

Simulation (Linked List)

Code:

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

// Node structure
struct Node {
    int data;
    struct Node* next;
};

struct Node* head = NULL;

// Function to insert at beginning
void insertAtBeginning(int value) {
    struct Node* newNode = (struct Node*)malloc(sizeof(struct Node));
    newNode->data = value;
    newNode->next = head;
    head = newNode;
}

// Function to insert at end
void insertAtEnd(int value) {
    struct Node* newNode = (struct Node*)malloc(sizeof(struct Node));
    newNode->data = value;
    newNode->next = NULL;

    if (head == NULL) {
        head = newNode;
        return;
    }

    struct Node* temp = head;
    while (temp->next != NULL)
        temp = temp->next;

    temp->next = newNode;
}
```

```

// Function to delete first node
void deleteBeginning() {
    if (head == NULL) {
        printf("List is empty\n");
        return;
    }

    struct Node* temp = head;
    head = head->next;
    free(temp);
}

// Function to delete last node
void deleteEnd() {
    if (head == NULL) {
        printf("List is empty\n");
        return;
    }

    struct Node* temp = head;

    if (temp->next == NULL) {
        free(temp);
        head = NULL;
        return;
    }

    struct Node* prev = NULL;

    while (temp->next != NULL) {
        prev = temp;
        temp = temp->next;
    }

    prev->next = NULL;
    free(temp);
}

// Function to search element
void search(int key) {
    struct Node* temp = head;
    int position = 1;

```

```

    while (temp != NULL) {
        if (temp->data == key) {
            printf("Element %d found at position %d\n", key, position);
            return;
        }
        temp = temp->next;
        position++;
    }

    printf("Element %d not found\n", key);
}

// Function to display list
void display() {
    if (head == NULL) {
        printf("List is empty\n");
        return;
    }

    struct Node* temp = head;
    printf("Linked List: ");

    while (temp != NULL) {
        printf("%d -> ", temp->data);
        temp = temp->next;
    }

    printf("NULL\n");
}

// Main function
int main() {
    int choice, value, key;

    while (1) {
        printf("\n--- Linked List Menu ---\n");
        printf("1. Insert at Beginning\n");
        printf("2. Insert at End\n");
        printf("3. Delete from Beginning\n");
        printf("4. Delete from End\n");
        printf("5. Search Element\n");
        printf("6. Display List\n");
    }
}

```

```
printf("7. Exit\n");
printf("Enter your choice: ");
scanf("%d", &choice);

switch (choice) {
case 1:
    printf("Enter value: ");
    scanf("%d", &value);
    insertAtBeginning(value);
    break;

case 2:
    printf("Enter value: ");
    scanf("%d", &value);
    insertAtEnd(value);
    break;

case 3:
    deleteBeginning();
    break;

case 4:
    deleteEnd();
    break;

case 5:
    printf("Enter element to search: ");
    scanf("%d", &key);
    search(key);
    break;

case 6:
    display();
    break;

case 7:
    exit(0);
default:
    printf("Invalid choice\n");
}
}
return 0;
}
```

Output:

--- Linked List Menu ---

1. Insert at Beginning
2. Insert at End
3. Delete from Beginning
4. Delete from End
5. Search Element
6. Display List
7. Exit Enter your choice: 1 Enter value: 2

--- Linked List Menu ---

1. Insert at Beginning
2. Insert at End
3. Delete from Beginning
4. Delete from End
5. Search Element
6. Display List
7. Exit Enter your choice: 6 Linked List: 2 -> NULL

--- Linked List Menu ---

1. Insert at Beginning
2. Insert at End
3. Delete from Beginning
4. Delete from End
5. Search Element
6. Display List
7. Exit Enter your choice: 1 Enter value: 1

--- Linked List Menu ---

1. Insert at Beginning
2. Insert at End
3. Delete from Beginning
4. Delete from End

5. Search Element
6. Display List
7. Exit Enter your choice: 6 Linked List: 1 -> 2 -> NULL

--- Linked List Menu ---

1. Insert at Beginning
2. Insert at End
3. Delete from Beginning
4. Delete from End
5. Search Element
6. Display List
7. Exit Enter your choice: 2 Enter value: 3

--- Linked List Menu ---

1. Insert at Beginning
2. Insert at End
3. Delete from Beginning
4. Delete from End
5. Search Element
6. Display List
7. Exit Enter your choice: 2 Enter value: 4

--- Linked List Menu ---

1. Insert at Beginning
2. Insert at End
3. Delete from Beginning
4. Delete from End
5. Search Element
6. Display List
7. Exit Enter your choice: 2 Enter value: 5

--- Linked List Menu ---

1. Insert at Beginning
2. Insert at End
3. Delete from Beginning
4. Delete from End

5. Search Element
6. Display List
7. Exit Enter your choice: 2 Enter value: 6

--- Linked List Menu ---

1. Insert at Beginning
2. Insert at End
3. Delete from Beginning
4. Delete from End
5. Search Element
6. Display List
7. Exit Enter your choice: 6 Linked List: 1 -> 2 -> 3 -> 4 -> 5 -> 6 -> NULL

--- Linked List Menu ---

1. Insert at Beginning
2. Insert at End
3. Delete from Beginning
4. Delete from End
5. Search Element
6. Display List
7. Exit Enter your choice: 3

--- Linked List Menu ---

1. Insert at Beginning
2. Insert at End
3. Delete from Beginning
4. Delete from End
5. Search Element
6. Display List
7. Exit Enter your choice: 4

--- Linked List Menu ---

1. Insert at Beginning
2. Insert at End
3. Delete from Beginning
4. Delete from End

5. Search Element
6. Display List
7. Exit Enter your choice: 5 Enter element to search: 3 Element 3 found at position 2

--- Linked List Menu ---

1. Insert at Beginning
2. Insert at End
3. Delete from Beginning
4. Delete from End
5. Search Element
6. Display List
7. Exit Enter your choice: 6 Linked List: 2 -> 3 -> 4 -> 5 -> NULL