

10. Write a C program to implement Linked list operations.

```
#include <stdio.h>

#include <stdlib.h>

// Define node structure
struct Node {
    int data;
    struct Node* next;
};

struct Node* head = NULL;

// Insert at 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);
}
```

```
}
```

```
// Delete by value
```

```
void deleteNode(int value) {
```

```
    struct Node *temp = head, *prev = NULL;
```

```
    // If head node holds the value
```

```
    if (temp != NULL && temp->data == value) {
```

```
        head = temp->next;
```

```
        free(temp);
```

```
        printf("Deleted %d from the list.\n", value);
```

```
        return;
```

```
    }
```

```
    // Search for the value
```

```
    while (temp != NULL && temp->data != value) {
```

```
        prev = temp;
```

```
        temp = temp->next;
```

```
    }
```

```
    // If value not found
```

```
    if (temp == NULL) {
```

```
        printf("Value %d not found in the list.\n", value);
```

```
        return;
```

```
    }
```

```
    // Remove node
```

```
    prev->next = temp->next;
```

```

    free(temp);

    printf("Deleted %d from the list.\n", value);
}

// Display list
void display() {
    struct Node* temp = head;

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

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

// Main menu
int main() {
    int choice, value;

    do {
        printf("\n--- Linked List Operations ---\n");
        printf("1. Insert\n");

```

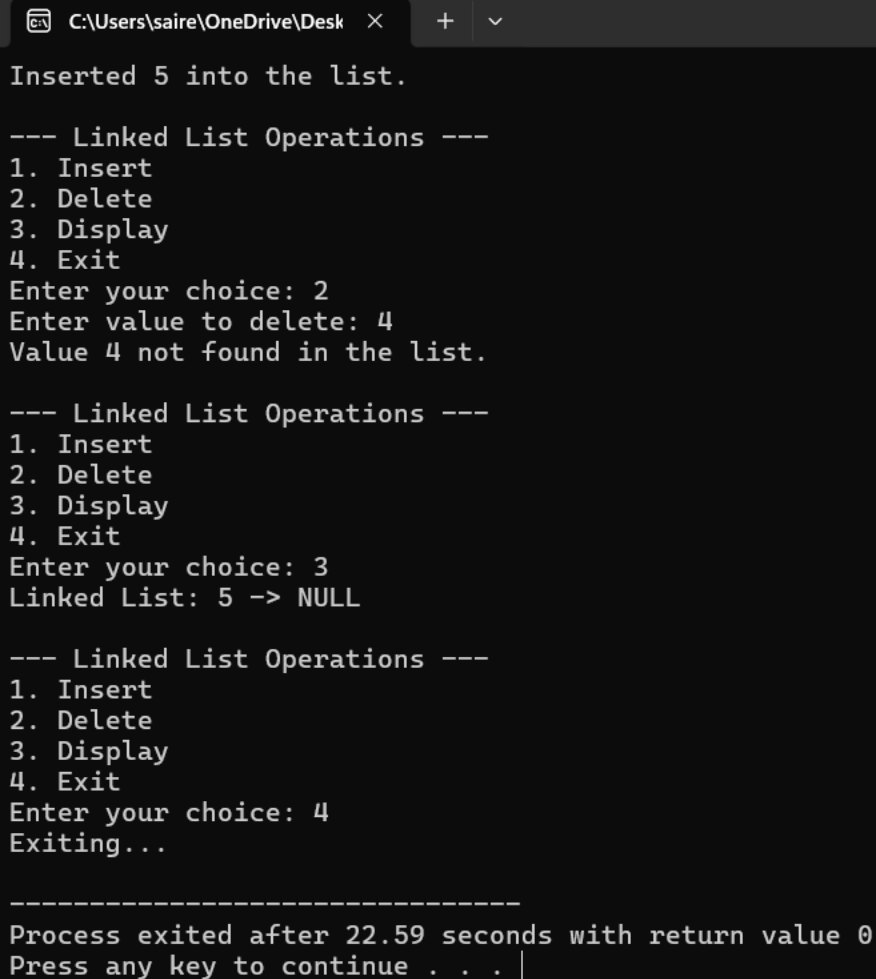
```
printf("2. Delete\n");
printf("3. Display\n");
printf("4. Exit\n");
printf("Enter your choice: ");
scanf("%d", &choice);

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);
        deleteNode(value);
        break;
    case 3:
        display();
        break;
    case 4:
        printf("Exiting...\n");
        break;
    default:
        printf("Invalid choice!\n");
}

} while(choice != 4);
```

```
    return 0;
}
```

OUTPUT



```
C:\Users\saire\OneDrive\Desktop >
Inserted 5 into the list.

--- Linked List Operations ---
1. Insert
2. Delete
3. Display
4. Exit
Enter your choice: 2
Enter value to delete: 4
Value 4 not found in the list.

--- Linked List Operations ---
1. Insert
2. Delete
3. Display
4. Exit
Enter your choice: 3
Linked List: 5 -> NULL

--- Linked List Operations ---
1. Insert
2. Delete
3. Display
4. Exit
Enter your choice: 4
Exiting...

-----
Process exited after 22.59 seconds with return value 0
Press any key to continue . . . |
```