**Linked List:**

|  |  |  |
| --- | --- | --- |
|  | **Description** | **Status** |
|  | Add a node at the front | **Done** |
|  | Add a node after a given node | **Done** |
|  | Add a node at the end | **Done** |
|  | Find previous node of the node to be deleted | **Done** |
|  | Move last element to first | **Done** |
|  | Free memory for the node to be deleted | **Done** |
|  | Delete a linked list node at the given position | **Done** |
|  | Find Length of a Linked List | **Done** |
|  | Search an element in a Linked List | **Done** |
|  | Write a function that modify head pointer of a Linked List | **Done** |
|  | Swap nodes in a linked list without swapping data |  |
|  | Swap data in a linked list without swapping address | **Done** |
|  | Write a function to get Nth node in a Linked List | **Done** |
|  | Print the middle of a given linked list | **Done** |
|  | Nth node from the end of a Linked List | **Done** |
|  | Write a function to delete a Linked List | **Done** |
|  | Write a function that counts the number of times a given int occurs in a Linked List | **Done** |
|  | Reverse a linked list | **Done** |
|  | Detect loop in a linked list | **Done** |
|  | Merge two sorted linked lists | **Done** |
|  | Given a linked list which is sorted, how will you insert in sorted way | **Done** |
|  | Given only a pointer to a node to be deleted in a singly linked list, how do you delete it? |  |
|  | Function to check if a singly linked list is palindrome | **Done** |
|  | Intersection point of two Linked Lists. | **Done** |
|  | Recursive function to print reverse of a Linked List | **Done** |
|  | Remove duplicates from a sorted linked list | **Done** |
|  | Remove duplicates from an unsorted linked list | **Done** |
|  | Pairwise swap elements of a given linked list |  |
|  | Move last element to front of a given Linked List | **Done** |
|  | Intersection of two Sorted Linked Lists | **Done** |
|  | Delete alternate nodes of a Linked List | **Done** |
|  | Alternating split of a given Singly Linked List | **Done** |
|  | Identical Linked Lists |  |
|  | Merge Sort for Linked Lists |  |
|  | Reverse a Linked List in groups of given size |  |
|  | Reverse alternate K nodes in a Singly Linked List |  |
|  | Delete nodes which have a greater value on right side |  |
|  | Detect and Remove Loop in a Linked List | **Done** |
|  | Add two numbers represented by linked lists | Set 1 |  |
|  | Union and Intersection of two Linked Lists |  |
|  | Find a triplet from three linked lists with sum equal to a given number |  |
|  | Rotate a Linked List |  |
|  | Flattening a Linked List |  |
|  | Add two numbers represented by linked lists |  |
|  | Sort a linked list of 0s, 1s and 2s |  |
|  | Merge two sorted linked lists such that merged list is in reverse order |  |
|  | Compare two strings represented as linked lists |  |
|  | Rearrange a linked list such that all even and odd positioned nodes are together |  |
|  | Rearrange a Linked List in Zig-Zag fashion |  |
|  | Add 1 to a number represented as linked list |  |
|  | Check if a linked list of strings forms a palindrome | **Done** |
|  | Delete last occurrence of an item from linked list | **Done** |
|  | Delete a Linked List node at a given position | **Done** |
|  | Circular Linked List Traversal |  |
|  | Split a Circular Linked List into two halves |  |
|  | Merge a linked list into another linked list at alternate positions |  |
|  | Segregate even and odd nodes in a Linked List |  |
|  | First non-repeating character in a stream |  |
|  | Segregate even and odd nodes in a Linked List |  |
|  | Arrange Consonants and Vowels |  |
|  | Merge List Alternatingly |  |
|  | Identical Linked Lists |  |
|  | Delete Alternate Nodes |  |
|  | Linked List Length Even or Odd? |  |
|  | Linked List that is Sorted Alternatingly |  |
|  | XOR Linked List |  |
|  | Split Singly Linked List Alternatingly |  |
|  | Polynomial Addition |  |
|  |  |  |
|  | Multiply two polynomials II |  |
|  |  | **Done** |
|  |  | **Done** |
|  |  | **Done** |
|  | Multiply two linked lists |  |
|  |  | **Done** |
|  | Subtraction in Linked List |  |
|  |  | **Done** |
|  | Merge K sorted linked lists |  |
|  | Remove duplicate element from sorted Linked List |  |
|  | Insert a node in Doubly linked list |  |
|  | Absolute List Sorting |  |
|  | Implement Queue using Linked List |  |
|  | Implement Stack using Linked List |  |
|  | Flattening a Linked List |  |
|  | Intersection of two sorted Linked lists |  |
|  | Clone a linked list with next and random pointer |  |
|  | Remove duplicates from an unsorted linked list | **Done** |
|  | Split a Circular Linked List into two halves |  |
|  | Linked List in Zig-Zag fashion |  |
|  | Decimal Equivalent of Binary Linked List |  |
|  | Merge two sorted linked lists |  |
|  | Delete Middle of Linked List | **Done** |
|  | Intersection Point in Y Shapped Linked Lists |  |
|  | Rotate a Linked List |  |
|  | Reverse a linked list |  |
|  | Circular Linked List |  |
|  | Finding middle element in a linked list | **Done** |
|  | n'th node from end of linked list | **Done** |
|  | Delete without head pointer |  |
|  | Reverse a Doubly Linked List |  |
|  | Binary Tree to DLL |  |
|  | Reorder List |  |
|  | QuickSort on Doubly Linked List |  |
|  | Delete node in Doubly Linked List |  |
|  | Sorted insert for circular linked list |  |
|  | Rearrange linked list in-place |  |
|  | Delete nodes having greater value on right |  |
|  | Detect Loop in linked list | **Done** |
|  | Merge 2 sorted linked list in reverse order |  |
|  | Compare two linked lists |  |
|  | linked list of strings forms a palindrome |  |
|  | Add 1 to a number represented as linked list |  |
|  | Rearrange a linked list |  |
|  | Add two numbers represented by linked lists | **Done** |
|  | Node at a given index in linked list | **Done** |
|  | Count nodes of linked list | **Done** |
|  | Given a linked list of 0s, 1s and 2s, sort it. |  |
|  | Pairwise swap elements of a linked list by swapping data |  |
|  | Delete N nodes after M nodes of a linked list | **Done** |
|  | Occurence of an integer in a Linked List |  |
|  | Reverse a Linked List in groups of given size. |  |
|  | Given a linked list, reverse alternate nodes and append at the end |  |
|  | Print Linked List elements | **Done** |
|  | Linked List Intersection | **Done** |
|  | Advantages of linked list over array. |  |
|  | Product of two polynomials |  |
|  | Reverse a linked list | **Done** |
|  | Write a list of all data structures and explain each of them along with their applications? |  |
|  | Write algorithm to delete a node in a doubly linked list? |  |
|  | Sorted lists and a God Processor |  |
|  | Delete a node from the linkedlist | **Done** |
|  | Linked list |  |
|  | Producer-Consumer Problem |  |
|  | Heterogenous Linked List - Pointer |  |
|  | Linked list types |  |