Question:

Given a Linked List of integers, write a function to modify the linked list such that all even numbers appear before all the odd numbers in the modified linked list. Also, keep the order of even and odd numbers the same.

Examples:

Input: 17->15->8->12->10->5->4->1->7->6->NULL
Output: 8->12->10->4->6->17->15->5->1->7->NULL

Input: 8->12->10->5->4->1->6->NULL
Output: 8->12->10->4->6->5->1->NULL

// If all numbers are even then do not change the list
Input: 8->12->10->NULL
Output: 8->12->10->NULL

// If all numbers are odd then do not change the list
Input: 1->3->5->7->NULL
Output: 1->3->5->7->NULL

##

Question:

Given two lists sorted in increasing order, create and return a new list representing the intersection of the two lists. The new list should be made with its own memory — the original lists should not be changed.
Example:

Input:
First linked list: 1->2->3->4->6
Second linked list be 2->4->6->8,
Output: 2->4->6.
The elements 2, 4, 6 are common in
both the list so they appear in the
intersection list.

Input:
First linked list: 1->2->3->4->5
Second linked list be 2->3->4,
Output: 2->3->4
The elements 2, 3, 4 are common in
both the list so they appear in the
intersection list.

##

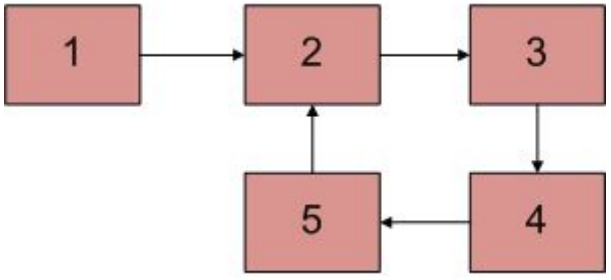
Question:

Given a singly linked list and a key, count the number of occurrences of given key in linked list. For example, if the given linked list is 1->2->1->2->1->3->1 and the given key is 1, then output should be 4.

##

Question:

Given a linked list, check if the linked list has a loop or not. Below diagram shows a linked list with a loop.



##

Question:

Given a singly linked list of characters, write a function that returns true if the given list is a palindrome, else false.

##

Question:

Write a function which takes a list unsorted in non-decreasing order and deletes any duplicate nodes from the list. The list should only be traversed once.

For example if the linked list is 11->11->11->21->43->43->60 then removeDuplicates() should convert the list to 11->21->43->60.

##

Question :

Given a linked list and two keys in it, swap nodes for two given keys. Nodes should be swapped by changing links. Swapping data of nodes may be expensive in many situations when data contains many fields.

It may be assumed that all keys in the linked list are distinct.

Examples:

Input: 10->15->12->13->20->14, x = 12, y = 20
Output: 10->15->20->13->12->14

Input: 10->15->12->13->20->14, x = 10, y = 20
Output: 20->15->12->13->10->14

Input: 10->15->12->13->20->14, x = 12, y = 13
Output: 10->15->13->12->20->14

##

Question:

Given a singly linked list, write a function to swap elements pairwise.

Input : 1->2->3->4->5->6->NULL

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

Input : 1->2->3->4->5->NULL

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

Input : 1->NULL

Output : 1->NULL

For example, if the linked list is 1->2->3->4->5 then the function should change it to 2->1->4->3->5, and if the linked list is then the function should change it to .

##

Question:

Write a GetNth() function that takes a linked list and an integer index and returns the data value stored in the node at that index position.

Example:

Input: 1->10->30->14, index = 2
Output: 30

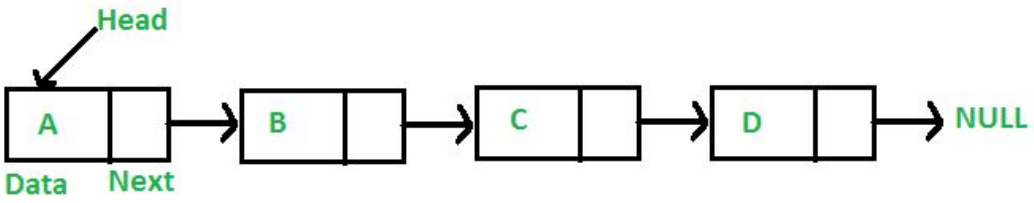
The node at index 2 is 30

##

Question:

Given a Linked List and a number n, write a function that returns the value at the n'th node from the end of the Linked List.

For example, if the input is below list and n = 3, then output is "B"



##

Question:

Given a string S. The task is to find the first repeated character in it. We need to find the character that occurs more than once and whose index of second occurrence is smallest. S contains only lowercase letters.

Input:

The first line of input contains an integer T denoting the no of test cases. Then T test cases follow. Each test case contains a string S.

Output:

For each test case in a new line print the first repeating character in the string. If no such character exist print -1.

Constraints:

1 <= T <= 100

1 <= |S| <=104

Example:

Input:

2

geeksforgeeks

hellogeeks

Output:

e

l

##

Question:

Given a Linked List of size N, where every node represents a sub-linked-list and contains two pointers:

(i) a next pointer to the next node,

(ii) a bottom pointer to a linked list where this node is head.

Each of the sub-linked-lists is in sorted order.

Flatten the Link List such that all the nodes appear in a single level while maintaining the sorted order.

Note: The flattened list will be printed using the bottom pointer instead of next pointer.

Example 1:

Input:
5 -> 10 -> 19 -> 28
| | | |
7 20 22 35
| | |
8 50 40
| |
30 45
Output: 5-> 7-> 8- > 10 -> 19-> 20->
22-> 28-> 30-> 35-> 40-> 45-> 50.

Explanation:
The resultant linked lists has every
node in a single level.
(Note: | represents the bottom pointer.)

Example 2:

Input:
5 -> 10 -> 19 -> 28
| |
7 22
| |
8 50
|
30
Output: 5->7->8->10->19->20->22->30->50
Explanation:
The resultant linked lists has every
node in a single level.

(Note: | represents the bottom pointer.)

Your Task:

You do not need to read input or print anything. Complete the function flatten() that takes the head of the linked list as input parameter and returns the head of flattened link list.

Expected Time Complexity: O(N*M)

Expected Auxiliary Space: O(1)

Constraints:

0 <= N <= 50

1 <= Mi <= 20

1 <= Element of linked list <= 103

##

Question:

Given a string S, find length of the longest substring with all distinct characters. For example, for input "abca", the output is 3 as "abc" is the longest substring with all distinct characters.

Input:

The first line of input contains an integer T denoting the number of test cases.

The first line of each test case is String str.

Output:

Print length of smallest substring with maximum number of distinct characters.

Note: The output substring should have all distinct characters.

Constraints:

1 ≤ T ≤ 100

1 ≤ size of str ≤ 10000

Example:

Input:

2

abababcdefababcdab

geeksforgeeks

Output:

6

7