10. Linked List Operations

Aim:

To write a C program to implement linked list operations such as insertion, deletion, and display.

Algorithm:

1. Start the program.
2. Define a struct node with data and pointer to next.
3. Implement functions for insertion, deletion, and display.
4. Use a menu-driven main function.
5. End.

Code:

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

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

struct Node* head = NULL;

void insertEnd(int value) {
    struct Node* newNode = (struct Node*)malloc(sizeof(struct Node));
    newNode->data = value;
    newNode->next = NULL;

    if (head == NULL) {
        head = newNode;
        return;
    }
    struct Node* temp = head;
```

```
    while (temp->next != NULL)
        temp = temp->next;
    temp->next = newNode;
}

void deleteBegin() {
    if (head == NULL) {
        printf("List is empty\n");
        return;
    }
    struct Node* temp = head;
    head = head->next;
    free(temp);
}

void display() {
    struct Node* temp = head;
    if (temp == NULL) {
        printf("List is empty\n");
        return;
    }
    while (temp != NULL) {
        printf("%d -> ", temp->data);
        temp = temp->next;
    }
    printf("NULL\n");
}

int main() {
    insertEnd(10);
```

```
    insertEnd(20);
    insertEnd(30);
    printf("Linked list: ");
    display();
    deleteBegin();
    printf("After deletion: ");
    display();
    return 0;
}
```

Sample Output:

```
Linked list: 10 -> 20 -> 30 -> NULL
After deletion: 20 -> 30 -> NULL


=== Code Execution Successful ===
```

Result:

Successfully implemented linked list operations.