|  |  |
| --- | --- |
| **Name** |  |
| **CMS ID** |  |
| **Date** |  |

**Objectives:**

After completing this Lab students will able to

1. Understand the concept and usage of Doubly linked lists.
2. Understand the concept and usage of Circular linked lists.
3. Understand the concept and usage of Doubly Circular linked lists.
4. Will learn something new that will help you in getting grip of the concepts of queue and stacks.
5. Create an unordered doubly linked list that insert nodes in ascending order, create a function to print data in forward direction and another function to print in backward direction. There should be another function that deletes the desired node. The headings of required functions are as below

void insertNode(int x, int index);

// This function should take x and index as parameters and create a node, insert x into it //and stores it after the indexth node. You should justify all the cases including the case when //index is wither negative or index exceeds the maximum length

int deleteNode(int x);

// This function should take any x as parameter, finds the node that have x value, delete it and //return its index (Node number). All the special cases should be handled. Also, if there is no //such node with data equal to x exists, your function should return -1

bool findNode(int x);

// Should return true if a node with value equal to x exists and false otherwise.

void printFwd();

//Should print the linked list in forward order

void printBwd();

//Should print the list in backward direction

1. Create a doubly link list with a single character as data, prompt user to enter the number of nodes, get input from user and check if doubly link list represent a palindrome or not. The structure/class of node is given below

struct Node

{

char data;

Node \*link;

};

The struct/class that manages linked list is

struct DoublyList

{

Node \*rear, \*head;

int count;

DoublyList(){rear = NULL; head = NULL; count = 0;}

void insertNode(char ch, int index); // gets a character as input and insert it after indexth node

bool isPalindrome(); //return true if characters stored in list represent a palindrome and false otherwise

char deleteNode(int index); // delete the indexth node and return the character in it

void destroyList();// Delete all the nodes

}

1. Create a program that takes a number of any length from the user (inside the main function), split it and stores it into nodes of a circular linked list. Suppose that user enters 57349345, the linked list should look like

In case if user enters 589, the list should be



you should implement insertNode and print functions only as implementing them will fulfil all the requirements.

1. Create a doubly linked list that has the following set of functions.

struct DoublyList

{

Node \*rear, \*head;

int count;

void insertNode(int x);

// this function should create a node, stores x in it and

// insert it as the **last** node of the doubly linked list.

int deleteNode();

// This function should always delete the **first** node of the linked list

// and return the value stored in it

void print();

// it should print the linked list

};

1. Create a doubly circular linked list that insert data from the front and always remove the first node of the data.

struct DoublyList

{

Node \*rear, \*head;

int count;

void insertNode(int x);

// this function should create a node, stores x in it and

// insert it as the **first** node of the doubly circular linked list.

int deleteNode();

// This function should always delete the **first** node of the linked list

// and return the value stored in it

void print();

// it should print the linked list

};