Implement doubly linked list

1. create own linkedList with functions add, remove  
2. delete middle node of a linked list.  
3. reverse a string.

Delete the middle node in doubly linked list

Given the head of a linked list, rotate the list to the right by k places.

Input: head : 2->1->3->5->6->4->7

Output: 2->3->6->7->1->5->4

5.Implement an algorithm, Given the head of a linked list, rotate the list to the right by k places.

Input: head = 1->2-.3->4->5, k = 2

Output: 4->5->1->2->3

Input: head = 0->1->2, k = 4

Output: 2->0->1

Given the head of a linked list, return the head of the linked after reversing it.

Given the string str, find whether it was a palindrome or not using stack.

1. WAP to check if two strings are opposite of each other.
2. 2print middle node of a LL
3. What is string
4. What is linked list
5. What are different types of linked list
6. Explain each type in detail
7. What is difference between stack and queue
8. Write code to implement linked list and reverse it and remove duplicate elements from it
9. 1. Valid parentheses -Leetcode.
10. 2. Reverse linked list - Leetcode.
11. 3. Length of the last word - Leetcode.

|  |
| --- |
| 1. Write a program to reverse a linked list ? |
| 2. What is the diffrence between stack and queue ? |
| 3. What is the complexity in insertion in stack ? |
| 4. Insertion in linked list ? |
| 5. What searching algo you will apply ? |
| 6. Find in-defined methods of class string ? |
| 1. sum of array element |
| 2. find the missing number |
| 3. string converted into string with counts e.g:- aabbccc-> a2b2c3 |

1. https://leetcode.com/problems/valid-anagram/

2. https://leetcode.com/problems/intersection-of-two-linked-lists/

3. time and space complexity of above solutions

1)Explain Linked List.

2)Do we have any advantage of using linked list over array.

3)Write a program to implement the stack using singly linked list.

4)Write a program to reverse the alternate nodes in the given singly linked list and append it at the end.

-what is stack  
-what is singly linked list  
-what is circular queue  
-what is doubly linked list ? give an exp  
-what is deque ?  
-what are the operations on stack ?  
  
-Given a singly linked list, delete middle of the linked list. For example, if given linked list is 1->2->3->4->5 then linked list should be modified to 1->2->4->5.  
  
-Given a string, reverse each word in the sentence " welcome to the newton school"

1. check if the string is palindrome or not  
2. Find the mid point of the LL  
3. implementation of Stack

1. Find loop in the linked list.

1) Check if String can be made Palindrome by replacing characters in given pairs

2) program to find the middle node of a LL

Rearrange a Linkedlist

https://practice.geeksforgeeks.org/problems/rearrange-a-linked-list/1

Print all unique words present in a string and remove all occurrences of any duplicate word -

Input - the sun rises in the east

Output - sun

rises

in

east

1) Check if String can be made Palindrome by replacing characters in given pairs

2) program to find the middle node of a LL

Reverse the linked list from the given left value to right value.

What is linked list and types

What is stack and queue and their operations

Program to check string is palindrome or not

Implement linked list and reverse

2. Check if two strings are anagrams. Balance parenthesis

"1. Explain Circular Linked List

2. Applications of Queue

3. What will be the output of following code

void fun1(struct node\* head)

{

if(head == NULL)

return;

fun1(head->next);

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

}

4. Suppose a stack is to be implemented with a linked list

instead of an array. What would be the effect on the time

complexity of the push and pop operations of the stack

implemented using linked list (Assuming stack is

implemented efficiently)

5. What will be the output of following code

String s1 = ""abc"";

String s2 = new String(""abc"");

System.out.println(s1==s2)

6. Remove duplicates from a given string

7. Given the head of a singly linked list, return true if it is a palindrome "

1. Longest substring with a count of 0s more than 1s

Given a binary string find the longest substring which contains 0’s more than 1’s.

Examples:

Input: 1010. Longest substring with a count of 0s more than 1s

Given a binary string find the longest substring which contains 0’s more than 1’s.

Examples:

Output : 3

Input: 1010

Substring 010 has 0 occurring more number of times than 1.

Output : 3

Input: 101100

Substring 010 has 0 occurring more number of times than 1.

Output : 5

Input: 101100

Substring 01100 has 0 occurring more number of times than 1.

Output : 5

2. Print All Possible Valid Combinations Of Parenthesis of Given ‘N’

Substring 01100 has 0 occurring more number of times than 1.

Objec­tive: – Given “n”, generate all valid parenthesis strings of length “2n”.

2. Print All Possible Valid Combinations Of Parenthesis of Given ‘N’

Example:Objec­tive: – Given “n”, generate all valid parenthesis strings of length “2n”.

Given n=2

Example:Output:

Given n=2

(())

Output:

()()

(())

For n=3

()()

Output:

For n=3

((()))

Output:

(()()) ((())) (())() (()()) ()(()) (())() ()()() ()(())

Problem 1:

Given a string containing just the characters '(' and ')',

find the length of the longest valid (well-formed) parentheses substring.

Example 1:

Input: s = "(()"

Output: 2

Explanation: The longest valid parentheses substring is "()".

Example 2:

Input: s = ")()())"

Output: 4

Explanation: The longest valid parentheses substring is "()()".

Example 3:

Input: s = ""

Output: 0

Problem 2:

Write a function detectAndRemoveLoop() that checks whether a given Linked List contains a loop and

if loop is present then removes the loop and returns true.

If the list doesn’t contain a loop then it returns false.