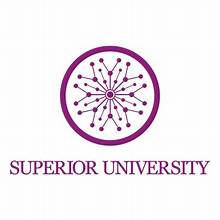
Muhammad Farhan

Roll No. SU92-BSSEM-S24-102



**SUBJECT: DSA LAB**

**ASSIGNMENT. LAB TASK**

**SUBMITTED TO: Sir Rasikh Ali**

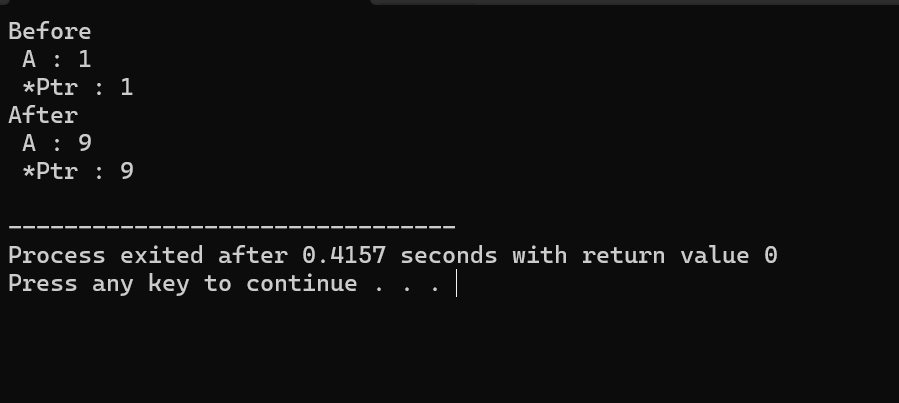
**THE SUPERIOR UNIVERSITY,GOLD CAMPUS**

**DEFENCE ROAD,LAHORE**

LAB1

Explanation

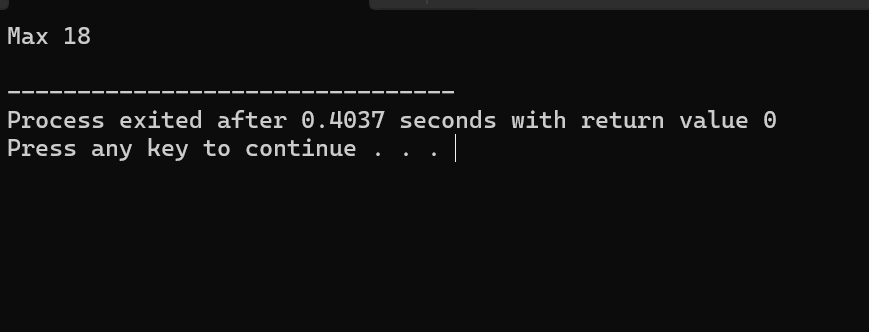
1. a is initialized to 1, and ptr points to a.
2. Before modification, both a and \*ptr (value at ptr) are 1.
3. After modifying \*ptr = 9, both a and \*ptr become 9.
4. Finally, ptr is set to nullptr, making it point to nothing.



LAB 2:

Explanation

1. findMax finds the largest number in an array.
2. It starts by assuming the first number is the biggest.
3. Then it checks the rest of the numbers and updates if a larger one is found.
4. In main, the numbers array is given to findMax, and the largest number is returned.
5. The result is 18, the biggest number in the array.



Lab 3:

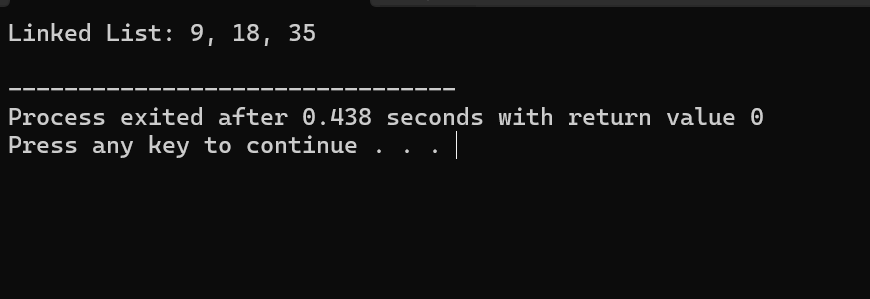
**Node class**: Holds a value and a pointer to the next node.

