

05. WAP to Implement Singly Linked List with following operations

- a) Create a linked list.
- b) Deletion of first element, specified element and last element in the list.
- c) Display the contents of the linked list.

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

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

struct Node* head = NULL;

// Function to create linked list (insert at end)
void create(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("Node inserted: %d\n", value);
}

// Delete first element
void deleteFirst() {
    if (head == NULL) {
        printf("List is empty!\n");
        return;
    }
    struct Node* temp = head;
    head = head->next;
    free(temp);
    printf("First node deleted.\n");
}
```

```

// Delete last element
void deleteLast() {
    if (head == NULL) {
        printf("List is empty!\n");
        return;
    }
    if (head->next == NULL) {
        free(head);
        head = NULL;
        printf("Last node deleted.\n");
        return;
    }
    struct Node* temp = head;
    while (temp->next->next != NULL) {
        temp = temp->next;
    }
    free(temp->next);
    temp->next = NULL;
    printf("Last node deleted.\n");
}

// Delete specific element
void deleteSpecific(int value) {
    if (head == NULL) {
        printf("List is empty!\n");
        return;
    }

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

    // If the node to delete is the head
    if (head->data == value) {
        head = head->next;
        free(temp);
        printf("Node %d deleted.\n", value);
        return;
    }

    // Search for node
    while (temp != NULL && temp->data != value) {
        prev = temp;
        temp = temp->next;
    }

    if (temp == NULL) {
        printf("Element %d not found.\n", value);
        return;
    }
}

```

```

}

prev->next = temp->next;
free(temp);
printf("Node %d deleted.\n", value);
}

// Display the linked list
void display() {
    if (head == NULL) {
        printf("List is empty!\n");
        return;
    }

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

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

    while (1) {
        printf("\n--- Singly Linked List Menu ---\n");
        printf("1. Create (Insert Node)\n");
        printf("2. Delete First Node\n");
        printf("3. Delete Last Node\n");
        printf("4. Delete Specific Node\n");
        printf("5. Display List\n");
        printf("6. Exit\n");

        printf("Enter your choice: ");
        scanf("%d", &choice);

        switch (choice) {
            case 1:
                printf("Enter value to insert: ");
                scanf("%d", &value);
                create(value);
                break;
        }
    }
}

```

```

case 2:
    deleteFirst();
    break;

case 3:
    deleteLast();
    break;

case 4:
    printf("Enter value to delete: ");
    scanf("%d", &value);
    deleteSpecific(value);
    break;

case 5:
    display();
    break;

case 6:
    exit(0);

default:
    printf("Invalid choice! Try again.\n");
}
}
return 0;
}

```

Output

```

--- Singly Linked List Menu ---
1. Create (Insert Node)
2. Delete First Node
3. Delete Last Node
4. Delete Specific Node
5. Display List
6. Exit
Enter your choice: 1
Enter value to insert: 10
Node inserted: 10

--- Singly Linked List Menu ---
1. Create (Insert Node)
2. Delete First Node
3. Delete Last Node
4. Delete Specific Node
5. Display List
6. Exit
Enter your choice: 1
Enter value to insert: 20
Node inserted: 20

```

```
C:\Users\Admin\Desktop\var X + - □ ×

--- Singly Linked List Menu ---
1. Create (Insert Node)
2. Delete First Node
3. Delete Last Node
4. Delete Specific Node
5. Display List
6. Exit
Enter your choice: 1
Enter value to insert: 30
Node inserted: 30

--- Singly Linked List Menu ---
1. Create (Insert Node)
2. Delete First Node
3. Delete Last Node
4. Delete Specific Node
5. Display List
6. Exit
Enter your choice: 1
Enter value to insert: 40
Node inserted: 40
```

```
C:\Users\Admin\Desktop\var X + - □ ×

--- Singly Linked List Menu ---
1. Create (Insert Node)
2. Delete First Node
3. Delete Last Node
4. Delete Specific Node
5. Display List
6. Exit
Enter your choice: 5
Linked List: 10 -> 20 -> 30 -> 40 -> NULL

--- Singly Linked List Menu ---
1. Create (Insert Node)
2. Delete First Node
3. Delete Last Node
4. Delete Specific Node
5. Display List
6. Exit
Enter your choice: 2
First node deleted.

--- Singly Linked List Menu ---
1. Create (Insert Node)
2. Delete First Node
3. Delete Last Node
4. Delete Specific Node
5. Display List
6. Exit
Enter your choice: 5
Linked List: 20 -> 30 -> 40 -> NULL
```

```
C:\Users\Admin\Desktop\var X + - □ ×

--- Singly Linked List Menu ---
1. Create (Insert Node)
2. Delete First Node
3. Delete Last Node
4. Delete Specific Node
5. Display List
6. Exit
Enter your choice: 3
Last node deleted.

--- Singly Linked List Menu ---
1. Create (Insert Node)
2. Delete First Node
3. Delete Last Node
4. Delete Specific Node
5. Display List
6. Exit
Enter your choice: 5
Linked List: 20 -> 30 -> NULL

--- Singly Linked List Menu ---
1. Create (Insert Node)
2. Delete First Node
3. Delete Last Node
4. Delete Specific Node
5. Display List
6. Exit
Enter your choice: 4
Enter value to delete: 20
```

```
C:\Users\Admin\Desktop\var + - □ ×
3. Delete Last Node
4. Delete Specific Node
5. Display List
6. Exit
Enter your choice: 4
Enter value to delete: 20
Node 20 deleted.

--- Singly Linked List Menu ---
1. Create (Insert Node)
2. Delete First Node
3. Delete Last Node
4. Delete Specific Node
5. Display List
6. Exit
Enter your choice: 5
Linked List: 30 -> NULL

--- Singly Linked List Menu ---
1. Create (Insert Node)
2. Delete First Node
3. Delete Last Node
4. Delete Specific Node
5. Display List
6. Exit
Enter your choice: 6

Process returned 0 (0x0) execution time : 68.465 s
Press any key to continue.
```