Page No.:
Date: / /

Assigment-07 (DSL)

Roll No. 21123 Batch: F1

Div. SE1 DOP: DOS:

TITLE: Circular link list (Doubly)

Problem statement: the ticket booking system of cinema x theater has to be implement using c++ program, there are 10 rows and 7 seat in each row.

Assume some random booking to start with use array to store pointers to each row. on demand

a) List of available seats is to be displayed.

b) Seats are to be booked

c) Booking can be cancelled.

objective: To implement doubly circular list.

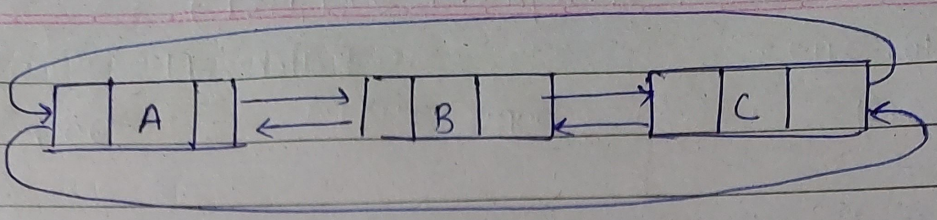
outcomes: Student will be able to write and execute c++ program to book tickets using concept of doubly circular linked list.

S/w & H/w requirement:

Windows 10, mingw compiler, Vs code, Intel i5, 8 logical processor, 8GB RAM, 512 SSD, 9th generation.

Theory: Link list is an example of dynamic data structure they can grow and shrink during the execution of program. It helps in efficient memory utilization as memory is allocated when needed

Insertion and deletion are easier and efficient for a doubly circular linked list, following is the represent



prev →

← next.

Each node in this list has 3 part : 1) Head 2) Data 3) Tail.

Algorithm / Pseudocode:

```
class Node{
    int data;
    node* next;
    node* prev;
} *(head[10]), *(last[10]);
class Ticket{
    public:
    Ticket()
    {
        For(int i=0; i<11; i++)
            for(j=0; j<8; j++)
            {
                head[i] = last[i] = NULL;

    for (int i=0; i<11; i++)
        for(int j=0; j<8; j++)
```

```
node * ptr = new node(); // Allocating new memory
if (head[i] == NULL)
then
    head[i] = ptr;
    last[i] = ptr;

    head[i] -> next = head[i];
    head[i] -> prev = head[i];
    last[i] -> next = last[i];
    last[i] -> prev = last[i];
else{
    node * temp = head[i];
    while (temp -> next != head[i])
    {
        temp = temp -> next;
    }
    temp -> next = ptr;
    ptr -> prev = temp;
    ptr -> next = head[i]
    ptr = last[i];
    head[i] -> prev = last[i]
    }
}
}

Void available_set()
{
  FOUT << "ROW/COL" << end;
    For(int i=1; j<8; i++)
      {
          cout << setw(1) i;
```

```
for(int i=0 ; i<11 ; i++)
{
    cout<<i;
    node * ptr = head[i];
    while(ptr->next != head[i])
    {
        if(ptr->data == 0)
            cout<<"0";
        else
            cout<<"1";
        ptr = ptr->next;
    }
    if (ptr->data == 0)
      cout<<"0";
    else cout<<"1";
```

```
void book seat(){
1) Read No. of rows & col
2) node* ptr = head[row]
   for if(ptr->data == 1){ cout << Ticket is booked }
      else
          ptr->data = 1
```

```
void cencel-booking()
{
    Read No. of rows & col.
    if (ptr->data == 0){ cout<< Ticket is not Booked yet}
    else (ptr->data = 0 ){ Ticket cancelaCeel}
```