**LinkedList class**: Manages the list with a head node.

**addAtBeginning**: Adds a new node at the start of the list.

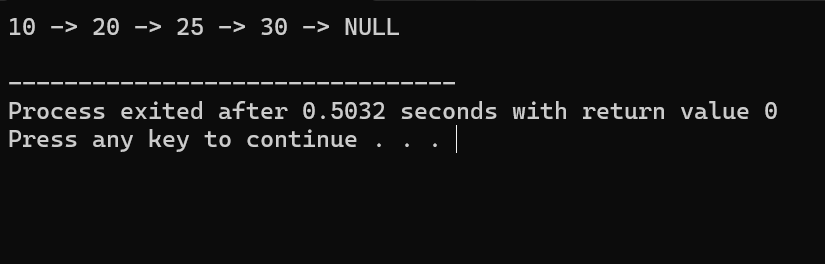
**addAtEnd**: Adds a new node at the end of the list.

**printList**: Prints the values in the list.



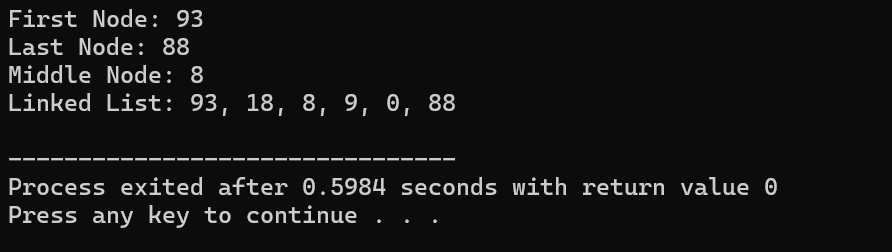
Lab 4:

1. The Linkedlist class stores numbers in a chain (linked list).
2. addfirst adds a number at the start, and addLast adds it at the end.
3. AddAt places a number at a specific position in the list.
4. Display prints all numbers in order.
5. The main function adds numbers and shows the final list.



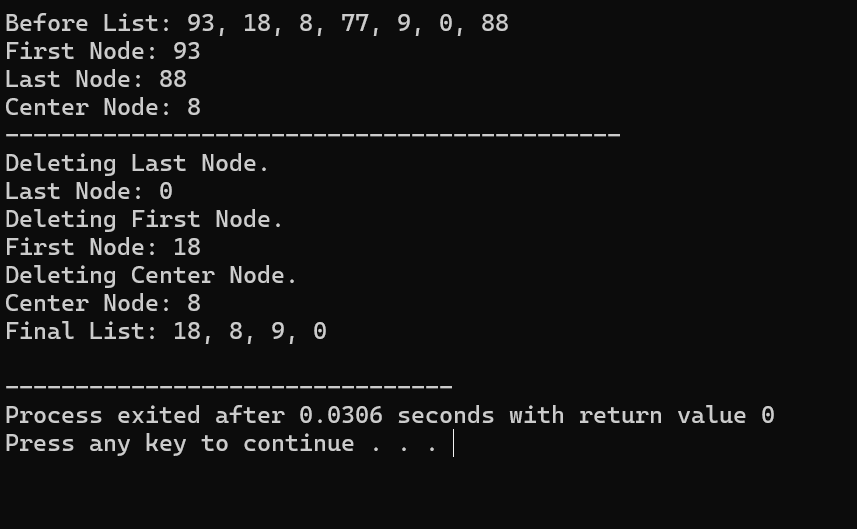
Lab 5:

1. The LinkedList class manages a list of numbers, allowing adding at the start, end, or a specific position.
2. printFirst, printLast, and printMiddle show the first, last, and middle numbers.
3. addAtPosition places a number at a given position, handling invalid positions.
4. printList displays all numbers in order.
5. The main function tests these features by adding numbers and printing the list.



