Doubly and Circular linked list

Assignment Solutions

PW SKILLS

1. Implement doubly linked list as a class with functions to insert a node and traverse the list in bot forward and reverse direction.

Ans: Code

2. Given the head of a doubly linked list, remove all the nodes from the linked list that have the value x stored in them.The first line of input contains n (the size of the linked list) and x.The second line of input contains the elements of the linked list.
INPUT
4
31 3 2 3
OUTPUT
1 2

Ans: Code

Explanation:
a. Shift the head forward till it contains 'x'. This will ensure that the head doesn't contains 'x'.
b. Traverse the list and remove the node that has the value 'x'.

3. Given the head of a doubly linked list which stores only binary values i.e. 1's and 0's. In the linked list make sure that between every 2 nodes containing value 1, there exist an even number of 0's. If the number of 0's is odd then insert a node with value 0 in between the two 1's. Assume that the first and the last node contains value 1.
The first line of input contains n, the size of the linked list.The second line of input contains the elements of the linked list.
INPUT
6
1 0 0 1 0 1
OUTPUT
1->0->0->1->0->0->1
INPUT
5
1 1 0 1 1
OUTPUT
1->1->0->0->1->1

Ans: Ans: Code

Explanation:
a. Make a counter to keep store of the number of times we encounter 0.
b. Traverse the list. At each iteration-
i. If you find the value 1, see how many times you have encountered the value 0 since the last 1.
1. If it is odd, then add a 0 before the current 1 so that we have an even number of zeros between the current and the previous 1.
Reset the counter.
ii. Else, increment the counter by 1.

4. You are given two non-empty doubly linked lists representing two non-negative integers, and each of their nodes contains a single digit. Subtract the two numbers and return the difference as a linked list. It is guaranteed that the first number will always be greater than the second number.The first line of input contains n and m, the size of the first and second linked list respectively.The second line of input contains the elements of the first linked list.The third line of input contains the elements of the second linked list.
Input
3 2
8 2 3
3 4
Output
7 8 9

Ans: You are given two non-empty linked lists representing two non-negative integers, and each of their nodes contains a single digit. Subtract the two numbers and return the difference as a linked list. It is guaranteed that the first number will always be greater than the second number.

Code

Explanation:

a. Start your traversal from the tail of the linked lists.
b. Let val1 and val2 be the values stored in the two linked lists at the current position.
c. In case we have already exhausted the second list, we can assume that the current value of the second list to be 0.
d. If the value in the first list is less than the value in the second list, then we can borrow from the next value in the first list. (it is guaranteed that the first list will have a bigger number compared to the second list so we can always take borrow)
e. Store the difference between the values in the first list.
f. Move to the next values.

5. Given the head of a doubly linked list. Divide the list into 2 parts- one containing odd values and the other containing even values. Return the array containing the two linked lists. The order of the nodes must be maintained.The first line of input contains n, the size of the linked list.The second line of input contains the elements of the linked list.
Input
61 10 2 5 3 4
Output: 1→5→3→NULL
10→2→4→NULL

Ans: Code

Explanation:

a. Make 2 new linked lists to store the odd and even numbers respectively.
b. Depending upon if the current value is odd or even, insert the node into the respective list.
c. Since here we are not generating new nodes while inserting them into the list and making updates, we need to store the next element in the sequence before the insertion.
Given the head of a doubly linked list and a pointer to a node present in the list. Delete the node from the list.

6. Given the head of a doubly linked list and a pointer to a node present in the list. Delete the node from the list.
Input
1 3 2 4
Node = 2
Output: 1→3→4→NULL

Ans: Code

Explanation:
a. If the node to be deleted is head, then remove the current node.
b. Else, find the previous and next node to the current node and delete the current node with their help.
Given the head of a doubly linked list. Find the number of non-decreasing sublists. A sublist is a part of a linked list that can be created by removing some number of elements from the beginning of the list and the end of the list.

