Aim

To write a C program to implement insertion, deletion, and display using a singly linked list.

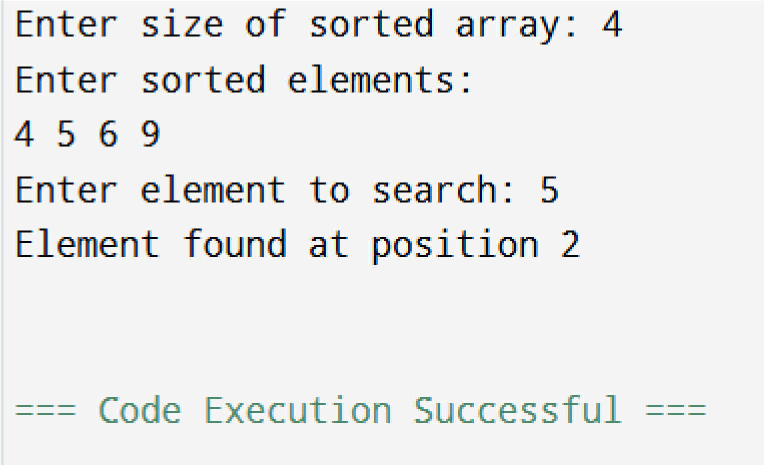
Algorithm

1. Start program.
2. Create a `struct Node` with `data` and `next`.
3. Provide menu: Insert at end, Delete first node, Display, Exit.
4. Perform operations using linked list pointers.
5. Repeat until exit.

Code

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

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

struct Node* head = NULL;

void insert(int val) {
    struct Node* newNode = (struct Node*)malloc(sizeof(struct Node));
    newNode->data = val;
    newNode->next = NULL;
    if (head == NULL)
        head = newNode;
    else {
        struct Node* temp = head;
        while (temp->next != NULL)
            temp = temp->next;
        temp->next = newNode;
    }
}

void delete() {
    if (head == NULL) {
        printf("List is empty\n");
        return;
    }
    struct Node* temp = head;
    head = head->next;
    free(temp);
}

void display() {
    struct Node* temp = head;
    if (temp == NULL) {
        printf("List is empty\n");
        return;
    }
    while (temp != NULL) {
        printf("%d -> ", temp->data);
        temp = temp->next;
    }
    printf("NULL\n");
}

int main() {
    int choice, val;
    do {
        printf("\n1. Insert\n2. Delete\n3. Display\n4. Exit\nEnter
choice: ");
        scanf("%d", &choice);
        switch(choice) {
            case 1:
                printf("Enter value: ");
                scanf("%d", &val);
                insert(val);
                break;
            case 2:
                delete();
                break;
            case 3:
                display();
                break;
        }
    } while(choice != 4);
    return 0;
}
```

Sample Output



```
Enter size of sorted array: 4
Enter sorted elements:
4 5 6 9
Enter element to search: 5
Element found at position 2


=== Code Execution Successful ===
```

Result

Linked list operations (insert, delete, display) work successfully.