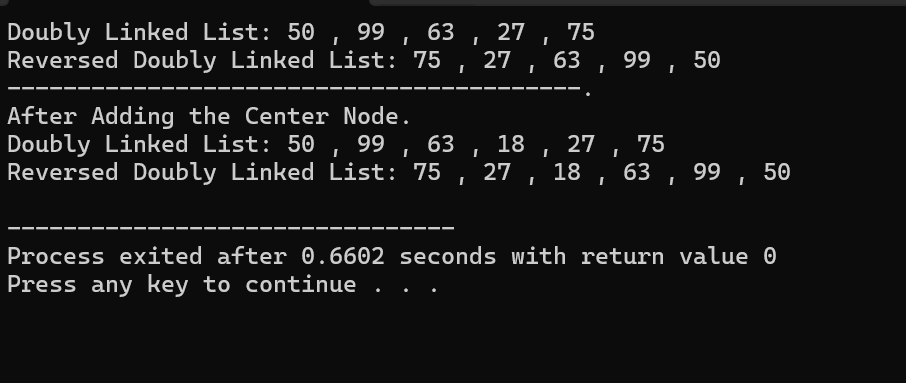
Lab6:

1. This program creates a linked list to store numbers.
2. Each number (node) points to the next one in the list.
3. You can add numbers at the start, end, or any position.
4. You can remove numbers from the start, end, middle, or by value.
5. The program can show all numbers or just the first, last, or middle one.
6. The main() function tests adding, showing, and deleting numbers.
7. Unlike arrays, a linked list grows and shrinks as needed.
8. It helps manage data easily and efficiently without wasting space.

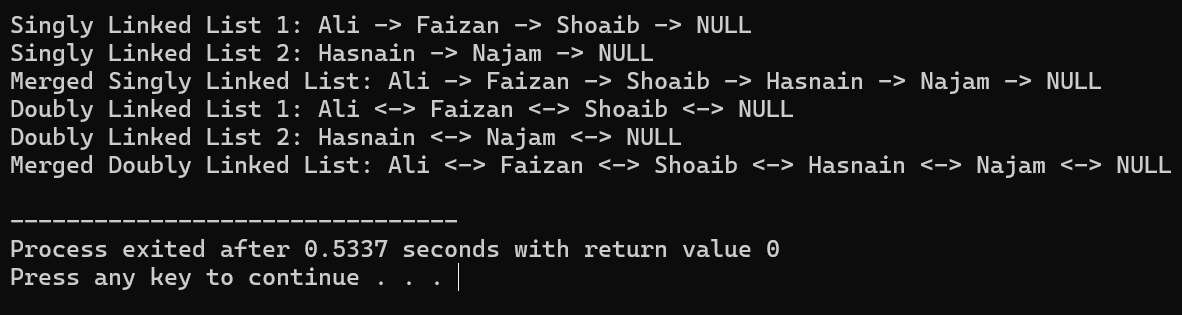


Lab 7:

1. The program creates a doubly linked list with nodes linked forward and backward.
2. Each node has data, a previous link, and a next link.
3. insertAtFirst(d) adds a new node at the beginning.
4. insertAtLast(d) adds a new node at the end.
5. insertAtCenter(d) adds a new node in the middle.
6. insertAtSpecificPosition(d, pos) adds a node at a chosen position.
7. displayInOrder() prints the list from start to end.
8. displayInReverse() prints the list from end to start.
9. The main() function adds numbers, displays the list, and adds a middle node.
10. This list allows easy adding, removing, and moving through nodes.



Lab 8:

1. Node class is created to store data and links to the next (and previous for doubly linked lists).
2. SinglyLinkedList class allows adding nodes at the end, merging two lists, and printing the list.
3. DoublyLinkedList class does the same but also supports backward navigation using a prev pointer.
4. insertLast() adds a new node at the end of the list by finding the last node and linking it.
5. mergeList() links the last node of the first list to the first node of the second list.
6. printList() goes through the list and prints all nodes in order.
7. In **main()**, two singly linked lists are created, filled with names, printed, merged, and printed again.
8. The same steps are repeated for doubly linked lists, ensuring both lists work properly
9. 

Lab 9:

1. The Node class creates a node with data and a pointer to the next node.
2. The circularList class manages the circular linked list.
3. insertAtBeginning adds a node at the start.
4. insertAtEnd adds a node at the end.
5. insertAtPosition inserts a node at a specific position.
6. display prints the list in order.
7. displayReverse prints the list in reverse.
8. The main function creates a circular linked list and performs insertions and display operations.