7. Given the head of a doubly linked list. Find the number of non-decreasing sublists. A sublist is a part of a linked list that can be created by removing some number of elements from the beginning of the list and the end of the list. Note: Each sublist should contain at least 2 members.The first line of input contains n, the size of the linked list.The second line of input contains the elements of the linked list.
Input
6
3 4 5 2 1 4
Output
3
Explanation:The 3 non-decreasing sublists are-3->4->54→51->4

Ans: Code

Explanation:
a. Since there will be at least 1 non-decreasing sequence, initialize the counter by 1.
b. Traverse the list starting from the second node.
i. If the data in the current node is less than the data in the previous node, then increment the counter by 1 i.e. a new non-decreasing sequence is starting from the current position.

8. Given a sorted doubly linked list, find the number of triplets of distinct nodes such that their sum is equal to x.
The first line of input contains n, the size of the linked list, and the value of x.
The second line of input contains the elements of the linked list.
Input
5 4
1 2 1 1 2
Output
6

Ans: Code

Explanation:
To solve this problem we are going to use the brute force method.
We are going to use 3 pointers for each of the 3 values that we need.

a. Initialise the first pointer with the head and start its traversal.
b. Initialise the second pointer with the value just after the first pointer and start its traversal.
c. Initialise the third pointer with the value just after the second pointer and start its traversal.
d. If the sum of three values pointed by the pointers is x then increment the counter.

9. Implement the class of a circular linked list with "insert(int val)", and "print()" functions.
Ans: Code

10 .You are given a circular linked list in the form of a string. Each node of the linked list contains either a dot('.') or a star('*'). Your task is to make sure that between every consecutive dot there are not more than 2 stars. For this you can delete a star in between any 2 dots. Count the minimum number of stars that you will have to delete.
It is guaranteed that there is at least 1 dot character.
The first line of input contains a string s, the nodes of the linked list.
INPUT.
.*.*
OUTPUT
3

Ans: Code

Explanation:
a. Since we have to deal with stars, keep moving forward till you encounter a star.
b. Make a counter to keep track of the number of stars encountered.
c. Start the traversal.
    i. If the current node contains a dot.
        1. If the counter is more than 2, then we need to delete all the nodes that have a star, except for the last 2.
        Reset the counter
    ii. Else, increment the counter.

11. You are given a circular linked list in the form of a string. Each node contains either a dot('.') or a star('*'). Your task is to make sure that between every consecutive dot there are at least 2 stars. For this you can add a star in between any 2 dots. Count the minimum number of stars that you will have to add.
It is guaranteed that there is at least 1 dot character.
The first line of input contains a string s, the nodes of the linked list.
INPUT
.*.**.
OUTPUT
3

Ans: Code

Explanation:
a. Since we have to deal with stars, keep moving forward till you encounter a star.
b. Make a counter to keep track of the number of stars encountered.
c. Start the traversal.
    i. If the current node contains a dot.
        1. If the counter is less than 2, then we need to add nodes that have a star till the counter becomes equal to 2.

Reset the counter

ii. Else, increment the counter.

12. You are given a circular linked list in the form of a string. Each node contains either a dot('.') or a star('*'). Your task is to make sure that between every consecutive dot there are exactly 2 stars. For this you can add or delete a star in between any 2 dots. Return the updated linked list.
It is guaranteed that there is at least 1 dot character.
The first line of input contains a string s, the nodes of the linked list.
INPUT
.*.*.**
OUTPUT.→*→*→.→*→*→.→*→*

Ans: Code

Explanation:

a. Use 2 pointers to keep track of the current and previous dot nodes.
b. Start your traversal.
c. Keep moving forward till you reach the consecutive dot node and count the number of stars along the way.
d. If the count is less than 2, then add star nodes between the 2 dot nodes till there are exactly 2 star nodes between them.
e. Else if the count is greater than 2, remove the star nodes between the 2 dot nodes till there are exactly 2 star nodes between them.

13. You are given a circular linked list with integer values. Your task is to find the sum of values present at opposite positions.
It is guaranteed that the size of the linked list is even.
The first line of input contains n, the size of the linked list.
The second line of input contains the elements of the linked list.
INPUT
6
1 2 3 7 8 9
OUTPUT

Ans: Code

Explanation:

a. Use the slow and fast pointer method to get the middle element of the list.
b. Now traverse the list normally with the two pointers to get the respective sums.