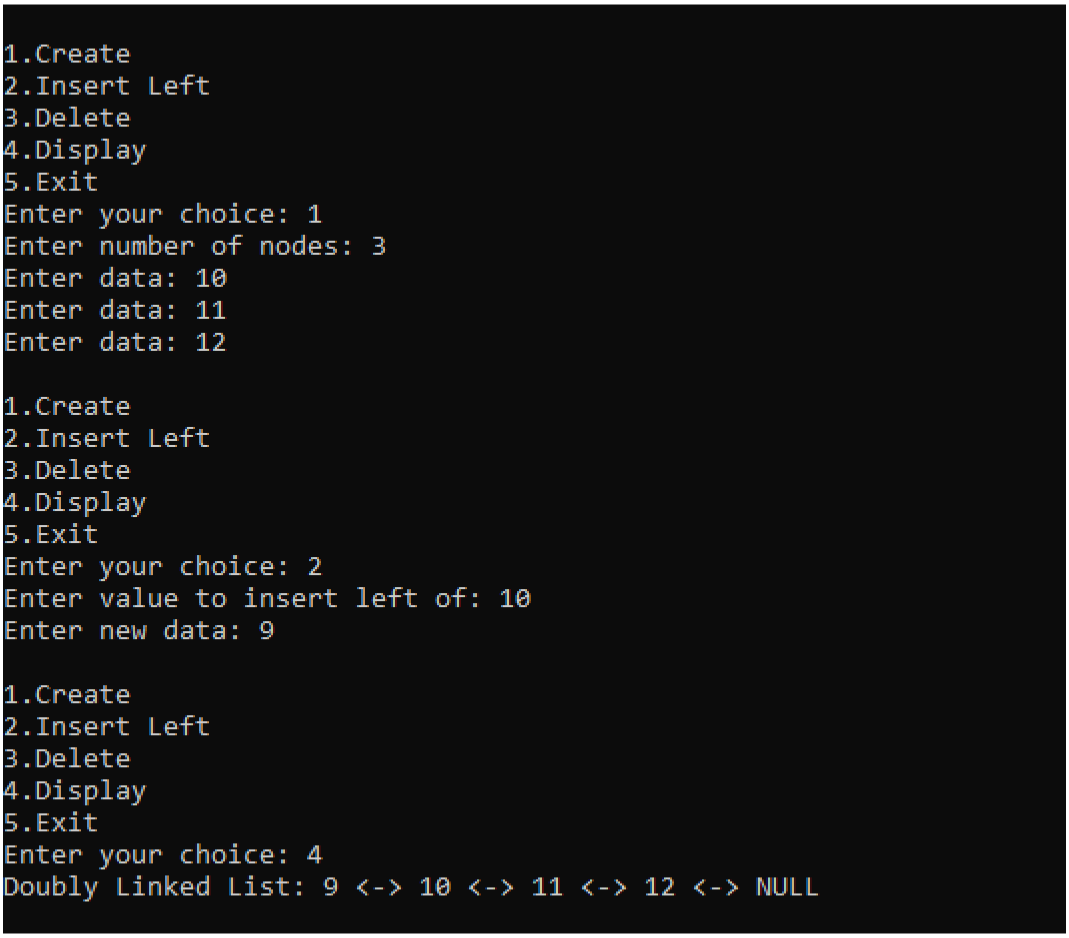
```
#include <stdio.h>
#include <stdlib.h>

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

struct node *head = NULL;

void create(int n) {
    struct node *newnode, *temp;
    int i, data;

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

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

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


void insert_left(int value, int newdata) {
    struct node *temp = head, *newnode;

    while (temp != NULL && temp->data != value)
        temp = temp->next;

    if (temp == NULL) {
        printf("Value not found!\n");
        return;
    }

    newnode = (struct node *)malloc(sizeof(struct node));
    newnode->data = newdata;
    newnode->next = temp;
    newnode->prev = temp->prev;

    if (temp->prev != NULL)
        temp->prev->next = newnode;
    else
        head = newnode;

    temp->prev = newnode;
}
```

```
void delete_value(int value) {
    struct node *temp = head;

    while (temp != NULL && temp->data != value)
        temp = temp->next;

    if (temp == NULL) {
        printf("Value not found!\n");
        return;
    }

    if (temp->prev != NULL)
        temp->prev->next = temp->next;
    else
        head = temp->next;

    if (temp->next != NULL)
        temp->next->prev = temp->prev;

    free(temp);
    printf("Node deleted successfully.\n");
}


void display() {
    struct node *temp = head;

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

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


int main() {
    int choice, n, value, newdata;

    while (1) {
        printf("\n1.Create\n2.Insert Left\n3.Delete\n4.Display\n5.Exit\n");
        printf("Enter your choice: ");
        scanf("%d", &choice);

        switch (choice) {
        case 1:
            printf("Enter number of nodes: ");
            scanf("%d", &n);
            create(n);
            break;
        case 2:
            printf("Enter value to insert left of: ");
            scanf("%d", &value);
            printf("Enter new data: ");
            scanf("%d", &newdata);
            insert_left(value, newdata);
            break;
          case 3:
            printf("Enter value to delete: ");
            scanf("%d", &value);
            delete_value(value);
            break;
          case 4:
            display();
            break;
          case 5:
            exit(0);
          default:
            printf("Invalid choice!\n");
          }
      }
      return 0;
  }
```



```
1.Create
2.Insert Left
3.Delete
4.Display
5.Exit
Enter your choice: 1
Enter number of nodes: 3
Enter data: 10
Enter data: 11
Enter data: 12

1.Create
2.Insert Left
3.Delete
4.Display
5.Exit
Enter your choice: 2
Enter value to insert left of: 10
Enter new data: 9

1.Create
2.Insert Left
3.Delete
4.Display
5.Exit
Enter your choice: 4
Doubly Linked List: 9 <-> 10 <-> 11 <-> 12 <-> NULL
```