22/1/23

Insertion and Deletion in a Linked list :

```
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;

}

struct Node * insertAtBeginning ( struct Node * head,
                                   int value) {

    struct Node * newNode = createNode (value);
    newNode -> next = head;
    return newNode;

}

Struct Node * insert At End ( struct Node* head,
                               int value ) {
    struct Node * ~~insert~~ newNode = createNode ( value);
    if(head = =NULL) {
          return newNode ;
    }

    struct Node* temp = head;
    while( temp.next != NULL) {
          temp = temp -> next;

    temp -> next = newNode;
    return head;

}


struct Node* insertAtposition (struct Node* head, int valu
                               int position) {
    struct Node* newNode = createNode( value);

    if (position = =1)
    {
        ~~positi~~
        newNode -> next = head;
        return newNode;
    }

    struct Node* temp = head;
                                  ~~if~~ temp! = NULL
    for( int i = 1 ; i < position - 1 ; i++) {
           temp = temp -> next ;
    }
```

Date
Page

```
        if (temp==NULL) {
                printf("Invalid position");
            return head;
        }

        newNode -> next = temp -> next;
        temp -> next = newNode;
        return head;
    }

    //display

        void display ( struct Node* head) {
                printf("LinkedList : ");
                while (head != NULL) {
                    printf("%d -> ", head -> data);
                    head = head -> next
                }
                printf("NULL\n");

    }

    // delete at Beginning

    struct Node* deleteFirst (struct Node* head)
    {
        if (head==NULL) {
                printf("List is empty");
                return NULL;
        }
        struct Node* temp = head;
        head=head -> next;
        free (temp);
        return head;
```

```
struct Node * deleteElement ( struct Node *head,
                                    int value) {
    if ( head = = NULL ) {
        printf ( " list is empty " ) ;
        return NULL ;
    }

    if ( head -> data = = value ) {
            struct Node * temp = head ;
            head = head -> next ;
            free (temp) ;
            return head ;
    }

     struct    Node *current  =  head ;
      ~~while~~
      while ( current -> next ! = NULL  &&
                current -> next -> data ! = value ) {
                    current = current -> next )
      }

      if ( current -> next = = NULL ) {
              printf ( " Element not found \n " ) ;
              return head ;
      }

      struct Node * temp = current -> next ;
        current -> next = current -> next -> next
          free ( temp ) ;
         return head )
```

Date
Page

```
struct Node * deletelast (struct Node *head)
{
    if (head == NULL) {
        printf("List is empty");
        return NULL;
    }

    if (head -> next == NULL) {
            free (head);
            return NULL;
    }

    struct Node *current = head;
     while ( current -> next -> next != NULL) {
             current = current -> next;
     }

      free (current -> next);
      current -> next = NULL;
      return head;
}

int main() {

    struct Node* head = NULL;

    head = insertAtBeginning (head, 3);
    head = insertAtBeginning (head, 2);
    head = insertAtBeginning (head, 1);
    head = insertAtEnd (head, 4);
    head = insertAtEnd (head, 5);
```

[illegible] 22/1/24

```
    display ( head );

    head =  deleteAtfirst ( head );
    display ( head );

    head = deleteElement ( head, 3 );
    display ( head );

    head = deleteLast ( head ) ;
    display ( head );


    return 0 ;
}
```

Output :

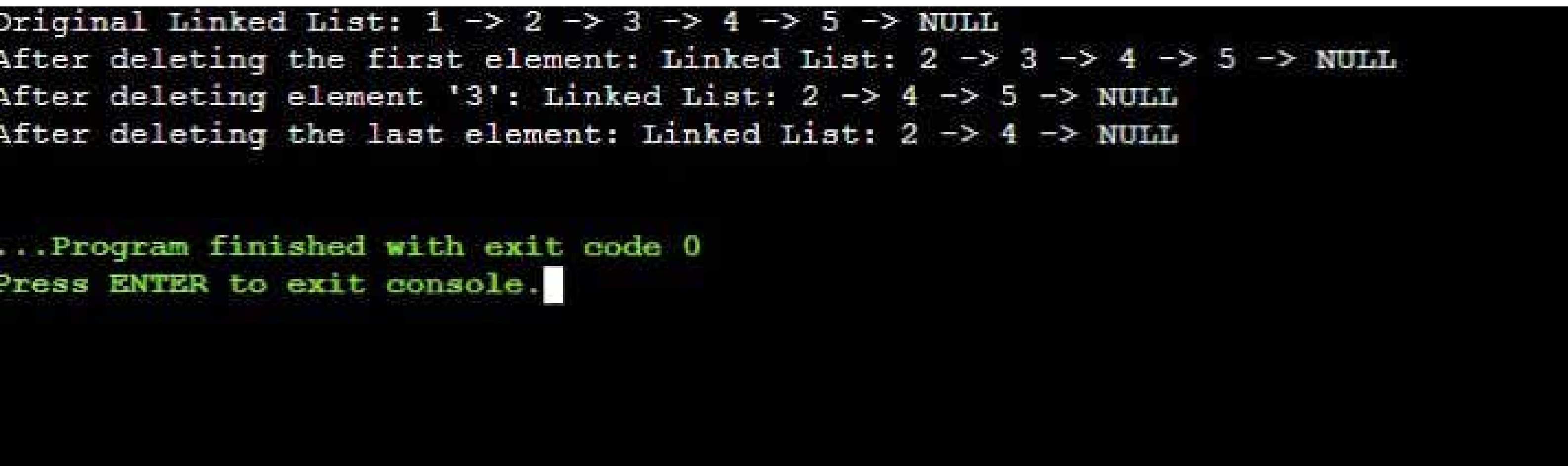
```
Linkedlist :   1 -> 2 -> 3 -> 4 -> 5 -> NULL

LinkedList :   2 -> 3 -> 4 -> 5 -> NULL

Linked list :   2 -> 4 -> 5 -> NULL

LinkedList :   2 -> 4 -> NULL
```

[illegible]
22/1/24


```
Original Linked List: 1 -> 2 -> 3 -> 4 -> 5 -> NULL
After deleting the first element: Linked List: 2 -> 3 -> 4 -> 5 -> NULL
After deleting element '3': Linked List: 2 -> 4 -> 5 -> NULL
After deleting the last element: Linked List: 2 -> 4 -> NULL


...Program finished with exit code 0
Press ENTER to exit console.
```