Linked List-2

1. **Reverse LL (Recursive)**

**Send Feedback**

Given a singly linked list of integers, reverse it using recursion and return the head to the modified list.

**Note :**

No need to print the list, it has already been taken care. Only return the new head to the list.

**Input format :**

The first line contains an Integer 't' which denotes the number of test cases or queries to be run. Then the test cases follow.

The first and the only line of each test case or query contains the elements of the singly linked list separated by a single space.

**Remember/Consider :**

While specifying the list elements for input, -1 indicates the end of the singly linked list and hence, would never be a list element

**Output format :**

For each test case/query, print the elements of the updated singly linked list.

Output for every test case will be printed in a seperate line.

**Constraints :**

1 <= t <= 10^2

0 <= M <= 10^4

Where M is the size of the singly linked list.

Time Limit: 1sec

**Sample Input 1 :**

1

1 2 3 4 5 6 7 8 -1

**Sample Output 1 :**

8 7 6 5 4 3 2 1

**Sample Input 2 :**

2

10 -1

10 20 30 40 50 -1

**Sample Output 2 :**

10

50 40 30 20 10

1. **Reverse LL (Iterative)**

**Send Feedback**

Given a linked list, reverse it iteratively.

**You don't need to print the elements, just reverse the LL duplicates and return the head of updated LL.**

Input format : Linked list elements (separated by space and terminated by -1)`

**Sample Input 1 :**

1 2 3 4 5 -1

**Sample Output 1 :**

5 4 3 2 1

1. **Midpoint of Linked list**

**Send Feedback**

For a given singly linked list of integers, find and return the node present at the middle of the list.

**Note :**

If the length of the singly linked list is even, then return the first middle node.

Example: Consider, 10 -> 20 -> 30 -> 40 is the given list, then the nodes present at the middle with respective data values are, 20 and 30. We return the first node with data 20.

**Input format :**

The first line contains an Integer 't' which denotes the number of test cases or queries to be run. Then the test cases follow.

The first and the only line of each test case or query contains the elements of the singly linked list separated by a single space.

**Remember/Consider :**

While specifying the list elements for input, -1 indicates the end of the singly linked list and hence, would never be a list element

**Output Format :**

For each test case/query, print the data value of the node at the middle of the given list.

Output for every test case will be printed in a seperate line.

**Constraints :**

1 <= t <= 10^2

0 <= M <= 10^5

Where M is the size of the singly linked list.

Time Limit: 1sec

**Sample Input 1 :**

1

1 2 3 4 5 -1

**Sample Output 1 :**

3

**Sample Input 2 :**

2

-1

1 2 3 4 -1

**Sample Output 2 :**

2

1. **Code : Merge two sorted LL**

**Send Feedback**

You have been given two sorted(in ascending order) singly linked lists of integers.

Write a function to merge them in such a way that the resulting singly linked list is also sorted(in ascending order) and return the new head to the list.

**Note :**

Try solving this in O(1) auxiliary space.

No need to print the list, it has already been taken care.

**Input format :**

The first line contains an Integer 't' which denotes the number of test cases or queries to be run. Then the test cases follow.

The first line of each test case or query contains the elements of the first sorted singly linked list separated by a single space.

The second line of the input contains the elements of the second sorted singly linked list separated by a single space.

**Remember/Consider :**

While specifying the list elements for input, -1 indicates the end of the singly linked list and hence, would never be a list element

**Output :**

For each test case/query, print the resulting sorted singly linked list, separated by a single space.

Output for every test case will be printed in a seperate line.

**Constraints :**

1 <= t = 10^2

0 <= N <= 10 ^ 4

0 <= M <= 10 ^ 4

Where N and M denote the sizes of the singly linked lists.

Time Limit: 1sec

**Sample Input 1 :**

1

2 5 8 12 -1

3 6 9 -1

**Sample Output 1 :**

2 3 5 6 8 9 12

**Sample Input 2 :**

2

2 5 8 12 -1

3 6 9 -1

10 40 60 60 80 -1

10 20 30 40 50 60 90 100 -1

**Sample Output 2 :**

2 3 5 6 8 9 12

10 10 20 30 40 40 50 60 60 60 80 90 100

1. **Code : Merge Sort**

**Send Feedback**

 Given a singly linked list of integers, sort it using 'Merge Sort.'

**Note :**

No need to print the list, it has already been taken care. Only return the new head to the list.

**Input format :**

The first line contains an Integer 't' which denotes the number of test cases or queries to be run. Then the test cases follow.

The first and the only line of each test case or query contains the elements of the singly linked list separated by a single space.

**Remember/Consider :**

While specifying the list elements for input, -1 indicates the end of the singly linked list and hence, would never be a list element

**Output format :**

For each test case/query, print the elements of the sorted singly linked list.

Output for every test case will be printed in a seperate line.

**Constraints :**

1 <= t <= 10^2

0 <= M <= 10^5

Where M is the size of the singly linked list.

