

**NAME: C Sahasra (192472078)**

**COURSE NAME : DATA STRUCTURES FOR MODERN COMPUTING SYSTEMS**

**COURSE CODE : CSA0302**

**WRITE A C PROGRAM TO PERFORM DELETION OF AN ELEMENT**

1. **At Beginning**
2. **At End**
3. **At any specific position**

**C PROGRAMMING CODE:**

#include <stdio.h>

#include <stdlib.h>

typedef struct Node {

int data;

struct Node\* next;

} Node;

Node\* head = NULL;

Node\* createNode(int value) {

Node\* newNode = (Node\*)malloc(sizeof(Node));

newNode->data = value;

newNode->next = NULL;

return newNode;

}

void insertAtEnd(int value) {

Node\* newNode = createNode(value);

if (head == NULL) {

head = newNode;

return;

}

Node\* temp = head;

while (temp->next != NULL) temp = temp->next;

temp->next = newNode;

}

void deleteAtBeginning() {

if (head == NULL) return;

Node\* temp = head;

head = head->next;

free(temp);

}

void deleteAtEnd() {

if (head == NULL) return;

if (head->next == NULL) {

free(head);

head = NULL;

return;

}

Node\* temp = head;

Node\* prev = NULL;

while (temp->next != NULL) {

prev = temp;

temp = temp->next;

}

prev->next = NULL;

free(temp);

}

void deleteAtPosition(int pos) {

if (head == NULL || pos <= 0) return;

if (pos == 1) {

deleteAtBeginning();

return;

}

Node\* temp = head;

Node\* prev = NULL;

for (int i = 1; i < pos && temp != NULL; i++) {

prev = temp;

temp = temp->next;

}

if (temp == NULL) return;

prev->next = temp->next;

free(temp);

}

void display() {

Node\* temp = head;

while (temp != NULL) {

printf("%d -> ", temp->data);

temp = temp->next;

}

printf("NULL\n");

}

int main() {

insertAtEnd(10);

insertAtEnd(20);

insertAtEnd(30);

insertAtEnd(40);

insertAtEnd(50);

deleteAtBeginning();

printf("After deletion at beginning: ");

display();

deleteAtEnd();

printf("After deletion at end: ");

display();

deleteAtPosition(2);

printf("After deletion at specific position: ");

display();

return 0;

}

**OUTPUT:**

