Linked List-1

1. **Predict The Output**

**Send Feedback**

What will be the output?

class Node:

def \_\_init\_\_(self, data):

self.data = data

self.next = None

def printLL(head):

while head is not None:

print(head.data,end=" ")

head = head.next

node1 = Node(10)

node2 = Node(20)

node2.next = node1

printLL(node2)

1. 10 20
2. 20 10 correct answer
3. Error
4. None of the above
5. **Predict The Output**

**Send Feedback**

What will be the output?

class Node:

def \_\_init\_\_(self, data):

self.data = data

self.next = None

def printLL(head):

while head is not None:

print(head.data,end=" ")

head = head.next

node1 = Node(10)

node2 = Node(20)

node3 = Node(30)

node4 = Node(40)

node1.next = node2

node2.next = node3

node3.next = node4

printLL(node2)

1. 10 20 30 40
2. 20 30 40 answer
3. 30 40
4. 10 30 40
5. **Predict The Output**

**Send Feedback**

What will be the output?

class Node:

def \_\_init\_\_(self, data):

self.data = data

self.next = None

def printLL(head):

while head is not None:

print(head.data,end=" ")

head = head.next

def increment(head):

temp = head

while temp is not None:

temp.data +=1

temp = temp.next

node1 = Node(10)

node2 = Node(20)

node1.next = node2

increment(node1)

printLL(node1)

1. 10 20
2. 11 21 answer
3. Error
4. None of the Above

**1 Length of LL**

**Send Feedback**

#### For a given singly linked list of integers, find and return its length. Do it using an iterative method.

##### 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.

First and the only line of each test case or query contains 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, print the length of the linked list.

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

##### Constraints :

1 <= t <= 10^2

0 <= N <= 10^5

Time Limit: 1sec

##### Sample Input 1 :

1

3 4 5 2 6 1 9 -1

##### Sample Output 1 :

7

##### Sample Input 2 :

2

10 76 39 -3 2 9 -23 9 -1

-1

##### Sample Output 2 :

8

0

**2. Print ith node**

**Send Feedback**

#### For a given a singly linked list of integers and a position 'i', print the node data at the 'i-th' position.

##### Note :

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

If the given position 'i', is greater than the length of the given singly linked list, then don't print anything.

##### 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 contains the value of 'i'. It denotes the position in the given singly linked list.

##### 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, print the node data at the 'i-th' position of the linked list(if exists).

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

##### Constraints :

1 <= t <= 10^2

0 <= N <= 10^5

i >= 0

Time Limit: 1sec

##### Sample Input 1 :

1

3 4 5 2 6 1 9 -1

3

##### Sample Output 1 :

2

##### Sample Input 2 :

2

3 4 5 2 6 1 9 -1

0

9 8 4 0 7 8 -1

3

##### Sample Output 2 :

3

0

1. **Delete node**

**Send Feedback**

#### You have been given a linked list of integers. Your task is to write a function that deletes a node from a given position, 'pos'.

##### Note :

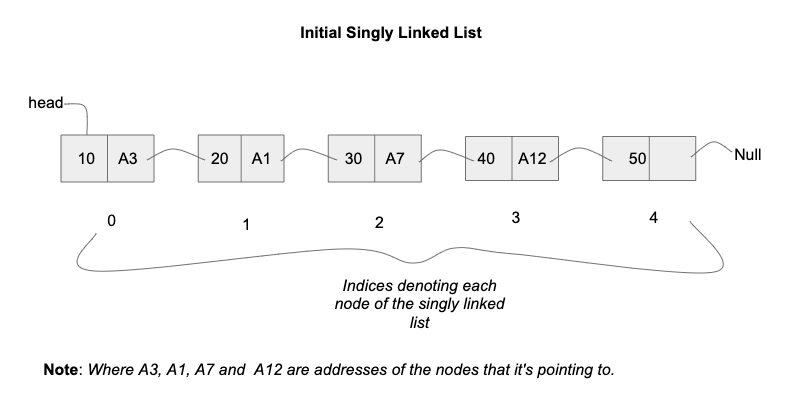
Assume that the Indexing for the linked list always starts from 0.

If the position is greater than or equal to the length of the linked list, you should return the same linked list without any change.

