5a. 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.

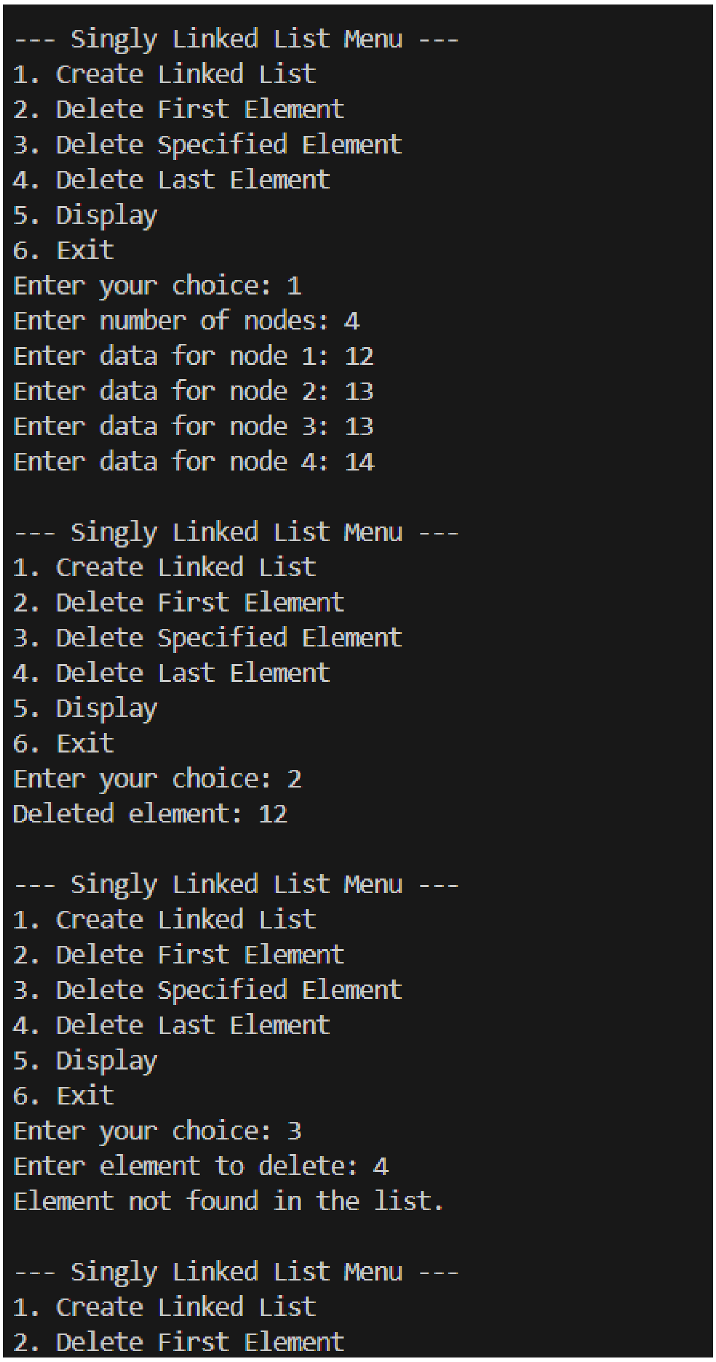
```
#include <stdio.h>
#include <stdlib.h>

// Definition of node
struct node {
    int data;
    struct node *next;
};

struct node *head = NULL;

// Create linked list
void create() {
    int n, value;
    struct node *temp, *newnode;

    printf("Enter number of nodes: ");
    scanf("%d", &n);

    for (int i = 0; i < n; i++) {
        newnode = (struct node *)malloc(sizeof(struct node));
        printf("Enter data for node %d: ", i + 1);
        scanf("%d", &value);

        newnode->data = value;
        newnode->next = NULL;

        if (head == NULL) {
            head = temp = newnode;
        } else {
            temp->next = newnode;
            temp = newnode;
        }
    }
}

// Delete first node
void delete_begin() {
    if (head == NULL) {
        printf("List is empty. Cannot delete.\n");
        return;
    }

    struct node *temp = head;
    head = head->next;
    printf("Deleted element: %d\n", temp->data);
    free(temp);
}
```

```
// Delete specified element
void delete_element() {
   int key;
   struct node *temp = head, *prev = NULL;

   if (head == NULL) {
      printf("List is empty. Cannot delete.\n");
      return;
   }

   printf("Enter element to delete: ");
   scanf("%d", &key);

   // If first node contains the element
   if (head->data == key) {
      delete_begin();
      return;
   }

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

   if (temp == NULL) {
      printf("Element not found in the list.\n");
      return;
   }
```

