

Experiment 1: Write a C program that uses functions to perform the following: a) Create a singly linked list of integers. b) Delete a given integer from the above linked list. c) Display the contents of the above list after deletion.

Exp 1.c

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  struct Node {
4      int data;
5      struct Node* next;
6  };
7  struct Node* createNode(int value) {
8      struct Node* newNode = (struct Node*)malloc(sizeof(struct Node));
9      newNode->data = value;
10     newNode->next = NULL;
11     return newNode;
12 }
13 void insertEnd(struct Node** head, int value) {
14     struct Node* newNode = createNode(value);
15     if (*head == NULL) {
16         *head = newNode;
17     } else {
18         struct Node* current = *head;
19         while (current->next != NULL) {
20             current = current->next;
21         }
22         current->next = newNode;
23     }
24 }
25 void deleteNode(struct Node** head, int value) {
26     if (*head == NULL) {
27         return;
28     }
29
30     if ((*head)->data == value) {
31         struct Node* temp = *head;
32         *head = (*head)->next;
33         free(temp);
34         return;
35     }
36
37     struct Node* current = *head;
38     while (current->next != NULL && current->next->data != value) {
39         current = current->next;
40     }
41
42     if (current->next != NULL) {
43         struct Node* temp = current->next;
44         current->next = temp->next;
45         free(temp);
46     }
47 }
```

```

48 void displayList(struct Node* head) {
49     struct Node* current = head;
50     while (current != NULL) {
51         printf("%d ", current->data);
52         current = current->next;
53     }
54     printf("\n");
55 }
56
57 int main() {
58     struct Node* head = NULL;
59
60     insertEnd(&head, 1);
61     insertEnd(&head, 2);
62     insertEnd(&head, 3);
63     insertEnd(&head, 4);
64     insertEnd(&head, 5);
65
66     printf("Original list: ");
67     displayList(head);
68
69     int valueToDelete;
70     printf("Enter the value to delete: ");
71     scanf("%d", &valueToDelete);
72
73     deleteNode(&head, valueToDelete);
74
75     printf("List after deletion: ");
76     displayList(head);
77
78     return 0;
79 }
80

```

Output:

```

D:\Academics\3rd Sem\DSUC  ×  +  ∨
Original list: 1 2 3 4 5
Enter the value to delete: 2
List after deletion: 1 3 4 5

-----
Process exited after 2.738 seconds with return value 0
Press any key to continue . . . |

```

Experiment 2: Write a C program that uses functions to perform the following: a) Create a doubly linked list of integers. b) Delete a given integer from the above doubly linked list. c) Display the contents of the above list after deletion.

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  struct Node {
4      int data;
5      struct Node* prev;
6      struct Node* next;
7  };
8  struct Node* createNode(int value) {
9      struct Node* newNode = (struct Node*)malloc(sizeof(struct Node));
10     newNode->data = value;
11     newNode->prev = NULL;
12     newNode->next = NULL;
13     return newNode;
14 }
15 void insertEnd(struct Node** head, int value) {
16     struct Node* newNode = createNode(value);
17     if (*head == NULL) {
18         *head = newNode;
19     } else {
20         struct Node* current = *head;
21         while (current->next != NULL) {
22             current = current->next;
23         }
24         current->next = newNode;
25         newNode->prev = current;
26     }
27 }
28 void deleteNode(struct Node** head, int value) {
29     if (*head == NULL) {
30         return;
31     }
32     struct Node* current = *head;
33     while (current != NULL && current->data != value) {
34         current = current->next;
35     }
36     if (current == NULL) {
37         printf("Value not found in the list.\n");
38         return;
39     }
40     if (current->prev != NULL) {
41         current->prev->next = current->next;
42     } else {
43         *head = current->next;
44     }
45     if (current->next != NULL) {
46         current->next->prev = current->prev;
47     }
48     free(current);
49 }
50 void displayList(struct Node* head) {
51     struct Node* current = head;
52     while (current != NULL) {
53         printf("%d ", current->data);
54     }
```

```

55         current = current->next;
56     }
57     printf("\n");
58 }
59 int main() {
60     struct Node* head = NULL;
61     insertEnd(&head, 1);
62     insertEnd(&head, 2);
63     insertEnd(&head, 3);
64     insertEnd(&head, 4);
65     insertEnd(&head, 5);
66     printf("Original list: ");
67     displayList(head);
68     int valueToDelete;
69     printf("Enter the value to delete: ");
70     scanf("%d", &valueToDelete);
71     deleteNode(&head, valueToDelete);
72     printf("List after deletion: ");
73     displayList(head);
74     return 0;
75 }
76

```

Output:

```

D:\Academics\3rd Sem\DSUC x + v
Original list: 1 2 3 4 5
Enter the value to delete: 3
List after deletion: 1 2 4 5

-----
Process exited after 2.197 seconds with return value 0
Press any key to continue . . .

```