EXERCISE-10

10.Write a c program to implement the Linked List operation.

AIM: To write a C program to implement basic operations on a singly linked list such as insertion, deletion, and display.

Algorithm:

1. Start the program.
2. Define a struct Node with data and next pointer.
3. Create a global pointer head = NULL.
4. Implement operations:
    - Insert at end: Create new node, append to end.
    - Delete by value: Search and unlink node with matching data.
    - Display list: Traverse and print all node values.
5. Use a menu to allow user to select an operation.
6. Loop until the user chooses to exit.
7. End the program.

Program Code:

```
#include <stdio.h>
#include <stdlib.h>
struct Node {
  int data;
  struct Node* next;
};
```

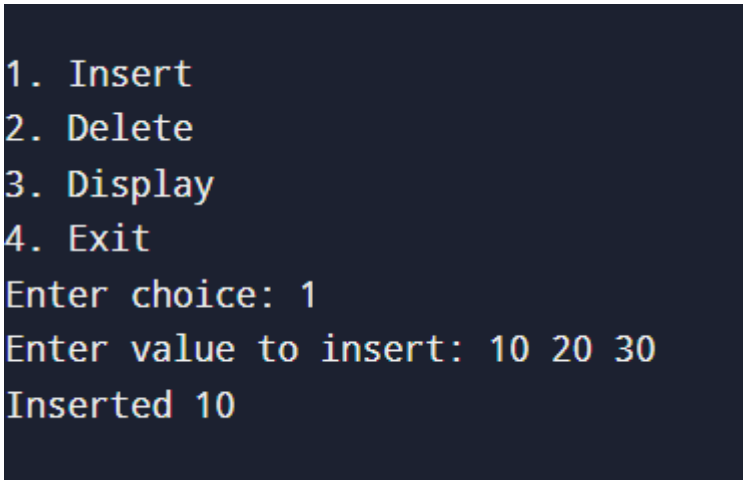
```
struct Node* head = NULL;
void insert(int value) {
   struct Node* newNode = (struct Node*)malloc(sizeof(struct
Node));
   newNode->data = value;
   newNode->next = NULL;
   if (head == NULL) {
      head = newNode;
   } else {
      struct Node* temp = head;
      while (temp->next != NULL)
         temp = temp->next;
      temp->next = newNode;
   }
   printf("Inserted %d\n", value);
}
void delete(int value) {
   struct Node *temp = head, *prev = NULL;
   while (temp != NULL && temp->data != value) {
      prev = temp;
      temp = temp->next;
   }
   if (temp == NULL) {
```

```
        printf("Value %d not found.\n", value);
        return;
    }
    if (prev == NULL)
        head = temp->next;
    else
        prev->next = temp->next;
    free(temp);
    printf("Deleted %d\n", value);
}
void display() {
    struct Node* temp = head;
    if (temp == NULL) {
        printf("List is empty.\n");
        return;
    }
    printf("Linked List: ");
    while (temp != NULL) {
        printf("%d -> ", temp->data);
        temp = temp->next;
    }
    printf("NULL\n");
}
```

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

```
            default:
                printf("Invalid choice.\n");
        }
    } while (choice != 4);
    return 0;
}
```

Input and Output:



Result:

The program successfully performs insertion, deletion, and display operations on a singly linked list.