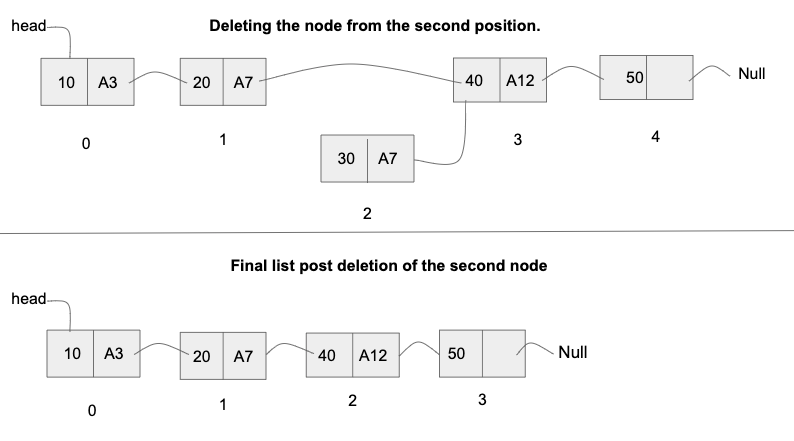
##### Illustration :

The following images depict how the deletion has been performed.

#### Image-I :



#### Image-II :



##### 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 linked list separated by a single space.

The second line of each test case contains the integer value of 'pos'. It denotes the position in the linked list from where the node has to be deleted.

##### 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 resulting linked list of integers in a row, separated by a single space.

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

You don't need to print explicitly, it has been taken care of.

##### Constraints :

1 <= t <= 10^2

0 <= N <= 10^5

pos >= 0

Time Limit: 1sec

Where 'N' is the size of the singly linked list.

##### Sample Input 1 :

1

3 4 5 2 6 1 9 -1

3

##### Sample Output 1 :

3 4 5 6 1 9

##### Sample Input 2 :

2

3 4 5 2 6 1 9 -1

0

10 20 30 40 50 60 -1

7

##### Sample Output 2 :

4 5 2 6 1 9

10 20 30 40 50 60

**4.Length of LL (recursive)**

**Send Feedback**

#### Given a linked list, find and return the length of input LL recursively.

##### Input format :

Linked list elements (separated by space and terminated by -1)

##### Output format :

Length of LL

##### Sample Input :

3 4 5 2 6 1 9 -1

##### Sample Output :

7

**5.Delete node (recursive)**

**Send Feedback**

#### Given a singly linked list of integers and position 'i', delete the node present at the 'i-th' position in the linked list recursively.

##### Note :

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

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 'i'.

##### 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

Time Limit: 1sec

##### Sample Input 1 :

1

3 4 5 2 6 1 9 -1

3

##### Sample Output 1 :

3 4 5 6 1 9

##### Sample Input 2 :

2

30 -1

0

10 20 30 50 60 -1

4

##### Sample Output 2 :

10 20 30 50

MCQ

1. **Search in LL**

**Send Feedback**

What will be the time complexity of searching an element in the linked list ?

1. O(1)
2. O(n) answer
3. O(nlogn)
4. O(n^2)
5. **Add element at last**

**Send Feedback**

Consider the Singly linked list having n elements. What will be the time taken to add an node at the end of linked list if is initially pointing to first node of the list.

That is only head is given to you.

1. O(n) answer
2. O(1)
3. O(nlogn)
4. O(n^2)
5. **Insert at 2nd position**

**Send Feedback**

There is reference (or pointer) to first Node of the Linked List, then time required to insert element to second position is \_\_\_\_\_\_\_\_\_\_.

Indexing starts from 0.

1. O(1) answer
2. O(n)
3. O(nlogn)
4. O(n^2)
5. **Operations in O(1)**

**Send Feedback**

Given an unsorted singly 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 IV answer
3. I, II and III
4. I, II and III
5. **Operations in O(1)**

**Send Feedback**

Given an unsorted singly 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 and IV answer

Assignment

1. **Find a Node in Linked List**

**Send Feedback**

#### You have been given a singly linked list of integers. Write a function that returns the index/position of an integer data denoted by 'N' (if it exists). Return -1 otherwise.

##### Note :

Assume that the Indexing for the singly 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 contains the integer value 'N'. It denotes the data to be searched in the given singly linked list.

##### 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, return the index/position of 'N' in the singly linked list. Return -1, otherwise.

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

##### Constraints :

1 <= t <= 10^2

0 <= M <= 10^5

Where 'M' is the size of the singly linked list.

