<u>Experiment 1:</u> 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

```c
#include <stdio.h>
#include <stdlib.h>
struct Node {
    int data;
    struct Node* next;
};
struct Node* createNode(int value) {
    struct Node* newNode = (struct Node*)malloc(sizeof(struct Node));
    newNode->data = value;
    newNode->next = NULL;
    return newNode;
}
void insertEnd(struct Node** head, int value) {
    struct Node* newNode = createNode(value);
    if (*head == NULL) {
        *head = newNode;
    } else {
        struct Node* current = *head;
        while (current->next != NULL) {
            current = current->next;
        }
        current->next = newNode;
    }
}
void deleteNode(struct Node** head, int value) {
    if (*head == NULL) {
        return;
    }

    if ((*head)->data == value) {
        struct Node* temp = *head;
        *head = (*head)->next;
        free(temp);
        return;
    }

    struct Node* current = *head;
    while (current->next != NULL && current->next->data != value) {
        current = current->next;
    }

    if (current->next != NULL) {
        struct Node* temp = current->next;
        current->next = temp->next;
        free(temp);
    }
}
```

```c
void displayList(struct Node* head) {
    struct Node* current = head;
    while (current != NULL) {
        printf("%d ", current->data);
        current = current->next;
    }
    printf("\n");
}

int main() {
    struct Node* head = NULL;

    insertEnd(&head, 1);
    insertEnd(&head, 2);
    insertEnd(&head, 3);
    insertEnd(&head, 4);
    insertEnd(&head, 5);

    printf("Original list: ");
    displayList(head);

    int valueToDelete;
    printf("Enter the value to delete: ");
    scanf("%d", &valueToDelete);

    deleteNode(&head, valueToDelete);

    printf("List after deletion: ");
    displayList(head);

    return 0;
}
```

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.**

```c
#include <stdio.h>
#include <stdlib.h>
struct Node {
    int data;
    struct Node* prev;
    struct Node* next;
};
struct Node* createNode(int value) {
    struct Node* newNode = (struct Node*)malloc(sizeof(struct Node));
    newNode->data = value;
    newNode->prev = NULL;
    newNode->next = NULL;
    return newNode;
}
void insertEnd(struct Node** head, int value) {
    struct Node* newNode = createNode(value);
    if (*head == NULL) {
        *head = newNode;
    } else {
        struct Node* current = *head;
        while (current->next != NULL) {
            current = current->next;
        }
        current->next = newNode;
        newNode->prev = current;
    }
}
```

```c
void deleteNode(struct Node** head, int value) {
    if (*head == NULL) {
        return;
    }
    struct Node* current = *head;
    while (current != NULL && current->data != value) {
        current = current->next;
    }

    if (current == NULL) {
        printf("Value not found in the list.\n");
        return;
    }
    if (current->prev != NULL) {
        current->prev->next = current->next;
    } else {
        *head = current->next;
    }
    if (current->next != NULL) {
        current->next->prev = current->prev;
    }
    free(current);
}
void displayList(struct Node* head) {
    struct Node* current = head;
    while (current != NULL) {
        printf("%d ", current->data);
```

```
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
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 . . .
```