**DOUBLY LINKED LIST**

A Mini Project Report Submitted by

**Adithya Holla K (4NM19CS007) Aditya Murugan(4NM19CS010)**

UNDER THE GUIDANCE OF

**MR.PUNEETH RP**

ASSISTANT PROFESSOR GD-11

Department of Computer Science and Engineering

# in partial fulfilment of the requirements for the award of the Degree of

**Bachelor of Engineering in**

**Computer Science & Engineering**

from

# **Visvesvaraya Technological University, Belagavi**



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

# N.M.A.M. INSTITUTE OF TECHNOLOGY

(An Autonomous Institution under VTU, Belagavi) (AICTE approved, NBA Accredited, ISO 9001:2008 Certified) NITTE -574 110, Udupi District, KARNATAKA.

(ISO 9001:2015 Certified), Accredited with ‘A’ Grade by NAAC

**TABLE OF CONTENTS :**

1. **ABSTRACT AND INTRODUCTION 3-4**

1. **IMPORTANCE 4**

1. **SYSTEM REQUIREMENT AND SPECIFICATION 4-5**

1. **SCREENSHOTS 6-10**

1. **CONCLUSION AND FUTURE WORK 10**

**8. REFERENCES 11**

**ABSTRACT AND INTRODUCTION**

A Doubly Linked List (DLL) contains an extra pointer, typically called previous pointer, together with next pointer and data which are there in singly linked list. Doubly linked list is a complex type of linked list in which a node contains a pointer to the previous as well as the next node in the sequence. Therefore, in a doubly linked list, a node consists of three parts: node data, pointer to the next node in sequence (next pointer) ,pointer to the previous node (previous pointer). A doubly linked list containing three nodes. Memory Representation of a doubly linked list is shown in the following image. Generally, doubly linked list consumes more space for every node and therefore, causes more expansive basic operations such as insertion and deletion. However, we can easily manipulate the elements of the list since the list maintains pointers in both the directions. Doubly Linked List is a variation of Linked list in which navigation is possible in both ways, either forward and backward easily as compared to Single Linked List. Following are the important terms to understand the concept of doubly linked list.

Some of the OpenGl functions we have used are :

**OpenGL Utility Library (GLU):** is a computer graphics library for OpenGL. It consists of a number of functions that use the base OpenGL library to provide higher-level drawing routines from the more primitive routines that OpenGL provides. It is usually distributed with the base OpenGL package. When using GLU library you will need to use #include in your program.GLU routines begin with the prefix glu.

**OpenGL Utility Toolkit (GLUT)** :is a library of utilities for OpenGL programs, which primarily perform system-level I/O with the host operating system. Functions performed include window definition, window control, and monitoring of keyboard and mouse input. Routines for drawing a number of geometric primitives (both in solid and wireframe mode) are also provided, including cubes, spheres and the Utah teapot. GLUT also has some limited support for creating pop-up menus. Here we need to use #include in your program. All GLUT functions start with the prefix glut.

**glutKetboardFunc** Registers callback handler for keyboard event.

**Handling mouse inputs with Glut** We can register callback function to handle mouse-click and mouse-motion. glutMouseFunc Registers callback handler for mouse click.

**IMPORTANCE**

It is used in the navigation systems where front and back navigation is required.

It is used by the browser to implement backward and forward navigation of visited web pages that is a back and forward button. It is also used to represent a classic game deck of cards.

It is also used by various applications to implement undo and redo functionality.

Doubly Linked List is also used in constructing MRU/LRU (Most/least recently used) cache.

Other data structures like stacks, Hash Tables, Binary trees can also be constructed or programmed using a doubly-linked list. Also in many operating systems, the thread scheduler(the thing that chooses what process needs to run at which time) maintains a doubly-linked list of all processes running at that time

**OBJECTIVE**

Our project has objective mainly where the user can insert node by inserting front or rear . user also can delete the node by using front or rear operations. The user can also specify the node . Our project has a feature to find the maximum and minimum node. There is also feature that it show the maximum number of node