Time Limit: 1sec

**Sample Input 1 :**

1

10 9 8 7 6 5 4 3 -1

**Sample Output 1 :**

3 4 5 6 7 8 9 10

**Sample Output 2 :**

2

-1

10 -5 9 90 5 67 1 89 -1

**Sample Output 2 :**

-5 1 5 9 10 67 89 90

MCQ’s

1. **Circular doubly LL**

**Send Feedback**

Given an unsorted circular doubly linked list, suppose you have reference (or pointer) to its head node only, which of the following operation can be implemented in O(1) time?

i) Insertion at the front of the linked list

ii) Insertion at the end of the linked list

iii) Deletion of the last node of the linked list

iv) Deletion of the front node of the linked list

1. I and II
2. I, II and III
3. I, II, III and IV answer
4. None
5. **Traversal in LL**

**Send Feedback**

In doubly linked lists, traversal can be done in ?

1. Only forward Direction
2. Only Reverse Direction
3. Both Directions
4. None of the Above
5. **Double LL**

**Send Feedback**

Given an unsorted doubly Linked List, suppose you have references (or pointer) to its head and tail nodes, which of the following operation can be implemented in O(1) time ?

i) Insertion at the front of the linked list

ii) Insertion at the end of the linked list

iii) Deletion of the last node of the linked list

iv) Deletion of the front node of the linked list

1. I and II
2. I and III
3. I, II and III
4. I, II, III and IV answer
5. **Circular LL**

**Send Feedback**

Given an unsorted circular linked list, suppose you have reference (or pointer) to its head node only, which of the following operation can be implemented in O(1) time?

i) Insertion at the front of the linked list

ii) Insertion at the end of the linked list

iii) Deletion of the last node of the linked list

iv) Deletion of the front node of the linked list

1. I and II
2. I and III
3. I, II and III
4. None answer

Assignment

1. **Find a node in LL (recursive)**

**Send Feedback**

Given a singly linked list of integers and an integer n, find and return the index for the first occurrence of 'n' in the linked list. -1 otherwise.

Follow a recursive approach to solve this.

**Note :**

Assume that the Indexing for the linked list always starts from 0.

**Input format :**

The first line contains an Integer 't' which denotes the number of test cases or queries to be run. Then the test cases follow.

The first line of each test case or query contains the elements of the singly linked list separated by a single space.

The second line of input contains a single integer depicting the value of 'n'.

**Remember/Consider :**

While specifying the list elements for input, -1 indicates the end of the singly linked list and hence, would never be a list element

**Output format :**

For each test case/query, print the elements of the updated singly linked list.

Output for every test case will be printed in a seperate line.

**Constraints :**

1 <= t <= 10^2

0 <= M <= 10^5

Where M is the size of the singly linked list.

Time Limit: 1sec

**Sample Input 1 :**

1

3 4 5 2 6 1 9 -1

100

**Sample Output 1 :**

-1

**Sample Input 2 :**

2

10 20010 30 400 600 -1

20010

100 200 300 400 9000 -34 -1

-34

**Sample Output 2 :**

1

5

1. **Even after Odd LinkedList**

**Send Feedback**

For a given singly linked list of integers, arrange the elements such that all the even numbers are placed after the odd numbers. The relative order of the odd and even terms should remain unchanged.

**Note :**

No need to print the list, it has already been taken care. Only return the new head to the list.

**Input format:**

The first line contains an Integer 't' which denotes the number of test cases or queries to be run. Then the test cases follow.

The first line of each test case or query contains the elements of the singly linked list separated by a single space.

**Remember/Consider :**

While specifying the list elements for input, -1 indicates the end of the singly linked list and hence, would never be a list element

**Output format:**

For each test case/query, print the elements of the updated singly linked list.

Output for every test case will be printed in a seperate line.

**Constraints :**

1 <= t <= 10^2

0 <= M <= 10^5

Where M is the size of the singly linked list.

Time Limit: 1sec

**Sample Input 1 :**

1

1 4 5 2 -1

**Sample Output 1 :**

1 5 4 2

**Sample Input 2 :**

2

1 11 3 6 8 0 9 -1

10 20 30 40 -1

**Sample Output 2 :**

1 11 3 9 6 8 0

10 20 30 40

1. **Delete every N nodes**

**Send Feedback**

You have been given a singly linked list of integers along with two integers, 'M,' and 'N.' Traverse the linked list such that you retain the 'M' nodes, then delete the next 'N' nodes. Continue the same until the end of the linked list.

To put it in other words, in the given linked list, you need to delete N nodes after every M nodes.

**Note :**

No need to print the list, it has already been taken care. Only return the new head to the list.

**Input format :**

The first line contains an Integer 't' which denotes the number of test cases or queries to be run. Then the test cases follow.

The first line of each test case or query contains the elements of the singly linked list separated by a single space.

The second line of input contains two integer values 'M,' and 'N,' respectively. A single space will separate them.

**Remember/Consider :**

While specifying the list elements for input, -1 indicates the end of the singly linked list and hence, would never be a list element

**Output format :**

For each test case/query, print the elements of the updated singly linked list.

