

***NAME : Arbab javed***

***Roll No : 053***

***Date : 03-03-2025***

***Subject : DSA( lab)***

***LAB TASKS***

**Lab 1:**

**Code Explanation**:

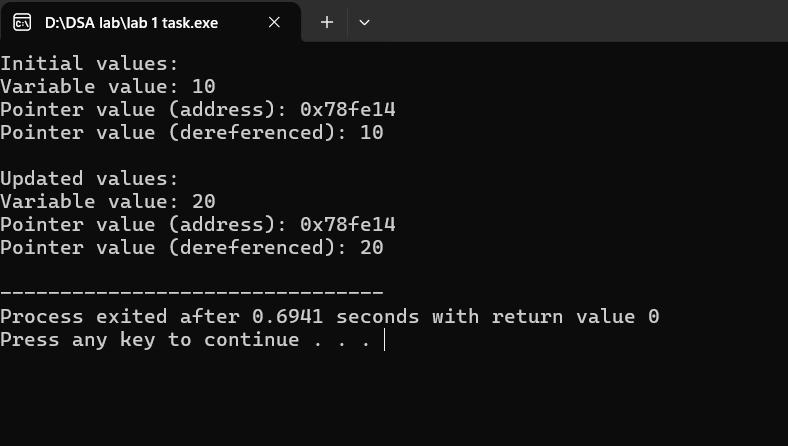
We create a Circular LinkedList class with a to store data and pointers

We implement functions to insert nodes at the first, last, N location, and center of the list.

**Why**

We use a circular linked list to easily insert and display nodes in a loop.

We use a simple do-while loop to display the list in order and reverse order.



**Lab 2:**

**What is Big O Notation?**

Big O notation measures how long an algorithm takes to complete.

**What is O (n)?**

O(n) means the algorithm takes longer as the input size (n) increases.

**Why is the max function O(n)?**

The max function checks each item in the list once, so it takes longer as the list gets bigger

If there are 10 people, it takes a short time.

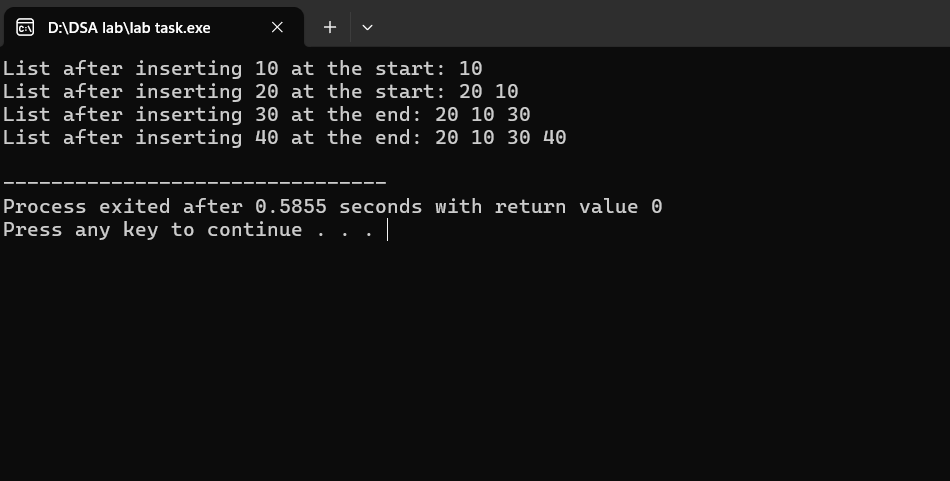
If there are 100 people, it takes longer.

If there are 1,000 people, it takes even longer.

The time it takes grows linearly with the number of people, so it's O(n)

**Lab3:**

We create a Circular LinkedList class to store and manipulate data in a loop. We implement insert functions (first, last, Nth, center) for flexibility and display functions (in order, reverse) using do-while loops for efficiency.