**HARDWARE / SOFTWARE REQUIREMENTS**

**Software Requirements:**

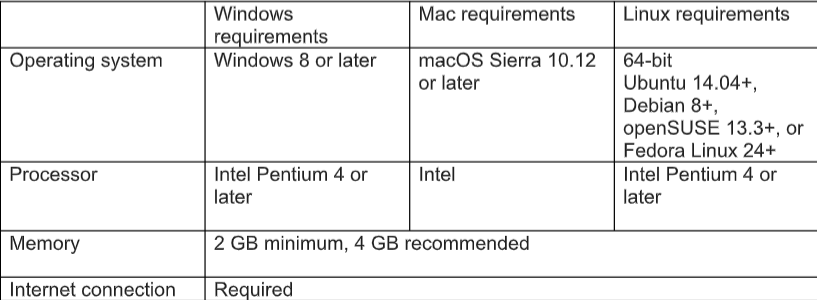
➢ IDE: Code Block

**HARDWARE SPECIFICATION:**

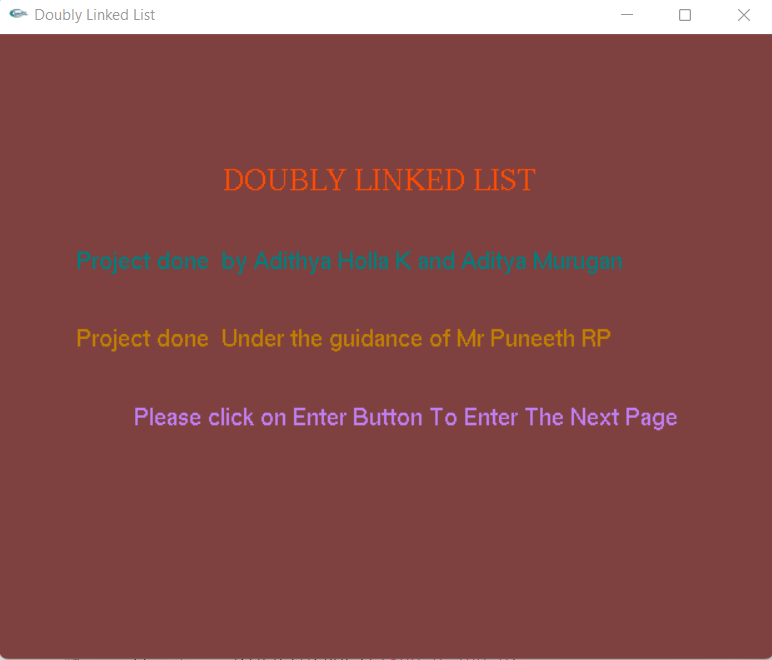
