lab week-4

WAP to delete the first element, specified element and last element and display linked list.

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

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

struct Node* createNode(int value)
{
    struct Node* newNode = (struct Node*) malloc
    (sizeof (struct Node));
    newNode -> data = value;
    newNode -> next = NULL;
    return newNode;
}

void insertAtEnd (struct Node** head, int value)
{
    struct Node* newNode = createNode (value);
    if (*head == NULL)
    {
        *head = newNode;
    }
    else
    {
        struct Node* temp = *head;
        while (temp -> next != NULL)
        {
            temp = temp -> next;
        }
        temp -> next = newNode;
    }
}

void deleteFirst (struct Node** head) {
    if (*head != NULL) {
        struct Node* temp = *head;
        *head = (*head) -> next;
        free (temp);
    }
}

void deleteElement (struct Node** head, int value){
    struct Node* current = *head;
    struct Node* prev = NULL;

    while (current != NULL && current -> data
            != value){
        prev = current;
        current = current -> next;
    }
    if (current == NULL) {
        return;
    }
    if (prev == NULL) {
        *head = current -> next;
    }
    else {
        prev -> next = current -> next;
    }
    free (current);
}
```

URBAN EDGE 18/1/24

```
Void deleteLast (Struct node ** head) {
    if (*head == NULL) {
        Return;
    }
    Struct Node * temp = *head;
    Struct Node * prev = NULL;

    while (temp -> next != NULL) {
        prev = temp;
        temp = temp -> next;
    }

    if (prev == NULL) {
        * head = NULL;
    }
    else {
        prev -> next = NULL;
    }
    free (temp);
}

Void displayList (Struct Node* head) {
    Struct Node * temp = head;
    while (temp != NULL) {
        printf ("%d ->", temp -> data);
        temp = temp -> next;
    }
    printf ("NULL \n");
}
```

URBAN EDGE 18/1/24

```
int main()
{
    Struct Node * head = NULL;

    insertAtEnd (& head, 1);
    insertAtEnd (& head, 2);
    insertAtEnd (& head, 3);
    insertAtEnd (& head, 4);
    insertAtEnd (& head, 5);

    printf (" Initial Linked List: ");
    displayList (head);

    deleteFirst (& head);
    printf ("\nAfter deleting the first element : ");
    displayList (head);

    deleteElement (& head, 2);
    printf ("\nAfter deleting the Specified element(2): ");
    displayList (head);

    deleteLast (& head);
    printf ("\nAfter deleting the last element :");
    displayList (head);

    Return 0;
}
```

o/p:-

Initial Linked List 1 -> 2 -> 3 -> 4 -> 5 -> NULL
After deleting the first element 2 -> 3 -> 4 -> 5 -> NULL
After deleting the Specified element (2) : 3 -> 4 -> 5 -> NULL
After deleting the last element : 3 -> 4 -> NULL

Sign
18/1/24