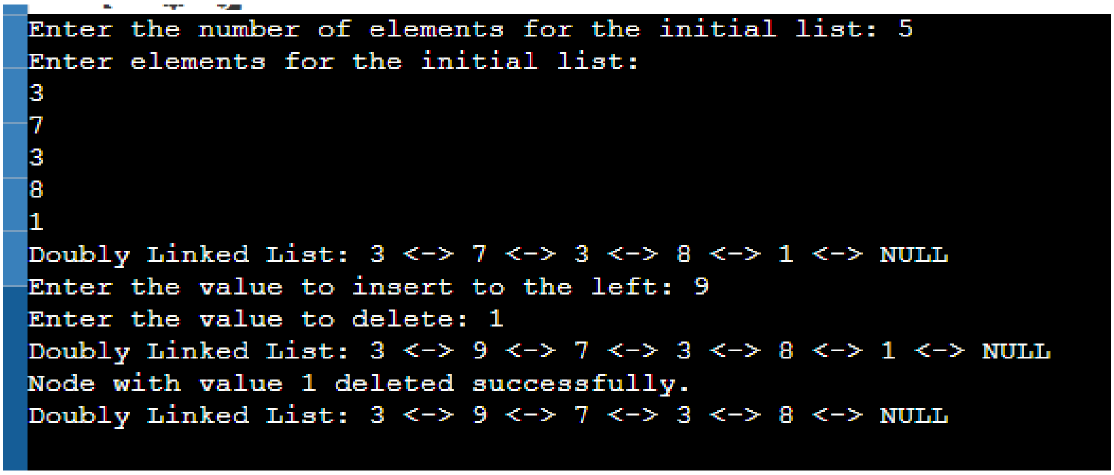
```
#include <stdio.h>
#include <stdlib.h>

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

struct Node* createList(int data) {
    struct Node* newNode = (struct Node*)malloc(sizeof(struct Node));
    newNode->data = data;
    newNode->prev = NULL;
    newNode->next = NULL;
    return newNode;
}

void insertLeft(struct Node** head, struct Node* target, int data) {
    if (*head == NULL) {
        printf("Error: List is empty!\n");
        return;
    }

    struct Node* newNode = (struct Node*)malloc(sizeof(struct Node));
    newNode->data = data;
    newNode->prev = NULL;
    newNode->next = NULL;

    if (target->prev != NULL) {
        target->prev->next = newNode;
        newNode->prev = target->prev;
    }

    target->prev = newNode;
    newNode->next = target;

    if (target == *head) {
        *head = newNode;
    }
}

void deleteNode(struct Node** head, int value) {
    if (*head == NULL) {
        printf("Error: List is empty!\n");
        return;
    }

    struct Node* current = *head;
    while (current != NULL) {
        if (current->data == value) {
            if (current->prev != NULL) {
                current->prev->next = current->next;
            } else {
                *head = current->next;
            }

            if (current->next != NULL) {
                current->next->prev = current->prev;
            }

            free(current);
            printf("Node with value %d deleted successfully.\n", value);
            return;
        }

        current = current->next;
    }

    printf("Node with value %d not found in the list.\n", value);
}

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

int main() {
    struct Node* list = NULL;
    int n, data;

    printf("Enter the number of elements for the initial list: ");
    scanf("%d", &n);

    if (n > 0) {
        printf("Enter elements for the initial list:\n");
        scanf("%d", &data);
        list = createList(data);

        struct Node* tail = list;

        for (int i = 1; i < n; ++i) {
            scanf("%d", &data);
            tail->next = createList(data);
            tail->next->prev = tail;
            tail = tail->next;
        }
    }

    displayList(list);

    int insertValue, deleteValue;

    printf("Enter the value to insert to the left: ");
    scanf("%d", &insertValue);

    printf("Enter the value to delete: ");
    scanf("%d", &deleteValue);

    insertLeft(&list, list->next, insertValue);
    displayList(list);

    deleteNode(&list, deleteValue);
    displayList(list);

    return 0;
}
```



```
Enter the number of elements for the initial list: 5
Enter elements for the initial list:
3
7
3
8
1
Doubly Linked List: 3 <-> 7 <-> 3 <-> 8 <-> 1 <-> NULL
Enter the value to insert to the left: 9
Enter the value to delete: 1
Doubly Linked List: 3 <-> 9 <-> 7 <-> 3 <-> 8 <-> 1 <-> NULL
Node with value 1 deleted successfully.
Doubly Linked List: 3 <-> 9 <-> 7 <-> 3 <-> 8 <-> NULL
```