Processor: Pentium-3 or Higher

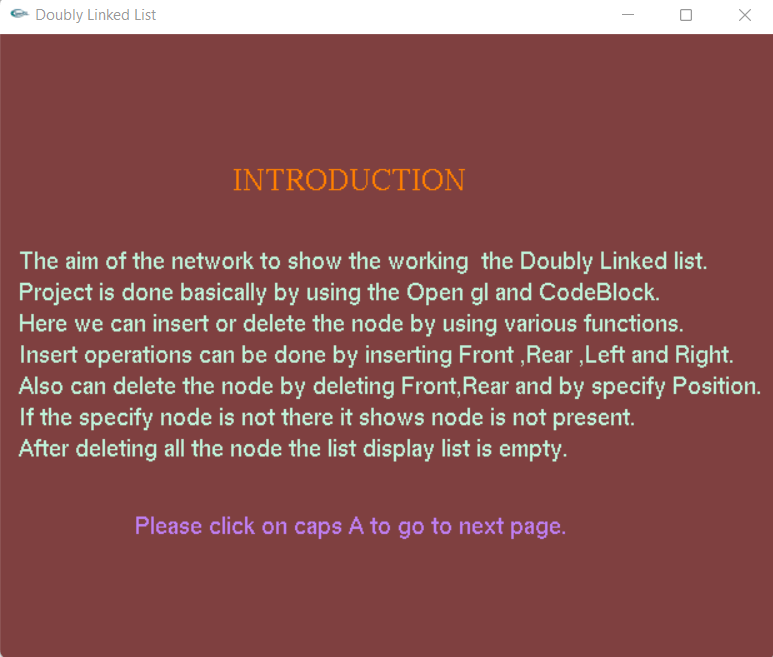
➢ RAM: 64MB or Higher

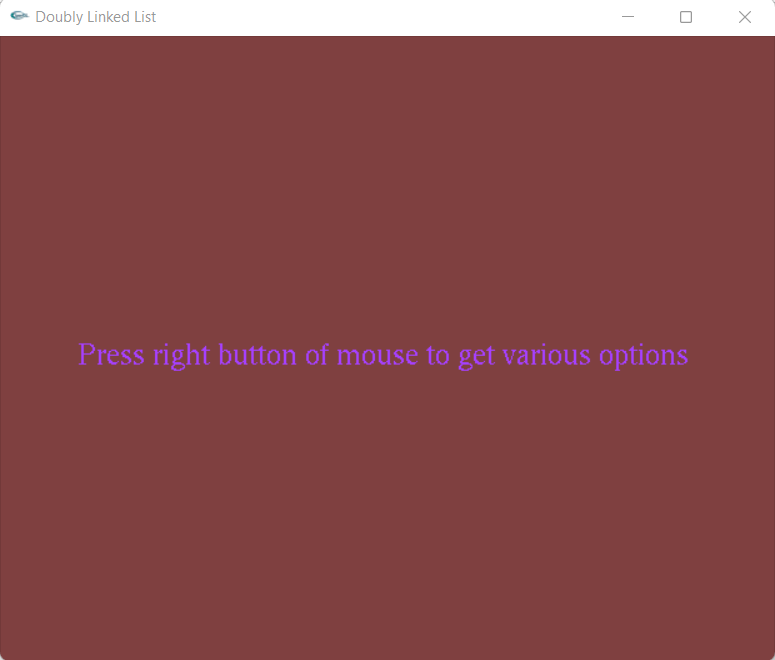
