CSA0317-DATA STRUCTURES

Program 10

```
#include <stdio.h>
#include <stdlib.h>
struct Node {
   int data;
   struct Node* next;
};
struct Node* head = NULL;
void insert(int val) {
   struct Node* n = malloc(sizeof(struct Node));
   n->data = val; n->next = NULL;
   if (!head) head = n;
   else {
      struct Node* t = head;
      while (t->next) t = t->next;
      t->next = n;
   }
}
void deleteVal(int val) {
   struct Node *t = head, *p = NULL;
   while (t && t->data != val) { p = t; t = t->next; }
   if (!t) { printf("Value %d not found!\n", val); return; }
   if (!p) head = t->next; else p->next = t->next;
   free(t);
   printf("Value %d deleted.\n", val);
}
void display() {
   struct Node* t = head;
   if (!t) { printf("List is empty.\n"); return; }
```

```
    printf("List: ");
    while (t) { printf("%d -> ", t->data); t = t->next; }
    printf("NULL\n");
}

int main() {
    int ch, val;
    while (1) {
        printf("\n--- MENU ---\n");
        printf("1. Insert\n2. Delete\n3. Display\n4. Exit\n");
        printf("Enter your choice: ");
        if (scanf("%d", &ch) != 1) {   // check input
            printf("Invalid input. Exiting.\n");
            break;
        }
        switch (ch) {
            case 1:
                printf("Enter value to insert: ");
                scanf("%d", &val);
                insert(val);
                break;
            case 2:
                printf("Enter value to delete: ");
                scanf("%d", &val);
                deleteVal(val);
                break;
            case 3:
                display();
                break;
            case 4:
                printf("Exiting...\n");
```

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

Output: