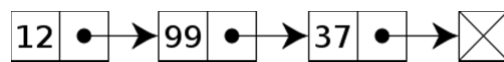


**Instructions:**

- Give sufficient comments in your program so that your program is understandable.
- You should save each program with the file name as mentioned in earlier labs.
- There will be partial evaluation even if your program does not run successfully for all the test cases as asked.
- You should upload all the programs (with .zip) before leaving the lab session. No credit of this lab will be given in case there is no Moodle submission.

1. A singly linked list whose nodes contain two fields: an integer value and a link to the next node. An example of such a list is shown in the figure below.



You have to store a list of numbers using a single linked list structure.

Define a structure definition say `struct Node` for a node in single linked list.

`struct Node *createList(int n):` To read `n` numbers from the keyboard and store them in a single linked list.

`void printList(struct Node *myList):` Print the elements in the list `myList`.

`struct Node *reverseList(struct Node *myList):` Reverse the elements in the list `myList` and then return the reversed list. Don't use any additional memory other than the input list.

**Test case#1:**

Input:

`L = {1, 9, 7, 5, 3}`

Output:

`L = {1, 9, 7, 5, 3}`

`L in reversed order = {3, 5, 7, 9, 1}`

**Test case#2:**

Input:

`L = {} // List does not contain any element`

Output:

`Error: Input list is empty`

`Reversing an empty list is ignored...`

[(2+5+5+8)]

2. You have given a linked list, say L. You have to check if the list is sorted either in ascending or descending order.

**Test case#1:**

Input:

L = {1, 3, 5, 7, 9}

Output:

The input list L is in ascending order.

**Test case#2:**

Input:

L = {1, 9, 7, 5, 3}

Output:

The input list L is not in sorted order.

**Test case#3:**

Input:

L = {9, 7, 5, 3, 1}

Output:

The input list L is in descending order.

Hint: You can reuse the functions defined in Q.1 in this lab.

[5+5+5]

3. You have given a linked list, say L. You have to check if the list contains any duplicate element or not. Also, find the min and maximum numbers in the input list.

**Test case#1:**

Input:

L = {1, 2, 3, 9, 7, 7, 6, 5, 4, 3, 2, 1}

Output:

The input list contains duplicate numbers of: 1, 2, 3, 5, 6, 7.

Minimum number is: 1

Maximum number is: 9

**Test case#2:**

Input:

L = {2, 4, 6, 8, 9, 7, 5, 3, 1}

Output:

The input list does not contain any duplicate number.

Minimum number is: 1

Maximum number is: 9

[5+5+5]

4. Define a structure `Customer` to specify data of customer in a bank. The data to be stored are: `Account number` (integer), `Name` (character string of exact length as appropriate to store a customer's name), and `Balance in account` (integer).

You have to read records from the keyboard and store them in a linked list and finally print the records in the list.

- Your program should read records till user wants to continue.
- Your program should insert a new record in the ascending order of the customers' account number.
- Once the record entry is over, print the record in the list in descending order of the balance of the customer.

[5+10+10]

5. Two lists are given, say `L1` and `L2`. You can use the function `struct Node *createLits (int n)` as defined in Q.1 to create the lists of size `n1` and `n2`, respectively.

- Find a new list `L` such that  $L = L1 \cup L2$ , that is, union of the lists `L1` and `L2`.
- Find a new list `L` such that  $L = L1 \cap L2$ , that is, intersection of the lists `L1` and `L2`.

**Test case#1:**

Input:

`L1 = {1, 3, 5, 7, 9}`

`L2 = {2, 3, 4, 5, 6, 7}`

Output:

`L (Union) = {1, 2, 3, 4, 5, 6, 7, 9}`

`L (Intersection) = {3, 5, 7}`

**Test case#2:**

Input:

`L1 = {1, 3, 5, 7, 9}`

`L2 = {0, 2, 4, 6, 8}`

Output:

`L (Union) = {0, 1, 2, 3, 4, 5, 6, 7, 8, 9}`

`L (Intersection) = {}`

**Test case#3:**

Input:

`L1 = {1, 2, 3, 4, 5}`

`L2 = {5, 4}`

Output:

`L (Union) = {1, 2, 3, 4, 5}`

`L (Intersection) = {4, 5}`

[5+10+10]

---\*---

**Submission instruction**

Zip your **Lab10** directory. Upload your zip file into the Moodle server latest by 4:30PM today.