

```
File Edit Selection View ... DSA 4
C Explorer
DSA 4
C.c
C.c
1 #include <stdio.h>
2 #include <stdlib.h>
3 struct Node {
4     int data;
5     struct Node* next;
6 };
7
8 struct Node* insert_at_beginning(struct Node* head, int data)
9 {
10     struct Node* newnode = (struct Node*)malloc(sizeof(struct Node));
11     if (newnode == NULL)
12     {
13         printf("Memory allocation failed!\n");
14         return head;
15     }
16     newnode->data = data;
17     newnode->next = head;
18     return newnode;
19 }
20
21 struct Node* insert_at_end(struct Node* head, int data)
22 {
23     struct Node* newnode = (struct Node*)malloc(sizeof(struct Node));
24     if (newnode == NULL)
25     {
26         printf("Memory allocation failed!\n");
27         return head;
28     }
29     newnode->data = data;
30     newnode->next = NULL;
31     if (head == NULL)
32     {
33         return newnode;
34     }
35     struct Node* temp = head;
36     while (temp->next != NULL)
37     {
38         temp = temp->next;
39     }
40     temp->next = newnode;
41     return head;
42 }
43
44 struct Node* delete_node(struct Node* head, int data)
45 {
46     struct Node* temp = head, *prev = NULL;
47
48     if (head == NULL)
49     {
50         printf("List is empty!\n");
51         return head;
52     }
53     if (temp->data == data)
54     {
55         head = temp->next;
56         free(temp);
57         return head;
58     }
59     while (temp != NULL && temp->data != data)
60     {
61         prev = temp;
62         temp = temp->next;
63     }
64     if (temp == NULL)
65     {
66         printf("Value not found!\n");
67         return head;
68     }
69     prev->next = temp->next;
70     free(temp);
71     return head;
72 }
73
74 void traverse(struct Node* head)
75 {
76     struct Node* temp = head;
77     if (temp == NULL)
78     {
79         printf("List is empty!\n");
80         return;
81     }
82     while (temp != NULL)
83     {
84         printf("%d -> ", temp->data);
85         temp = temp->next;
86     }
87     printf("NULL\n");
88 }
89
90 int main()
91 {
92     struct Node* head = NULL;
93 }
```

```
File Edit Selection View ... DSA 4
C Explorer
DSA 4
C.c
C.c
46 struct Node* insert_at_end(struct Node* head, int data)
47 {
48     struct Node* temp = head, *prev = NULL;
49
50     if (head == NULL)
51     {
52         printf("List is empty!\n");
53         return head;
54     }
55     if (temp->data == data)
56     {
57         head = temp->next;
58         free(temp);
59         return head;
60     }
61     while (temp != NULL && temp->data != data)
62     {
63         prev = temp;
64         temp = temp->next;
65     }
66     if (temp == NULL)
67     {
68         printf("Value not found!\n");
69         return head;
70     }
71     prev->next = temp->next;
72     free(temp);
73     return head;
74 }
75
76 void traverse(struct Node* head)
77 {
78     struct Node* temp = head;
79     if (temp == NULL)
80     {
81         printf("List is empty!\n");
82         return;
83     }
84     while (temp != NULL)
85     {
86         printf("%d -> ", temp->data);
87         temp = temp->next;
88     }
89     printf("NULL\n");
90 }
91
92 int main()
93 {
94     struct Node* head = NULL;
95 }
```

