

7	3	5	<p>WAP to Implement doubly link list with primitive operations</p> <ul style="list-style-type: none"> a) Create a doubly linked list. b) Insert a new node to the left of the node. c) Delete the node based on a specific value d) Display the contents of the list
		5	<p>Program - Leetcode platform</p>

*tj.c - Code::Blocks 25.03

File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help

<global>

S C

Start here x c.j.c x 56.c x t.c x *tj.c x

```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 /* Node structure */
5 struct node {
6     int data;
7     struct node *prev;
8     struct node *next;
9 };
10
11 struct node *head = NULL;
12
13 /* Create Doubly Linked List */
14 void create() {
15     int n, val;
16     struct node *newnode, *temp;
17
18     printf("Enter number of nodes: ");
19     scanf("%d", &n);
20
21     for (int i = 0; i < n; i++) {
22         newnode = (struct node*)malloc(sizeof(struct node));
23         printf("Enter data: ");
24         scanf("%d", &val);
25
26         newnode->data = val;
27         newnode->prev = newnode->next = NULL;
28
29         if (head == NULL) {
30             head = temp = newnode;
31         } else {
32             temp->next = newnode;
33             newnode->prev = temp;
34             temp = newnode;
35         }
36     }
37 }
```

Logs & others

C:\Users\HP\Documents\tj.c C/C++ Windows (CR+LF) WINDOWS-1252 Line 158, Col 1, Pos 3462 Insert Modified Read/Write default

*tj.c - Code::Blocks 25.03

File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help

<global>

S C

```
Start here x cj.c x 56.c x tc x *tj.c x
40 void insert_left() {
41     int key;
42     struct node *newnode, *temp;
43
44     if (head == NULL) {
45         printf("List is empty.\n");
46         return;
47     }
48
49     printf("Enter value to insert left of: ");
50     scanf("%d", &key);
51
52     temp = head;
53     while (temp != NULL && temp->data != key)
54         temp = temp->next;
55
56     if (temp == NULL) {
57         printf("Value not found.\n");
58         return;
59     }
60
61     newnode = (struct node*)malloc(sizeof(struct node));
62     printf("Enter new data: ");
63     scanf("%d", &newnode->data);
64
65     newnode->next = temp;
66     newnode->prev = temp->prev;
67
68     if (temp->prev != NULL)
69         temp->prev->next = newnode;
70     else
71         head = newnode;
72
73     temp->prev = newnode;
74 }
75
76 /* Delete node based on value */
```

Logs & others

C:\Users\HP\Documents\tj.c C/C++ Windows (CR+LF) WINDOWS-1252 Line 158, Col 1, Pos 3462 Insert Modified Read/Write default

*tj.c - Code::Blocks 25.03

File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help

<global>

S C

```
Start here x c.j.c x 56.c x t.c x *tj.c x
76  /* Delete node based on value */
77  void delete_value() {
78      int key;
79      struct node *temp;
80
81      if (head == NULL) {
82          printf("List is empty.\n");
83          return;
84      }
85
86      printf("Enter value to delete: ");
87      scanf("%d", &key);
88
89      temp = head;
90      while (temp != NULL && temp->data != key)
91          temp = temp->next;
92
93      if (temp == NULL) {
94          printf("Value not found.\n");
95          return;
96      }
97
98      if (temp->prev != NULL)
99          temp->prev->next = temp->next;
100     else
101         head = temp->next;
102
103     if (temp->next != NULL)
104         temp->next->prev = temp->prev;
105
106     printf("Deleted element: %d\n", temp->data);
107     free(temp);
108 }
109
110 /* Display Doubly Linked List */
111 void display() {
112     struct node *temp = head;
```

Logs & others

C:\Users\HP\Documents\tj.c C/C++ Windows (CR+LF) WINDOWS-1252 Line 158, Col 1, Pos 3462 Insert Modified Read/Write default

*tj.c - Code::Blocks 25.03

File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help

<global>

S C

Start here x cj.c x 56.c x tc x *tj.c x

```
122     temp = temp->next;
123 }
124 printf("NULL\n");
125 }
126
127 /* Main Function */
128 int main() {
129     int choice;
130
131     do {
132         printf("\n--- DOUBLY LINKED LIST MENU ---");
133         printf("\n1. Create List");
134         printf("\n2. Insert Left of Node");
135         printf("\n3. Delete Node by Value");
136         printf("\n4. Display");
137         printf("\n5. Exit");
138         printf("\nEnter your choice: ");
139         scanf("%d", &choice);
140
141         switch (choice) {
142             case 1: create();
143                     break;
144             case 2: insert_left();
145                     break;
146             case 3: delete_value();
147                     break;
148             case 4: display();
149                     break;
150             case 5: printf("Exiting program.\n");
151                     break;
152             default: printf("Invalid choice!\n");
153         }
154     } while (choice != 5);
155
156
157     return 0;
158 }
```

Logs & others

C:\Users\HP\Documents\tj.c C/C++ Windows (CR+LF) WINDOWS-1252 Line 158, Col 1, Pos 3462 Insert Modified Read/Write default

C:\Users\HP\Documents\tj.ex

--- DOUBLY LINKED LIST MENU ---

1. Create List
2. Insert Left of Node
3. Delete Node by Value
4. Display
5. Exit

Enter your choice: 7

Invalid choice!

--- DOUBLY LINKED LIST MENU ---

1. Create List
2. Insert Left of Node
3. Delete Node by Value
4. Display
5. Exit

Enter your choice: 3

Enter value to delete: 5

Deleted element: 5

--- DOUBLY LINKED LIST MENU ---

1. Create List
2. Insert Left of Node
3. Delete Node by Value
4. Display
5. Exit

Enter your choice: 4

Doubly Linked List contents:

3 <-> 4 <-> 6 <-> NULL

--- DOUBLY LINKED LIST MENU ---

1. Create List
2. Insert Left of Node
3. Delete Node by Value
4. Display
5. Exit

Enter your choice: |