CODTECH Internship - Task 2

Task Title: Linked List Implementation

Objective: Create a program to implement a singly linked list with operations like insertion, deletion, and traversal.

Deliverable: A modular and efficient linked list implementation.

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

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

struct Node* head = NULL;

// Function to insert a node at the end
void insert(int value) {
    struct Node* newNode = (struct Node*)malloc(sizeof(struct Node));
    newNode->data = value;
    newNode->next = NULL;

    if (head == NULL) {
        head = newNode;
    } else {
        struct Node* temp = head;
        while (temp->next != NULL) {
            temp = temp->next;
        }
        temp->next = newNode;
    }
    printf("Inserted %d into the list.\n", value);
}

// Function to delete a node by value
void delete(int value) {
    struct Node *temp = head, *prev = NULL;

    if (temp != NULL && temp->data == value) {
        head = temp->next;
        free(temp);
        printf("Deleted %d from the list.\n", value);
        return;
    }

    while (temp != NULL && temp->data != value) {
        prev = temp;
        temp = temp->next;
    }
```

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

    prev->next = temp->next;
    free(temp);
    printf("Deleted %d from the list.\n", value);
}

// Function to traverse and print the linked list
void display() {
    struct Node* temp = head;
    printf("Linked List: ");
    while (temp != NULL) {
        printf("%d -> ", temp->data);
        temp = temp->next;
    }
    printf("NULL\n");
}

int main() {
    int choice, value;

    do {
        printf("\n======= CODTECH LINKED LIST MENU =======\n");
        printf("1. Insert\n");
        printf("2. Delete\n");
        printf("3. Display\n");
        printf("4. Exit\n");
        printf("Enter your choice: ");
        scanf("%d", &choice);

        switch (choice) {
            case 1:
                printf("Enter value to insert: ");
                scanf("%d", &value);
                insert(value);
                break;
            case 2:
                printf("Enter value to delete: ");
                scanf("%d", &value);
                delete(value);
                break;
            case 3:
                display();
                break;
            case 4:
                printf("Exiting program.\n");
                break;
            default:
                printf("Invalid choice! Try again.\n");
        }
    } while (choice != 4);
```

```
    return 0;
}
```