➢ Hard Disk: 80GB or Higher



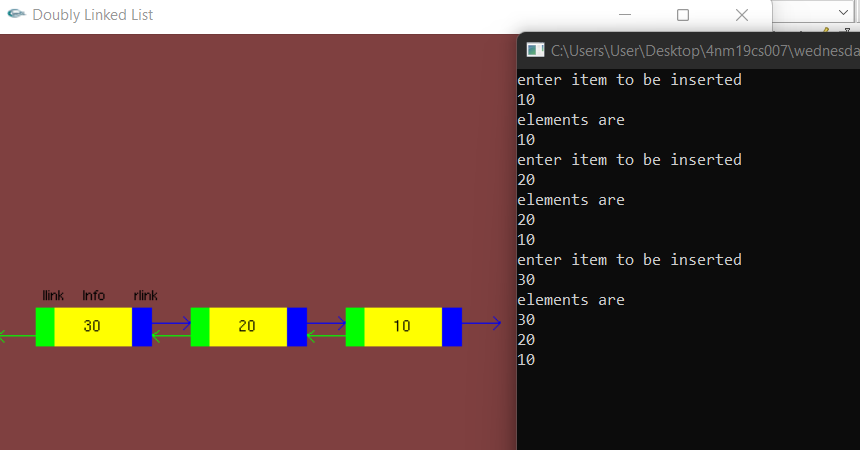
**SCREENSHOTS OF THE PROJECT**

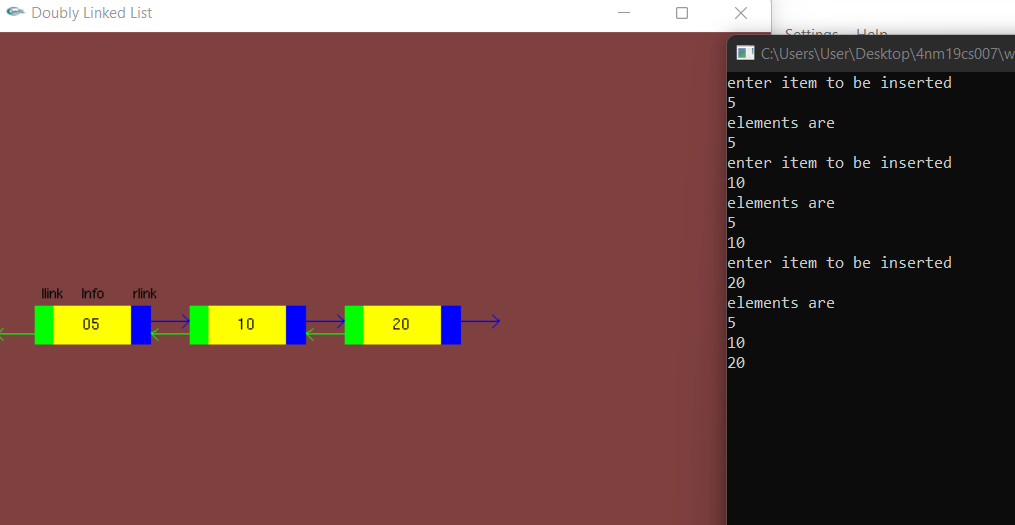


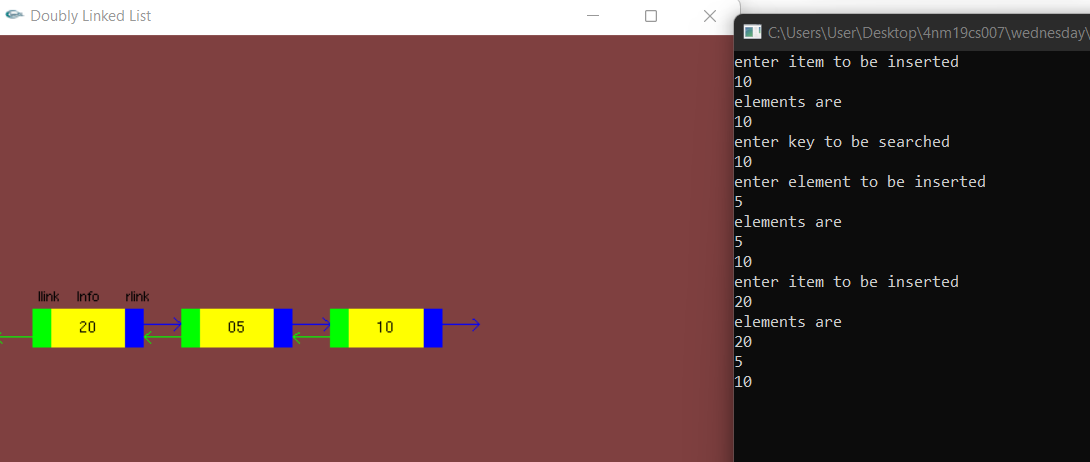