Output for every test case will be printed in a seperate line.

**Constraints :**

1 <= t <= 10^2

0 <= P <= 10^5

Where P is the size of the singly linked list.

0 <= M <= 10^5

0 <= N <= 10^5

Time Limit: 1sec

**Sample Input 1 :**

1

1 2 3 4 5 6 7 8 -1

2 2

**Sample Output 1 :**

1 2 5 6

**Sample Input 2 :**

2

10 20 30 40 50 60 -1

0 1

1 2 3 4 5 6 7 8 -1

2 3

**Sample Output 2 :**

1 2 6 7

**Explanation of Sample Input 2 :**

For the first query, we delete one node after every zero elements hence removing all the items of the list. Therefore, nothing got printed.

For the second query, we delete three nodes after every two nodes, resulting in the final list, 1 -> 2 -> 6 -> 7.

1. **Swap two Nodes of LL**

**Send Feedback**

You have been given a singly linked list of integers along with two integers, 'i,' and 'j.' Swap the nodes that are present at the 'i-th' and 'j-th' positions.

**Note :**

Remember, the nodes themselves must be swapped and not the datas.

No need to print the list, it has already been taken care. Only return the new head to the list.

**Input format :**

The first line contains an Integer 't' which denotes the number of test cases or queries to be run. Then the test cases follow.

The first line of each test case or query contains the elements of the singly linked list separated by a single space.

The second line of input contains two integer values 'i,' and 'j,' respectively. A single space will separate them.

**Remember/consider :**

While specifying the list elements for input, -1 indicates the end of the singly linked list and hence, would never be a list element

**Output format :**

For each test case/query, print the elements of the updated singly linked list.

Output for every test case will be printed in a seperate line.

**Constraints :**

1 <= t <= 10^2

0 <= M <= 10^5

Where M is the size of the singly linked list.

0 <= i < M

0 <= j < M

Time Limit: 1sec

**Sample Input 1 :**

1

3 4 5 2 6 1 9 -1

3 4

**Sample Output 1 :**

3 4 5 6 2 1 9

**Sample Input 2 :**

2

10 20 30 40 -1

1 2

70 80 90 25 65 85 90 -1

0 6

**Sample Output 2 :**

10 30 20 40

90 80 90 25 65 85 70

1. **kReverse**

**Send Feedback**

Given a singly linked list of integers, reverse the nodes of the linked list 'k' at a time and return its modified list.

 'k' is a positive integer and is less than or equal to the length of the linked list. If the number of nodes is not a multiple of 'k,' then left-out nodes, in the end, should be reversed as well.

**Example :**

Given this linked list: 1 -> 2 -> 3 -> 4 -> 5

For k = 2, you should return: 2 -> 1 -> 4 -> 3 -> 5

For k = 3, you should return: 3 -> 2 -> 1 -> 4 -> 5

**Note :**

No need to print the list, it has already been taken care. Only return the new head to the list.

**Input format :**

The first line contains an Integer 't' which denotes the number of test cases or queries to be run. Then the test cases follow.

The first line of each test case or query contains the elements of the singly linked list separated by a single space.

The second line of input contains a single integer depicting the value of 'k'.

**Remember/Consider :**

While specifying the list elements for input, -1 indicates the end of the singly linked list and hence, would never be a list element

**Output format :**

For each test case/query, print the elements of the updated singly linked list.

Output for every test case will be printed in a seperate line.

**Constraints :**

1 <= t <= 10^2

0 <= M <= 10^5

Where M is the size of the singly linked list.

0 <= k <= M

Time Limit: 1sec

**Sample Input 1 :**

1

1 2 3 4 5 6 7 8 9 10 -1

4

**Sample Output 1 :**

4 3 2 1 8 7 6 5 10 9

**Sample Input 2 :**

2

1 2 3 4 5 -1

0

10 20 30 40 -1

4

**Sample Output 2 :**

1 2 3 4 5

40 30 20 10

1. **Bubble Sort (Iterative) LinkedList**

**Send Feedback**

Given a singly linked list of integers, sort it using 'Bubble Sort.'

**Note :**

No need to print the list, it has already been taken care. Only return the new head to the list.

**Input format :**

The first line contains an Integer 't' which denotes the number of test cases or queries to be run. Then the test cases follow.

The first and the only line of each test case or query contains the elements of the singly linked list separated by a single space.

**Remember/Consider :**

While specifying the list elements for input, -1 indicates the end of the singly linked list and hence, would never be a list element

**Output format :**

For each test case/query, print the elements of the sorted singly linked list.

Output for every test case will be printed in a seperate line.

**Constraints :**

1 <= t <= 10^2

0 <= M <= 10^5

Where M is the size of the singly linked list.

Time Limit: 1sec

**Sample Input 1 :**

1

10 9 8 7 6 5 4 3 -1

**Sample Output 1 :**

3 4 5 6 7 8 9 10

**Sample Output 2 :**

2

-1

10 -5 9 90 5 67 1 89 -1

**Sample Output 2 :**

-5 1 5 9 10 67 89 90