSHPA        |" << endl
             << "     ----------------------" << endl
             << endl;
    }
};

int main()
{
    int i = 0, choice;
    cinema_hall p;
    while (1)
    {
        cout << endl;
        cout << "menu" << endl
             << "1)booking" << endl
             << "2)canceling" << endl
             << "3)display availability/occupancy" << endl
             << "4)exit" << endl
             << "enter the choice:";
        cin >> choice;
        cout << endl;
        switch (choice)
        {
        case 1:
            p.book();
            break;
        case 2:
            p.cancel();
            break;
        case 3:
            p.display();
            break;
```

```
        case 4:
            return 0;
            break;
        default:
            cout << "wrong choice" << endl;
            break;
        }
    }
    return 0;
}
```

OUTPUT:

```
menu
1)booking
2)canceling
3)display availability/occupancy
4)exit
enter the choice:1

enter the row number (rows are from 0 to 14)
0
enter seat number(seat are from 1 to 10)
1
succesfully booked
want to book more 1)yes
1
enter the row number (rows are from 0 to 14)
0
enter seat number(seat are from 1 to 10)
3
succesfully booked
want to book more 1)yes
1
enter the row number (rows are from 0 to 14)
2
enter seat number(seat are from 1 to 10)
5
succesfully booked
want to book more 1)yes
1
```

```
enter the row number (rows are from 0 to 14)
2
enter seat number(seat are from 1 to 10)
4
succesfully booked
want to book more 1)yes
0

menu
1)booking
2)canceling
3)display availability/occupancy
4)exit
enter the choice:3
```

```
  number 1=seat not available (already booked)
         0=seat available

          1 2 3 4 5 6 7 8 9 10
  row  0| 1 0 1 0 0 0 0 0 0 0
  row  1| 0 0 0 0 0 0 0 0 0 0
  row  2| 0 0 0 1 1 0 0 0 0 0
  row  3| 0 0 0 0 0 0 0 0 0 0
  row  4| 0 0 0 0 0 0 0 0 0 0
  row  5| 0 0 0 0 0 0 0 0 0 0
  row  6| 0 0 0 0 0 0 0 0 0 0
  row  7| 0 0 0 0 0 0 0 0 0 0
  row  8| 0 0 0 0 0 0 0 0 0 0
  row  9| 0 0 0 0 0 0 0 0 0 0
  row 10| 0 0 0 0 0 0 0 0 0 0
  row 11| 0 0 0 0 0 0 0 0 0 0
  row 12| 0 0 0 0 0 0 0 0 0 0
  row 13| 0 0 0 0 0 0 0 0 0 0
  row 14| 0 0 0 0 0 0 0 0 0 0
           all eyes here
       ----------------------
       |       PUSHPA       |
       ----------------------
```

```
menu
1)booking
2)canceling
3)display availability/occupancy
4)exit
enter the choice:2

enter the row number (rows are from 0 to 14)
2
enter seat number(seat are from 1 to 10)
4
succesfully cancelled

menu
1)booking
2)canceling
3)display availability/occupancy
4)exit
enter the choice:2

enter the row number (rows are from 0 to 14)
0
enter seat number(seat are from 1 to 10)
0
seat not booked,cancelation is not possible

menu
1)booking
2)canceling
3)display availability/occupancy
4)exit
enter the choice:3
```

```
enter the choice:3


number 1=seat not available (already booked)
       0=seat available

        1 2 3 4 5 6 7 8 9 10
row  0| 1 0 1 0 0 0 0 0 0 0
row  1| 0 0 0 0 0 0 0 0 0 0
row  2| 0 0 0 0 1 0 0 0 0 0
row  3| 0 0 0 0 0 0 0 0 0 0
row  4| 0 0 0 0 0 0 0 0 0 0
row  5| 0 0 0 0 0 0 0 0 0 0
row  6| 0 0 0 0 0 0 0 0 0 0
row  7| 0 0 0 0 0 0 0 0 0 0
row  8| 0 0 0 0 0 0 0 0 0 0
row  9| 0 0 0 0 0 0 0 0 0 0
row 10| 0 0 0 0 0 0 0 0 0 0
row 11| 0 0 0 0 0 0 0 0 0 0
row 12| 0 0 0 0 0 0 0 0 0 0
row 13| 0 0 0 0 0 0 0 0 0 0
row 14| 0 0 0 0 0 0 0 0 0 0
         all eyes here
     ----------------------
     |       PUSHPA       |
     ----------------------


menu
1)booking
2)canceling
3)display availability/occupancy
4)exit
enter the choice:4

PS D:\program\secondyear>
```