****

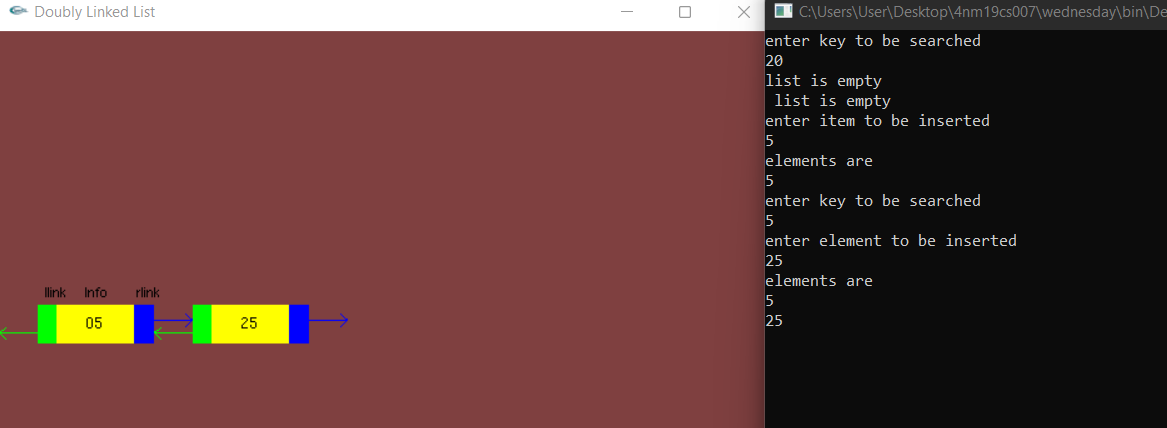
****

**INSERTING NODE AT FRONT , REAR , LEFT , RIGHT**

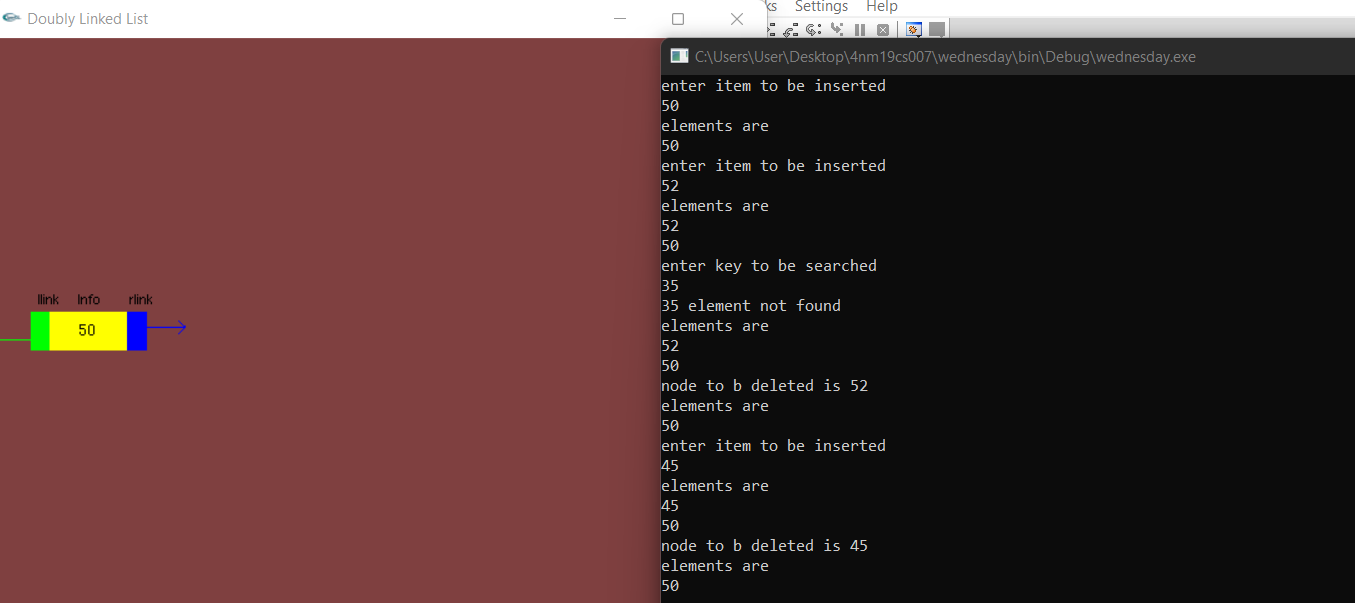
****

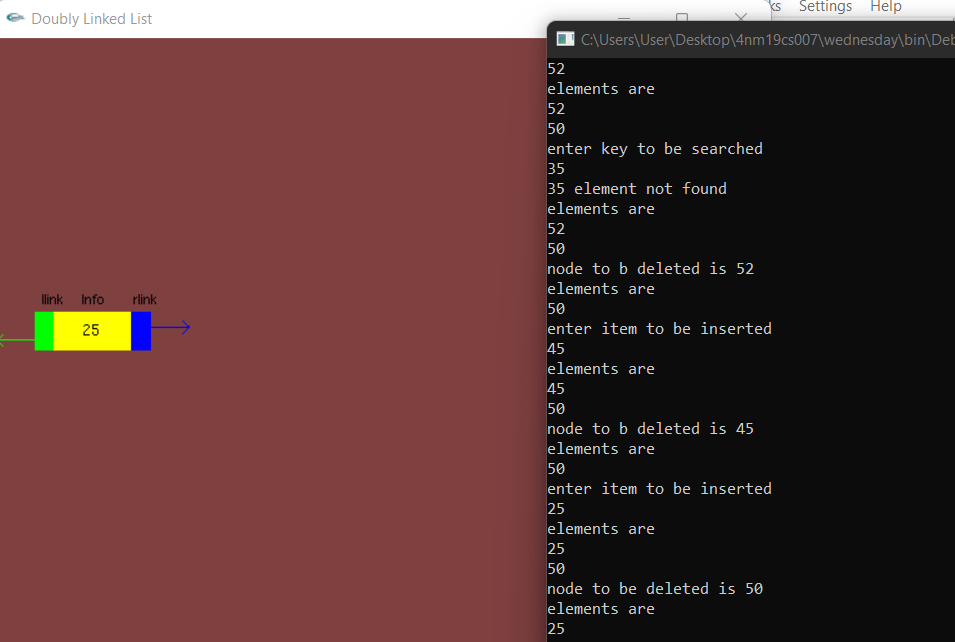
