Implementation of Doubly Linked List

Write a C program to implement the following operations on Doubly Linked List.

(i) Insertion

(ii) Deletion

(iii) Search

(iv) Display

PROGRAM:

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

// Define the structure for a node
struct Node {
   int data;
   struct Node* prev;
   struct Node* next;
};

// Function to create a new node
struct Node* createNode(int data) {
   struct Node* newNode = (struct Node*)malloc(sizeof(struct Node));
   newNode->data = data;
   newNode->prev = NULL;
   newNode->next = NULL;
   return newNode;
}

// Function to insert a node at the beginning
void insertAtBeginning(struct Node** head, int data) {
   struct Node* newNode = createNode(data);
   newNode->next = *head;
   if (*head != NULL) {
      (*head)->prev = newNode;
   }
```

```
    *head = newNode;
}

// Function to insert a node at the end
void insertAtEnd(struct Node** head, int data) {
    struct Node* newNode = createNode(data);
    if (*head == NULL) {
        *head = newNode;
        return;
    }
    struct Node* temp = *head;
    while (temp->next != NULL) {
        temp = temp->next;
    }
    temp->next = newNode;
    newNode->prev = temp;
}

// Function to delete a node
void deleteNode(struct Node** head, struct Node* delNode) {
    if (*head == NULL || delNode == NULL) {
        return;
    }
    if (*head == delNode) {
        *head = delNode->next;
    }
    if (delNode->next != NULL) {
        delNode->next->prev = delNode->prev;
    }
    if (delNode->prev != NULL) {
        delNode->prev->next = delNode->next;
    }
    free(delNode);
}
```

```
// Function to search for a node
struct Node* searchNode(struct Node* head, int data) {
    struct Node* temp = head;
    while (temp != NULL) {
        if (temp->data == data) {
            return temp;
        }
        temp = temp->next;
    }
    return NULL;
}

// Function to display the list
void displayList(struct Node* head) {
    struct Node* temp = head;
    while (temp != NULL) {
        printf("%d -> ", temp->data);
        temp = temp->next;
    }
    printf("NULL\n");
}

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

    // Insert nodes
    insertAtBeginning(&head, 3);
    insertAtBeginning(&head, 2);
    insertAtBeginning(&head, 1);
    insertAtEnd(&head, 4);
    insertAtEnd(&head, 5);
    printf("List after insertions: ");
```

```
    displayList(head);

    // Search for a node
    struct Node* foundNode = searchNode(head, 3);
    if (foundNode) {
        printf("Element 3 found\n");
    } else {
        printf("Element 3 not found\n");
    }

    // Delete a node
    deleteNode(&head, foundNode);
    printf("List after deleting element 3: ");
    displayList(head);

    return 0;
}
```

OUTPUT:

```
List after insertions: 1 -> 2 -> 3 -> 4 -> 5 -> NULL
Element 3 found
List after deleting element 3: 1 -> 2 -> 4 -> 5 -> NULL
```