Time Limit: 1 sec

##### Sample Input 1 :

2

3 4 5 2 6 1 9 -1

5

10 20 30 40 50 60 70 -1

6

##### Sample Output 1 :

2

-1

##### Sample Input 2 :

1

3 4 5 2 6 1 9 -1

6

##### Sample Output 2 :

4

##### Explanation for Sample Input 2 :

For the given singly linked list, considering the indices starting from 0, progressing in a left to right manner with a jump of 1, then the N = 6 appears at position 4.

1. **AppendLastNToFirst**

**Send Feedback**

#### You have been given a singly linked list of integers along with an integer 'N'. Write a function to append the last 'N' nodes towards the front of the singly linked list and returns 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 contains the integer value 'N'. It denotes the number of nodes to be moved from last to the front of the singly linked list.

##### 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 resulting singly linked list of integers in a row, separated by a single space.

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

##### Constraints :

1 <= t <= 10^2

0 <= M <= 10^5

0 <= N < M

Time Limit: 1sec

Where 'M' is the size of the singly linked list.

##### Sample Input 1 :

2

1 2 3 4 5 -1

3

10 20 30 40 50 60 -1

5

##### Sample Output 1 :

3 4 5 1 2

20 30 40 50 60 10

##### Sample Input 2 :

1

10 6 77 90 61 67 100 -1

4

##### Sample Output 2 :

90 61 67 100 10 6 77

##### Explanation to Sample Input 2 :

We have been required to move the last 4 nodes to the front of the list. Here, "90->61->67->100" is the list which represents the last 4 nodes. When we move this list to the front then the remaining part of the initial list which is, "10->6->77" is attached after 100. Hence, the new list formed with an updated head pointing to 90.

1. **Eliminate duplicates from LL**

**Send Feedback**

#### You have been given a singly linked list of integers where the elements are sorted in ascending order. Write a function that removes the consecutive duplicate values such that the given list only contains unique elements and returns the head to the updated 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(in ascending order) 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 resulting singly linked list of integers in a row, separated by a single space.

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

##### Constraints :

1 <= t <= 10^2

0 <= M <= 10^5

Time Limit: 1sec

Where 'M' is the size of the singly linked list.

##### Sample Input 1 :

1

1 2 3 3 4 3 4 5 4 5 5 7 -1

##### Sample Output 1 :

1 2 3 4 3 4 5 4 5 7

##### Sample Input 2 :

2

10 20 30 40 50 -1

10 10 10 10 -1

##### Sample Output 2 :

10 20 30 40 50

10

1. **Print Reverse LinkedList**

**Send Feedback**

#### You have been given a singly linked list of integers. Write a function to print the list in a reverse order.

#### To explain it further, you need to start printing the data from the tail and move towards the head of the list, printing the head data at the end.

##### Note :

You can’t change any of the pointers in the linked list, just print it in the reverse order.

##### 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/Constraints :

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, print the singly linked list of integers in a reverse fashion, in a row, separated by a single space.

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

##### Constraints :

1 <= t <= 10^2

0 <= M <= 10^3

Time Limit: 1sec

Where 'M' is the size of the singly linked list.

##### Sample Input 1 :

1

1 2 3 4 5 -1

##### Sample Output 1 :

5 4 3 2 1

##### Sample Input 2 :

2

1 2 3 -1

10 20 30 40 50 -1

##### Sample Output 2 :

3 2 1

50 40 30 20 10

1. **Palindrome LinkedList**

**Send Feedback**

#### You have been given a head to a singly linked list of integers. Write a function check to whether the list given is a 'Palindrome' or not.

##### 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.

First and the only line of each test case or query contains the 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, the only line of output that print 'true' if the list is Palindrome or 'false' otherwise.

##### Constraints :

1 <= t <= 10^2

0 <= M <= 10^5

Time Limit: 1sec

Where 'M' is the size of the singly linked list.

##### Sample Input 1 :

1

9 2 3 3 2 9 -1

##### Sample Output 1 :

true

##### Sample Input 2 :

2

0 2 3 2 5 -1

-1

##### Sample Output 2 :

false

true

##### Explanation for the Sample Input 2 :

For the first query, it is pretty intuitive that the the given list is not a palindrome, hence the output is 'false'.

For the second query, the list is empty. An empty list is always a palindrome , hence the output is 'true'.