**Lab 4:**

**What:**

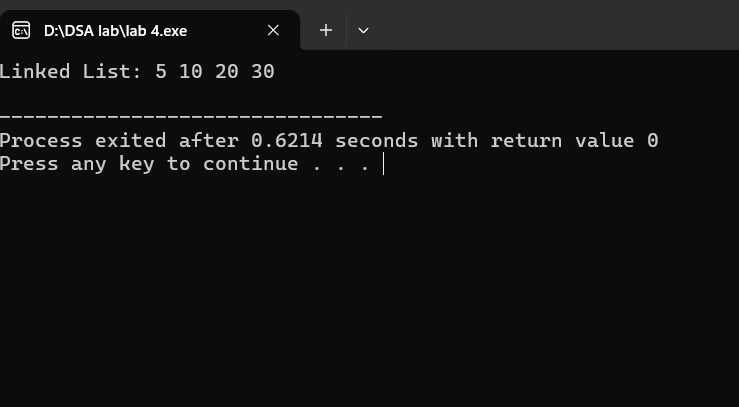
Circular Linked List implementation in C++.

**How:**

Using Node struct and Circular LinkedList class with insert and display functions.

**Why:**

Efficient data storage and nipulation in a loop, with flexible insertion and display options.



**Lab 5:**

**Code:**

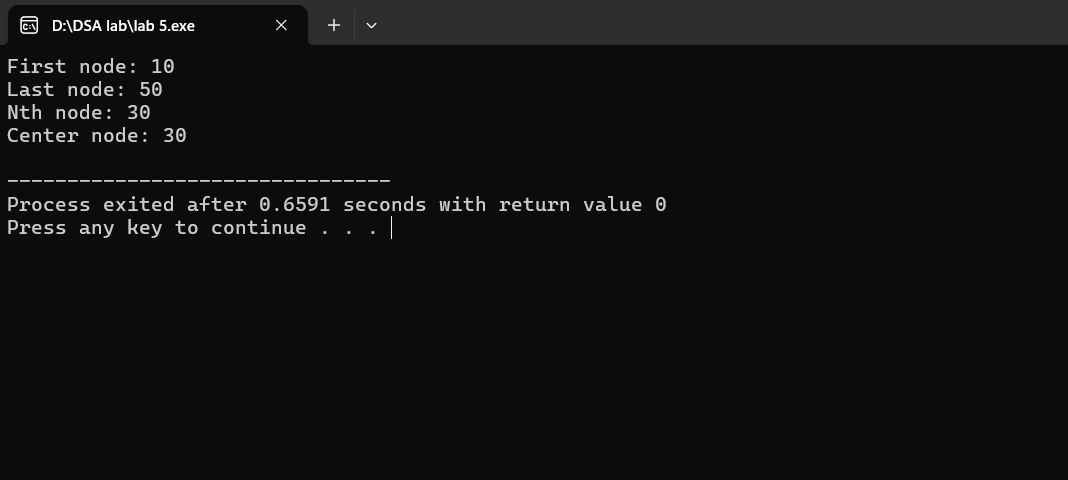
Circular Linked List in C++.

**How:**

Node struct + CircularLinkedList class with insert/display functions.

**Why:**

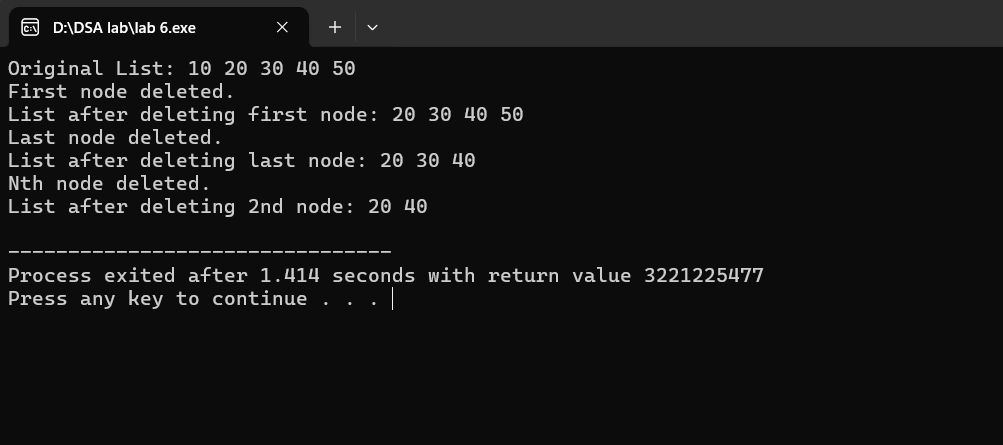
Efficient looped data storage and flexible manipulation.