****

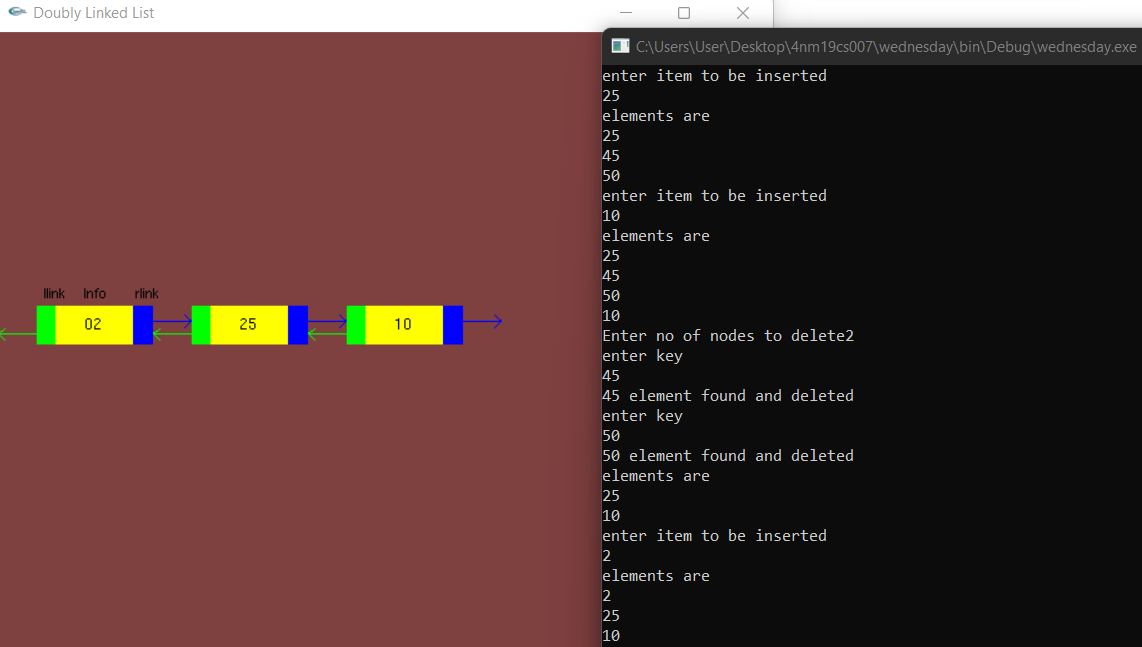
****

****

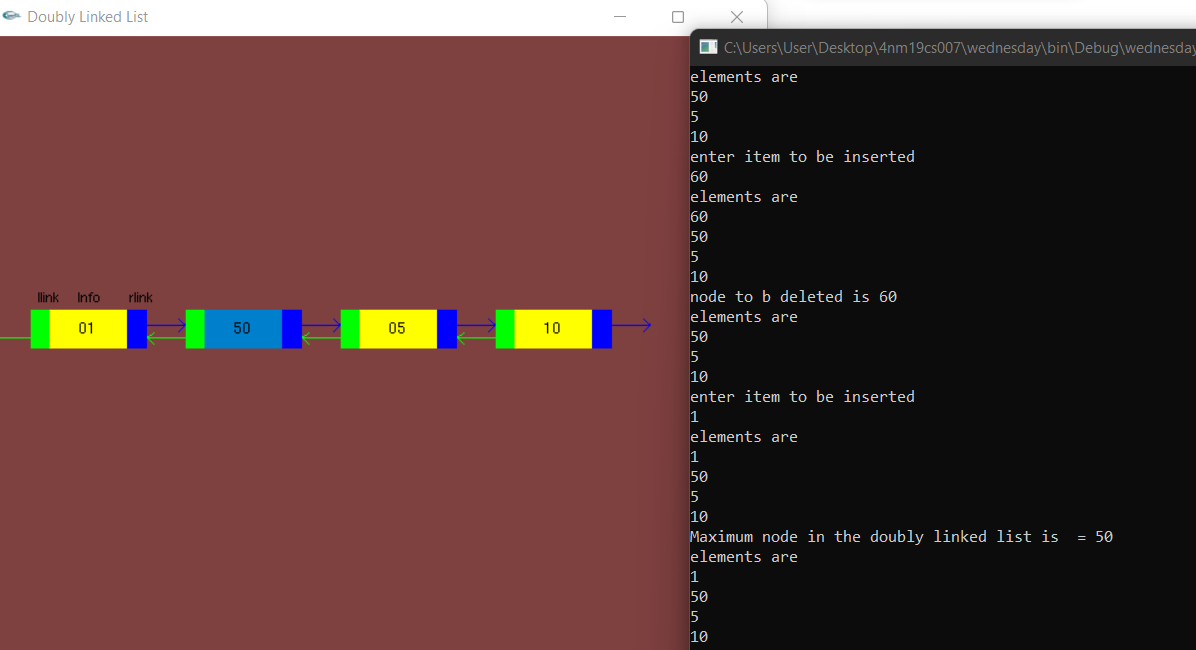
**DELETING NODE AT FRONT , REAR , SPECIFY THE NODE:**

