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

/* Structure of doubly linked list node */
struct node {
    int data;
    struct node *prev;
    struct node *next;
};

struct node *head = NULL;

/* Create doubly linked list */
void create(int n) {
    int i, x;
    struct node *temp, *ptr;

    for (i = 0; i < n; i++) {
        scanf("%d", &x);
        temp = (struct node *)malloc(sizeof(struct node));
        temp->data = x;
        temp->prev = temp->next = NULL;

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

/* Insert a node to the left of a given value */
void insert_left(int value, int newdata) {
    struct node *temp, *ptr = head;

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

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

```
    temp = (struct node *)malloc(sizeof(struct node));
    temp->data = newdata;

    temp->next = ptr;
    temp->prev = ptr->prev;

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

    ptr->prev = temp;
}

/* Delete node with specific value */
void delete_value(int value) {
    struct node *ptr = head;

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

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

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

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

    free(ptr);
}

/* Display doubly linked list */
void display() {
    struct node *ptr = head;
    while (ptr != NULL) {
        printf("%d <-> ", ptr->data);
        ptr = ptr->next;
```

```
    }
    printf("NULL\n");
}

/* Main function */
int main() {
    int n, value, newdata;

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

    printf("Enter elements: ");
    create(n);

    printf("List: ");
    display();

    printf("Enter value to insert left of and new value: ");
    scanf("%d %d", &value, &newdata);
    insert_left(value, newdata);
    display();

    printf("Enter value to delete: ");
    scanf("%d", &value);
    delete_value(value);
    display();

    return 0;
}
```

Input
Enter number of nodes: 4
Enter elements: 10 20 30 40
Enter value to insert left of and new value: 30 25
Enter value to delete: 20

Output
List: 10 <-> 20 <-> 30 <-> 40 <-> NULL
10 <-> 20 <-> 25 <-> 30 <-> 40 <-> NULL
10 <-> 25 <-> 30 <-> 40 <-> NULL