AS THE MATTER OF THIRD

OVERVIEW

This C program reads a list of integers from a text file and constructs a singly linked list from the data. It then finds and prints the third-to-last element of the linked list. If the linked list contains less than three elements, it outputs "-1" to indicate that the third-to-last element does not exist.

1. LINKED LIST NODE STRUCTURE (STRUCT NODE)

- The program defines a structure called **Node** to represent a node in the linked list.
- Each node contains the following fields:
    - **data**: An integer representing the data stored in the node.
    - **next**: A pointer to the next node in the linked list.

2. FUNCTION TO FIND THE THIRD-LAST ELEMENT (FINDTHIRDLAST)

- The **findThirdLast** function is responsible for finding the third-last element of the linked list.
- It takes a pointer to the head of the linked list as a parameter.
- The function first checks if the linked list contains fewer than three elements. If so, it returns -1 to indicate that the third-last element doesn't exist.
- If the linked list has at least three elements, it traverses the list using a **current** pointer until it reaches the third-last node.
- The data stored in the third-last node is returned as the result.

3. FUNCTION TO CREATE A NEW NODE (CREATENODE)

- The **createNode** function is a helper function that creates a new node and initializes it with the provided data.
- It takes an integer **data** as a parameter.
- The function dynamically allocates memory for a new node, sets its data field, and initializes the next pointer to **NULL**.
- It returns a pointer to the newly created node.

4. FUNCTION TO INSERT A NODE AT THE END (INSERTNODE)

- The **insertNode** function is responsible for inserting a new node at the end of the linked list.
- It takes a pointer to a pointer to the head of the linked list and an integer **data** as parameters.
- If the linked list is empty (head is **NULL**), the new node becomes the head of the list.
- Otherwise, the function traverses the linked list using a **current** pointer until it reaches the last node. The new node is then attached as the next node to the current last node.

5. Function to Print the Linked List (printList)

- The **printList** function is responsible for printing the elements of the linked list.
- It takes a pointer to the head of the linked list as a parameter.
- The function iterates through the linked list using a **current** pointer, printing each node's data.
- The elements are printed space-separated, and a newline is added at the end of the list to improve readability.

6. Main Function

- The **main** function is the entry point of the program.
- It initializes the **head** pointer of the linked list to **NULL**.
- The program opens the text file in read mode to read integer data.
- If the file opening fails, an error message is printed, and the program returns with a status code of 0.
- The program reads integers from the file and inserts them into the linked list using the **insertNode** function.
- After reading all the data, it closes the file.
- The program then prints the contents of the linked list using the **printList** function.
- It calls the **findThirdLast** function to find the third-last element in the linked list.
- If the linked list contains less than three elements, it prints a message indicating that and sets the third-last element to -1.
- If the third-last element exists, it prints its value.
- The program returns with a status code of 0 to indicate successful execution.