University of Engineering & Management, Kolkata

Course: B.Tech **Semester: 3rd**

Paper Name: Data Structure & Algorithm Laboratory

Paper Code: PCC - CS391

LAST DATE OF YOUR ASSIGNMENT SUBMISSION IS 13/08/2021

ASSIGMENT NO. - 5 **(Related to Single Linked List)**

1. Write a program to create a singly linked list.

This is an to practice traversing a linked list. Given a pointer to the head node of a linked list, print each node's data element, one per line. If the head pointer is null (indicating the list is empty), there is nothing to print.
Function Description

Complete the printLinkedList function in the editor below.

printLinkedList has the following parameter(s):

```
SinglyLinkedListNode head: a reference to the head of the list
```

Input Format
The first line of input contains n, the number of elements in the linked list. The next n lines contain one element each, the data values for each node.
Constraints
1<=1000 1<=list[i]<=1000
Output Format

NA

Sample Input 0

```
2
16
13
```

Sample Output 0

```
16
13
```

2. **Write a menu program for following operation of Singly Linked List toInsert a node at beginning.**

Given a pointer to the head of a linked list, insert a new node before the head. The next value in the new node should point to head and the data value should be replaced with a given value. Return a reference to the new head of the list. The head pointer given may be null meaning that the initial list is empty.

Function Description

Complete the function insertNodeAtHead in the editor below.

insertNodeAtHead has the following parameter(s):

```
SinglyLinkedListNodellist: a reference to the head of a list
data: the value to insert in the data field of the new node
```

Input Format

The first line contains an integer n , the number of elements to be inserted at the head of the list. The next n lines contain an integer each, the elements to be inserted, one per function call.

Constraints

1<=n<=1000 1<=list[i]<=1000

Output Format

NA

Sample Input 0

```
5
383
484
392
975
321
```

Sample Output 0

```
321
975
 392
484
383
```

3. **Write a menu program for following operation of Singly Linked List to Insert a node as per given position.**

Given the pointer to the head node of a linked list and an integer to insert at a certain position, create a new node with the given integer as its data ttribute, insert this node at the desired position and return the head node.A position of 0 indicates head, a position of 1 indicates one node away from the head and so on. The head pointer given may be null meaning that the initial list is empty.

Example head: refers to the first node in the list 1->2->3 data=4 position=2

Insert a node at position 2 with data=4. The new list is 1->2->4->3

Function Description Complete the function insertNodeAtPosition in the editor below. It must return a reference to the head node of your finished list.

insertNodeAtPosition has the following parameters:

```
head: a SinglyLinkedListNode pointer to the head of the list
data: an integer value to insert as data in your new node
position: an integer position to insert the new node, zero based
indexing
```

Returns

```
SinglyLinkedListNode pointer: a reference to the head of the
revised list
```

Input Format

Input Format

The first line contains an integer n, the number of elements in the linked list. Each of the next n lines contains an integer SinglyLinkedListNode[i].data. The next line contains an integer data, the data of the node that is to be inserted. The last

line contains an integer position .

Constraints

NA

Output Format

NA

Sample Input 0

3
16
13
7
1
2

Sample Output 0

16 13 1 7

4. **Write a menu program for following operation of Singly Linked List to Insert a node at the last of the linked list.**

   You are given the pointer to the head node of a linked list and an integer to add to the list. Create a new node with the given integer. Insert this node at the tail of the linked list and return the head node of the linked list formed after inserting this new node. The given head pointer may be null, meaning that the initial list is empty. Function Description

   Complete the insertNodeAtTail function in the editor below.

   insertNodeAtTail has the following parameters:

   SinglyLinkedListNode pointer head: a reference to the head of a list
   int data: the data value for the node to insert

   Returns

   SinglyLinkedListNode pointer: reference to the head of the modified linked list

   Input Format

   The first line contains an integer n, the number of elements in the linked list. The next n lines contain an integer each, the value that needs to be inserted at tail.

   Constraints

   NA

   Output Format

   NA

   Sample Input 0

4
141
302
164
530
474

Sample Output 0

141
302
164
530
474

5. **Write a program to delete a node at beginning of Singly Linked List.**
   Given a linked list, the task is to remove the first node of the linked list and update the head pointer of the linked list.

   Sample Input 0

   2 -> 4 -> 6 -> 8 -> 33 -> 67 -> NULL

   Sample Output 0

   4 -> 6 -> 8 -> 33 -> 67 -> NULL

6. **Write a program to delete a node as per given position of Singly Linked List.**

   Input Format

   The first line of input contains an integer n, the number of elements in the linked list. Each of the next n lines contains an integer, the node data values in order. The last line contains an integer, position , the position of the node to delete.

   Constraints

   1<=n<= 1000 1<=list[i]<=1000

   Sample Input 0

   8
   20
   6
   2

19
7
4
15
9
3
Sample Output 0

20 6 2 7 4 15 9

Explanation 0

The original list contain 20->6->19->2->7->4->15->9 . After deleting the node at position 3 , the list is 20->6->2->7->4->15->9

7. Write a program to delete last node of Singly Linked.

Here just declare a temporary pointer temp and assign it to head of the list. We also need to keep track of the second last node of the list. For this purpose, two pointers ptr and ptr1 will be used where ptr will point to the last node and ptr1 will point to the second last node of the list.

Sample Input

The list contains: 10 20 30 40

Sample Output

The list contains: 10 20 30

8. Write a Program to reverse a link list

Input Format

The first line contains an integer t , the number of test cases. Each test case has the following format: The first line contains an integer n , the number of elements in the linked list. Each of the next lines contains an n integer, the data values of the elements in the linked list.

Constraints

1<=t<=10 1<=n<=1000 1<=list[i]<=1000

Output Format

NA

Sample Input 0
1
5
1
2
3
4
5
Sample Output 0
5 4 3 2 1

9. **Write a program in which contain singly linked list whose elements are type integer. Now break these linked list into two linked list and one list contain all even node and another contain all odd node.**

Sample Input 0

Original Linked List

0 1 4 6 9 10 11

Sample Output 0

Modified Linked List

0 4 6 10 1 9 11