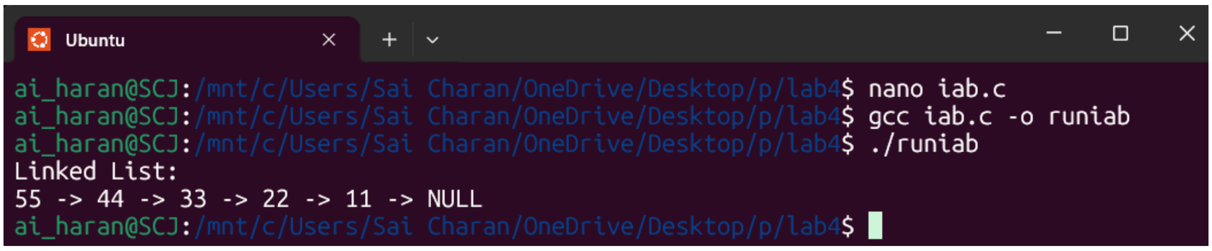
4. Write a program to create a class/structure Node to represent a node in a singly linked list. Each node should have two attributes: data and next. Implement the following operations:

i) insert_at_beginning(data): Insert a new node with the given data at the beginning of the list.

Code:

```c
#include <stdio.h>
#include <stdlib.h>

struct Node {
    int data;
    struct Node *next;
};

struct Node *head = NULL;

void insert_at_beginning(int data) {
    struct Node *newNode = (struct Node*)malloc(sizeof(struct Node));
    newNode->data = data;
    newNode->next = head;
    head = newNode;
}

// To display list
void display() {
    struct Node *temp = head;
    while (temp != NULL) {
        printf("%d -> ", temp->data);
        temp = temp->next;
    }
    printf("NULL\n");
}

int main() {
    insert_at_beginning(11);
    insert_at_beginning(22);
    insert_at_beginning(33);
    insert_at_beginning(44);
    insert_at_beginning(55);

    printf("Linked List:\n");
    display();

    return 0;
}
```

OutPut:



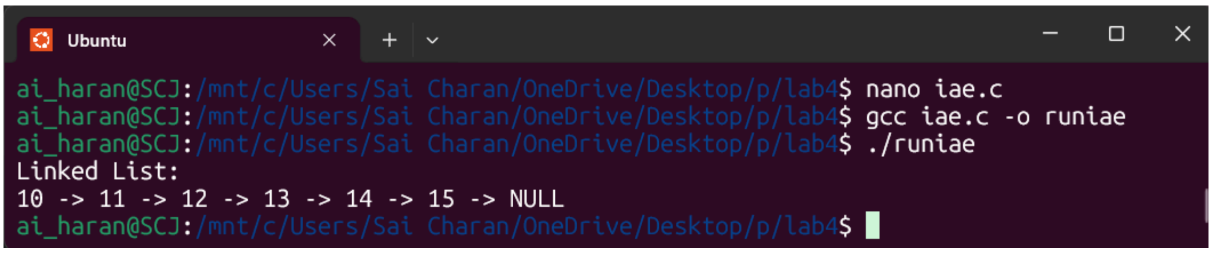
```
ai_haran@SCJ:/mnt/c/Users/Sai Charan/OneDrive/Desktop/p/lab4$ nano iab.c
ai_haran@SCJ:/mnt/c/Users/Sai Charan/OneDrive/Desktop/p/lab4$ gcc iab.c -o runiab
ai_haran@SCJ:/mnt/c/Users/Sai Charan/OneDrive/Desktop/p/lab4$ ./runiab
Linked List:
55 -> 44 -> 33 -> 22 -> 11 -> NULL
ai_haran@SCJ:/mnt/c/Users/Sai Charan/OneDrive/Desktop/p/lab4$
```

ii) insert_at_end(data): Insert a new node with the given data at the end of the list.

Code:

```c
// insert at end
void insert_at_end(int data) {
    struct Node *newNode = (struct Node*)malloc(sizeof(struct Node));
    newNode->data = data;
    newNode->next = NULL;

    if (head == NULL) {
        head = newNode;
        return;}

    // Traverse
    struct Node *temp = head;
    while (temp->next != NULL) {
        temp = temp->next;}
    temp->next = newNode;
}

int main() {
    insert_at_end(10);
    insert_at_end(11);
    insert_at_end(12);
    insert_at_end(13);
    insert_at_end(14);
    insert_at_end(15);

    printf("Linked List:\n");
    display();

    free_list();

    return 0; }
```

OutPut:



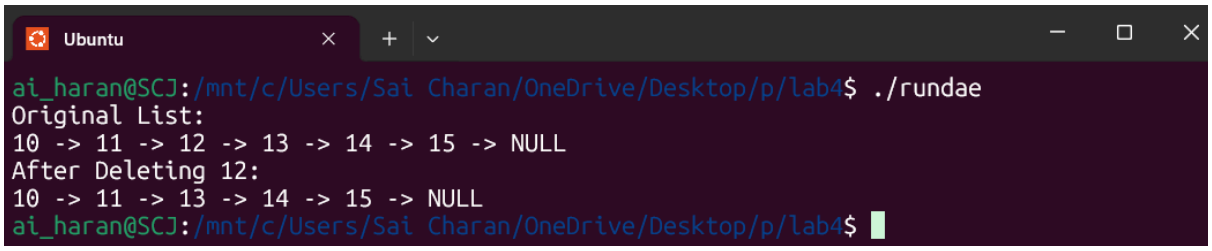
```
ai_haran@SCJ:/mnt/c/Users/Sai Charan/OneDrive/Desktop/p/lab4$ nano iae.c
ai_haran@SCJ:/mnt/c/Users/Sai Charan/OneDrive/Desktop/p/lab4$ gcc iae.c -o runiae
ai_haran@SCJ:/mnt/c/Users/Sai Charan/OneDrive/Desktop/p/lab4$ ./runiae
Linked List:
10 -> 11 -> 12 -> 13 -> 14 -> 15 -> NULL
ai_haran@SCJ:/mnt/c/Users/Sai Charan/OneDrive/Desktop/p/lab4$
```

iii) delete_node(data): Delete the first node in the list that contains the given data

Code:

```c
void delete_node(int data) {
    struct Node *temp = head;
    struct Node *prev = NULL;

    //if head node contains data
    if (temp != NULL && temp->data == data) {
        head = temp->next;
        free(temp);
        return;
    }

    //node to delete
    while (temp != NULL && temp->data != data) {
        prev = temp;
        temp = temp->next;
    }

    if (temp == NULL) {
        printf("Value %d not found in list.\n", data);
        return;
    }

    prev->next = temp->next;
    free(temp);
}
int main() {
    insert_at_end(10);
    insert_at_end(11);
    insert_at_end(12);
    insert_at_end(13);
    insert_at_end(14);
    insert_at_end(15);

    printf("Original List:\n");
    display();

    delete_node(12);
    printf("After Deleting 12:\n");
    display();

    free_list();

    return 0;
}
```

OutPut:



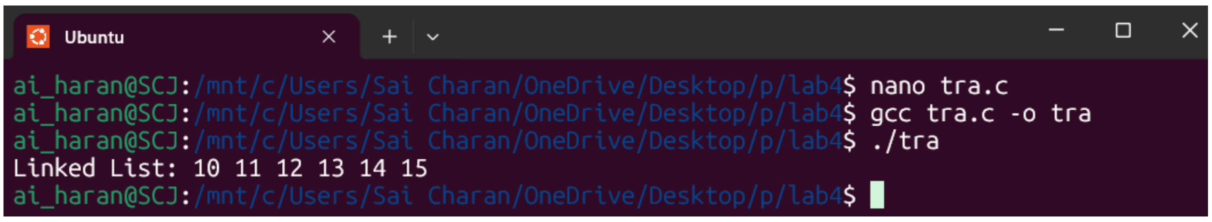
```
ai_haran@SCJ:/mnt/c/Users/Sai Charan/OneDrive/Desktop/p/lab4$ ./rundae
Original List:
10 -> 11 -> 12 -> 13 -> 14 -> 15 -> NULL
After Deleting 12:
10 -> 11 -> 13 -> 14 -> 15 -> NULL
ai_haran@SCJ:/mnt/c/Users/Sai Charan/OneDrive/Desktop/p/lab4$
```

iv) traverse (): Traverse the list and print the data of each node.

Code:

```c
void traverse() {
    struct Node *temp = head;

    if (temp == NULL) {
        printf("List is empty.\n");
        return;
    }

    printf("Linked List: ");
    while (temp != NULL) {
        printf("%d ", temp->data);
        temp = temp->next;
    }
    printf("\n");
}

int main() {
    insert_at_end(10);
    insert_at_end(11);
    insert_at_end(12);
    insert_at_end(13);
    insert_at_end(14);
    insert_at_end(15)

    traverse();

    free_list();

    return 0;
}
```

OutPut:



```
ai_haran@SCJ:/mnt/c/Users/Sai Charan/OneDrive/Desktop/p/lab4$ nano tra.c
ai_haran@SCJ:/mnt/c/Users/Sai Charan/OneDrive/Desktop/p/lab4$ gcc tra.c -o tra
ai_haran@SCJ:/mnt/c/Users/Sai Charan/OneDrive/Desktop/p/lab4$ ./tra
Linked List: 10 11 12 13 14 15
ai_haran@SCJ:/mnt/c/Users/Sai Charan/OneDrive/Desktop/p/lab4$
```