program 7 : Linked list → deletion

a) Deletion of first element

b) Deletion at specified position

c). Deletion at end

pseudo code: void deleteFirst(){:

```
if (head == NULL){ printf("Empty list");}
temp = head;
head = head -> next; }
```

void deleteLast(){

```
if (head == NULL) { printf("Empty list");}
if(head -> next == NULL){ head = NULL}
temp = head;
while (temp -> next != NULL){ prev = temp; temp = temp -> next;
} prev -> next = NULL }
```

void deleteSpecific (int value){

```
if(head == NULL) {printf("Empty list");
if(head -> data == value){head = head -> next;}
while(temp -> next != NULL && temp -> data != value)
{ prev = temp ; temp = temp -> next;}
if ( temp == NULL){ printf("Not found");
prev -> next = temp -> next;}
```

18/11

Code:

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

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

Struct  Node  *head = NULL;

void  deleteAtBeginning(){
    if (head==NULL){
      printf("Empty List");
      return;
    }
    struct Node *temp= head;
    printf(" deleted element : %d \n" temp->data);
    head= head->next;
    free(temp);
}

void deleteAtEnd(){
      if(head ==NULL) {
         printf("List is empty \n");
         return;
      }
     struct Node *temp = head, *prev = NULL;
     if(head -> next==NULL){
     printf(" deleted element : %d \n", head ->data);
     free (head);
     head= NULL;
     return; }
```

```
while (temp->next != NULL){
    prev = temp;
    temp = temp->next;
}
printf("deleted element : %d \n", temp->data);
prev->next = NULL;
free(temp);
}

void deleteAtPosition(int data){
    if (head == NULL){
        printf("List is empty");
        return;
    }
    struct Node *temp = head, *prev = NULL;
    if (head->data == data){
        printf("deleted element : %d \n", head->data);
        head = head->next;
        free(temp);
        return;
    }
    while (temp != NULL && temp->data != data){
        prev = temp;
        temp = temp->next;
    }
    if (temp == NULL){
        printf("Element not found");
        return;
    }
    prev->next = temp->next;
    printf("deleted element", temp->data);
    free(temp);
}
```

```
int main() {
    int choice, n, data;
    printf("1. Delete at beginning");
    printf("2. Delete at value");
    printf("3. Delete at end");
    while(1) {
        printf("Enter your choice: ");
        scanf("%d", &choice);
        switch(choice) {
        case 1:
                deleteAtBeginning();
                break;
        case 2:
                printf("Enter data to delete:");
                scanf("%d", &data);
                deleteAtPosition(data);
                break;
        case 3: deleteAtEnd();
                break;
        default: printf("wrong choice");
                break;
    }
    return 0;
}
```

13/11/25

output:

```
--Singly Linked list operations--
1. create linked list
2. Delete at beginning
3. Delete by value
4. Delete at End
5. Display List
6. Exit
```

Enter your choice: 1
Enter number of nodes: 4
Enter data for node 1: 1
Enter data for node 2: 2
Enter data for node 3: 3
Enter data for node 4: 4
Linked list created
Enter your choice 2
Deleted element 1
Enter your choice 3
Enter data to delete: 3
Deleted element: 3
Enter your choice 4
Deleted element: 4
Enter your choice 5
Linked list: 2 → NULL
Enter your choice 6
Exiting...

o/p seen