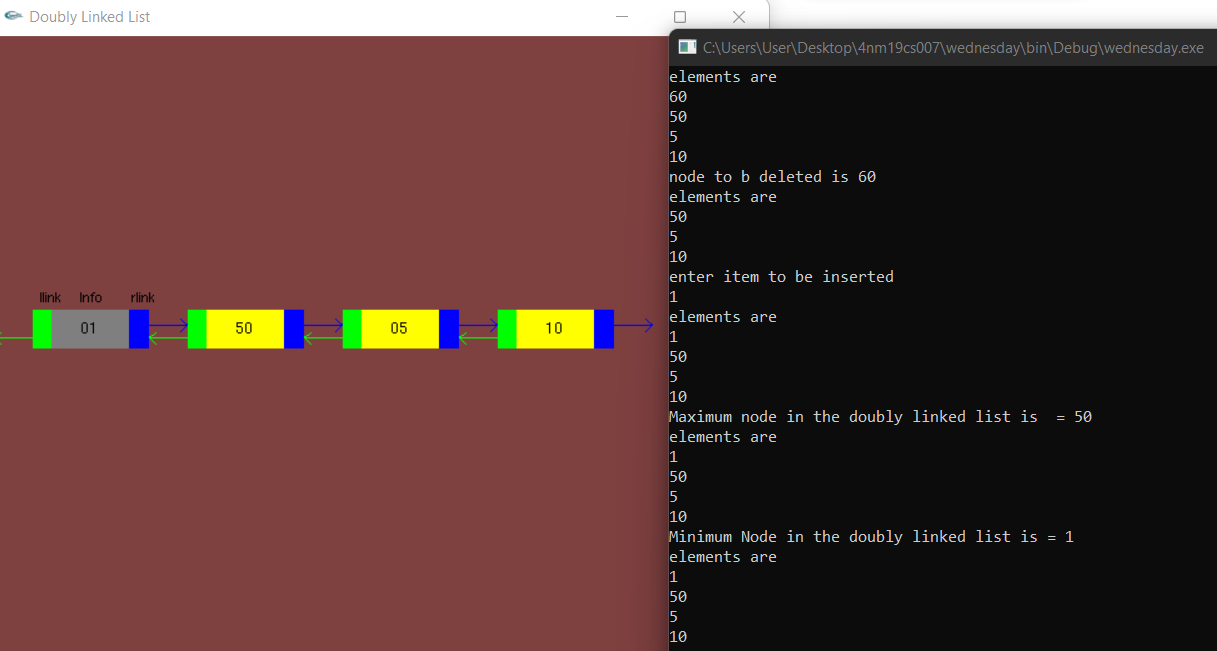
****

****

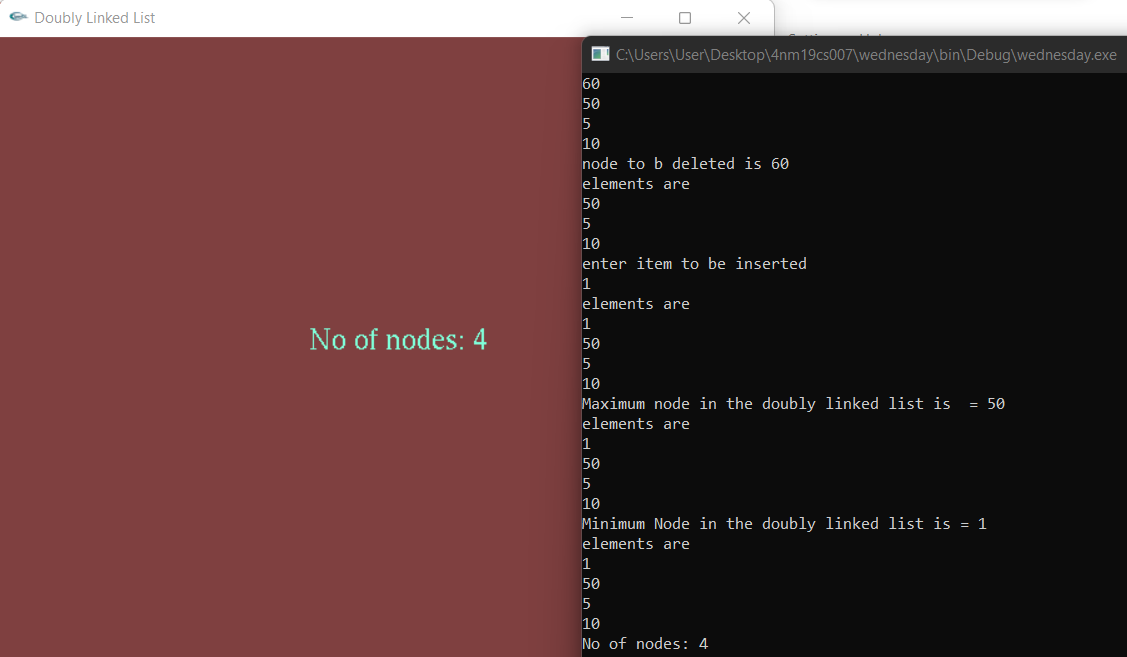
****

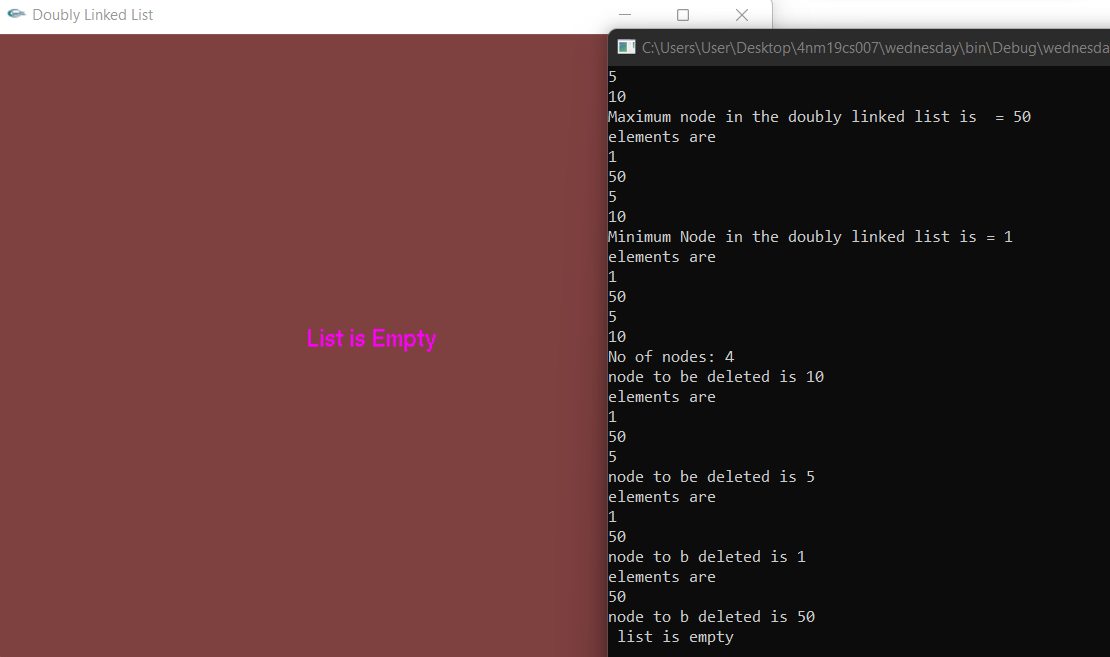
**MAXIMUM AND MINIMUM NODE:**

****

****

**TOTAL NO OF NODE AND DISPLAYING LIST IS EMPTY**

****

****

**CONCLUSION :**

We successfully implemented this project using OpenGL functionalities and learnt a lot about its implementation and used it thereby satisfying the proper requirements.

**FUTURE SCOPE :**

• Can be implemented by adding many more operations.

**REFERENCES**

* “Overview Guides Reference Samples Libraries Support Go to console” Documentation Firebase, <https://firebase.google.com/docs/>
* Stack Overflow, [https://stackoverflow.com](https://stackoverflow.com/)
* YouTube, [https://www.youtube.com](https://www.youtube.com/)