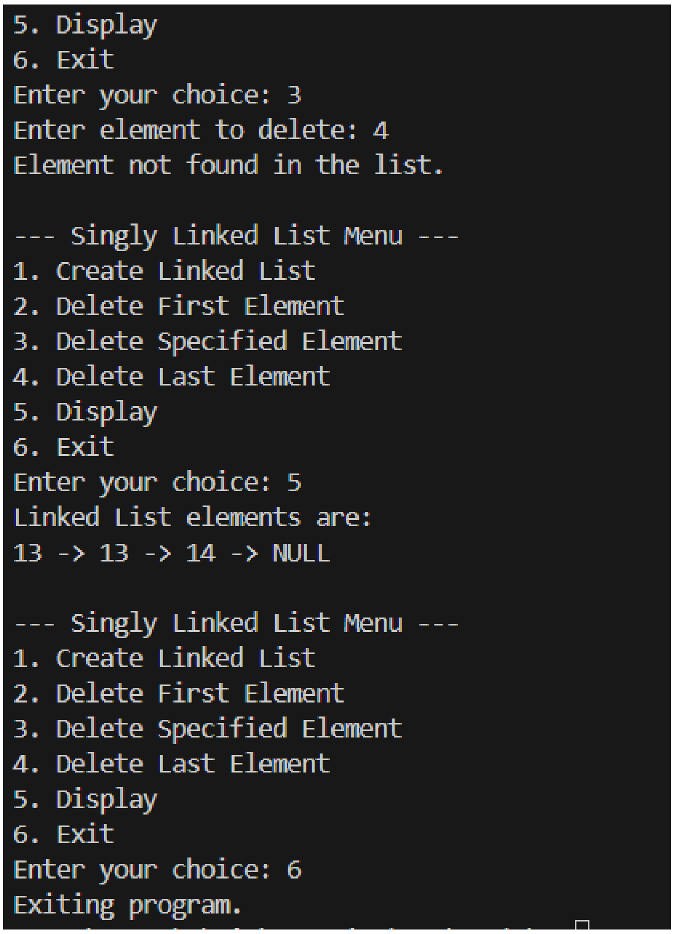
```
    prev->next = temp->next;
    printf("Deleted element: %d\n", temp->data);
    free(temp);
}

// Delete last node
void delete_end() {
    struct node *temp = head, *prev = NULL;

    if (head == NULL) {
        printf("List is empty. Cannot delete.\n");
        return;
    }

    // If only one node
    if (head->next == NULL) {
        printf("Deleted element: %d\n", head->data);
        free(head);
        head = NULL;
        return;
    }

    while (temp->next != NULL) {
        prev = temp;
        temp = temp->next;
    }
    prev->next = NULL;
    printf("Deleted element: %d\n", temp->data);
    free(temp);
}

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

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

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

int main() {
    int choice;

    do {
        printf("\n--- Singly Linked List Menu ---\n");
        printf("1. Create Linked List\n");
        printf("2. Delete First Element\n");
        printf("3. Delete Specified Element\n");
        printf("4. Delete Last Element\n");
        printf("5. Display\n");
        printf("6. Exit\n");
        printf("Enter your choice: ");
        scanf("%d", &choice);

        switch (choice) {
            case 1:
                create();
                break;
            case 2:
                delete_begin();
                break;
            case 3:
                delete_element();
                break;
            case 4:
                delete_end();
                break;
            case 5:
                display();
                break;
            case 6:
                printf("Exiting program.\n");
                break;
            default:
```

```
                printf("Invalid choice! Try again.\n");
            }
        } while (choice != 6);

        return 0;
    }
```

OUTPUT:



```
--- Singly Linked List Menu ---
1. Create Linked List
2. Delete First Element
3. Delete Specified Element
4. Delete Last Element
5. Display
6. Exit
Enter your choice: 1
Enter number of nodes: 4
Enter data for node 1: 12
Enter data for node 2: 13
Enter data for node 3: 13
Enter data for node 4: 14

--- Singly Linked List Menu ---
1. Create Linked List
2. Delete First Element
3. Delete Specified Element
4. Delete Last Element
5. Display
6. Exit
Enter your choice: 2
Deleted element: 12

--- Singly Linked List Menu ---
1. Create Linked List
2. Delete First Element
3. Delete Specified Element
4. Delete Last Element
5. Display
6. Exit
Enter your choice: 3
Enter element to delete: 4
Element not found in the list.

--- Singly Linked List Menu ---
1. Create Linked List
2. Delete First Element
```



```
5. Display
6. Exit
Enter your choice: 3
Enter element to delete: 4
Element not found in the list.

--- Singly Linked List Menu ---
1. Create Linked List
2. Delete First Element
3. Delete Specified Element
4. Delete Last Element
5. Display
6. Exit
Enter your choice: 5
Linked List elements are:
13 -> 13 -> 14 -> NULL

--- Singly Linked List Menu ---
1. Create Linked List
2. Delete First Element
3. Delete Specified Element
4. Delete Last Element
5. Display
6. Exit
Enter your choice: 6
Exiting program.
```