**Lab 6:**

Uses Node class + Circula rLinkedList class.

Efficient looped data storage and flexible manipulation.



**Lab 7:**

**How:**

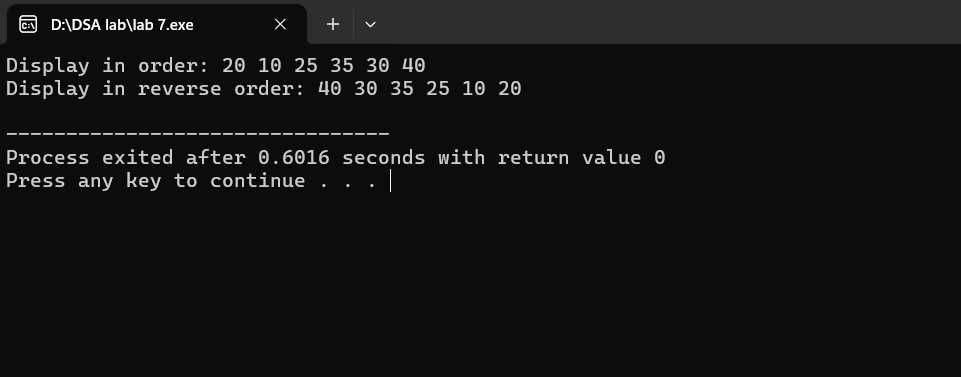
Node Class: A simple class to store data and a pointer to the next node.

Circular LinkedList Class: A class to manage the nodes and provide methods for insertion and display.

**Why:**

Efficient Data Storage: Circular Linked List allows for efficient use of memory and fast insertion/deletion of nodes.

Flexible Data Manipulation: The CircularLinkedList class provides methods for inserting nodes

****

**Lab 8:**

1. Create a Node class to store data and a pointer to the next node.

2. Create a Circular LinkedList class to manage the nodes.

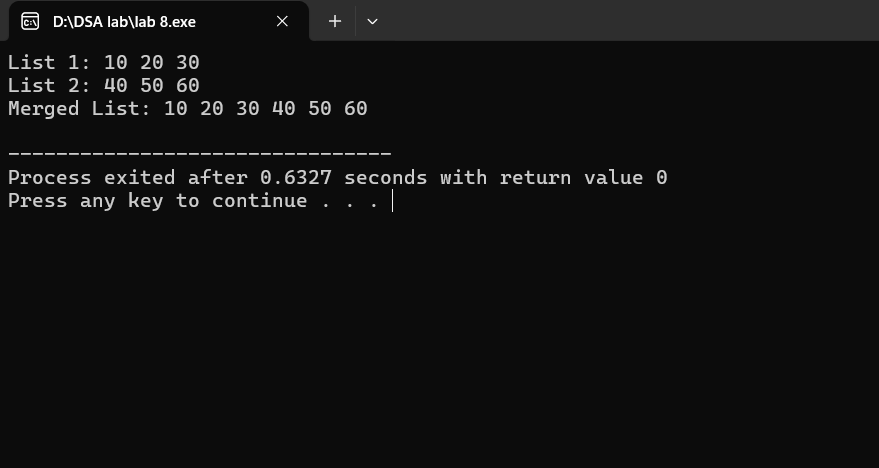
3. Implement methods for inserting nodes at different positions (first, last, Nth, center).

4. Implement methods for displaying the list in order and reverse order.

**Why:**

1 . Efficient Data Storage: Circular Linked List allows for efficient use of memory.

3. Looped Data Structure: Circular linked list allows for easy traversal of the data in a loop.

****

**Lab 9:**

**How:**

1. Create nodes to store data.

2. Connect nodes in a circle.

3. Add methods to insert and display data.

**Why:**

1. Efficient data storage.

2. Easy data manipulation.

3. Fast data access.

