Experiment no 4

Singly Linked List

OBJECTIVES:

- To learn about the concept of linked list.
- To learn different types of linked lists i.e Singly and Doubly linked list.
- To implement linked lists using class and functions.

1. LINKED LIST:

A singly linked list is a linear data structure in which the elements are not stored in contiguous memory locations and each element is connected only to its next element using a pointer.

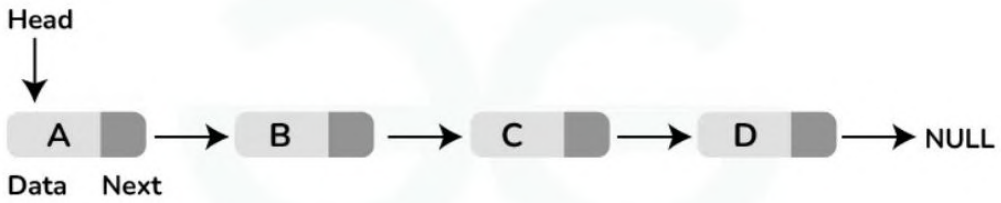


Figure 1: Linked List Block Diagram

2. SINGLY LINKED LIST:

A **singly linked list** is a fundamental data structure in computer science and programming. It is a collection of nodes where each node contains a **data field** and a **reference** (link) to the next node in the sequence. The last node in the list points to **null**, indicating the end of the list. This linear data structure allows for efficient insertion and deletion operations, making it a popular choice for various applications.

2.1 Node Definition:

We have created a Node class in which we have defined a __init__ function to initialize the node with the data passed as an argument and a reference with None because if we have only one node then there is nothing in its reference.

```
class Node:
    def __init__(self, data):
        self.data = data
        self.next = None
```

Figure 2: Node Class Creation

3. INSERTION IN LINKED LIST:

This method inserts the node at the beginning of the linked list. In this method, we create a new_node with the given data and check if the head is an empty node or not if the head is empty then we make the new_node as head and return else we insert the head at the next new_node and make the head equal to new_node.

```
def insertAtBegin(self, data):
    new_node = Node(data)
    if self.head is None:
        self.head = new_node
        return
    else:
        new_node.next = self.head
        self.head = new_node
```

Figure 3: Insertion at the beginning of list

4. INSERT A NODE AT A SPECIFIC POSITION:

This method inserts the node at the given index in the linked list. In this method, we create a new_node with given data, a current_node that equals to the head, and a counter 'position' initializes with 0. Now, if the index is equal to zero it means the node is to be inserted at begin so we called insertAtBegin() method else we run a while loop until the current_node is not equal to None or (position+1) is not equal to the index we have to at the one position back to insert at a given position to make the linking of nodes and in each iteration, we increment the position by 1 and make the current_node next of it. When the loop breaks and if current_node is not equal to None we insert new_node at after to the current_node. If current_node is equal to None it means that the index is not present in the list and we print "Index not present".

```
# Indexing starts from 0.
def insertAtIndex(self, data, index):
        new_node = Node(data)
        current_node = self.head
        position = 0
        if position == index:
            self.insertAtBegin(data)
        else:
            while(current_node != None and position+1 != index):
                position = position+1
                current_node = current_node.next

            if current_node != None:

                new_node.next = current_node.next
                current_node.next = new_node
            else:
                print("Index not present")
```

Figure 4: Inserting Node at a specific position

5. REMOVE LAST NODE FROM THE LINKED LIST:

In this method, we will delete the last node. First, we traverse to the second last node using the while loop, and then we make the next of that node **None** and last node will be removed.

```
def remove_last_node(self):

    if self.head is None:
        return

    current_node = self.head
    while(current_node.next.next):
        current_node = current_node.next

    current_node.next = None
```

Figure 5: Removing Last Node from the linked list

6. ACTIVITIES:

A1) Write a code for linked lists and also show how to Insertion in Linked List at End.
A2) Write a code for linked list and also show how to Update the Node of a Linked List.
A3) Write a code for linked list for 5 nodes and also loop through the linked list to show its next node address.
A4) Write a python program to get the n number of nodes from the end of a singly linked list.

IMPORTANT: All the activities` code should be attached to the manual before summary section.

Lab Exercise and Summary

Summary should cover Introduction, Procedure, Data Analysis and Evaluation.

Student's Signature:_____ **Date:**_____

LABORATORY SKILLS ASSESSMENT (Psychomotor)

Total Marks:100

Criteria (Max Marks)	Level 1 0% ≤ S < 50%	Level 2 50% ≤ S< 70%	Level 3 70% ≤ S< 90%	Level 4 90%≤ S ≤100%	Score (S)
Procedural Awareness (30)	Selects inappropriate skills and/or strategies Required by the task.	Selects and applies appropriate skills and/or strategies required by the task with major errors.	Selects and applies the appropriate strategies and/or skills specific to the task without significant errors.	Selects and applies appropriate strategies and/or skills specific to the task without any error.	
Practical Implementation (30)	Makes major critical errors in applying procedural knowledge related to python singly linked list.	Makes numerous critical errors in applying procedural knowledge related to python singly linked list.	Makes some non-critical errors in applying procedural knowledge related to python singly linked list.	Applies the procedural knowledge in optimized ways related to python Lists, singly linked list.	
Use of Tool/Equipment (30)	Uses tools, equipment and materials with limited competence.	Uses tools, equipment and materials with some competence.	Uses tools, equipment and materials with considerable competence.	Uses tools, equipment and materials with a high degree of competence.	
Safety (10)	Requires constant reminders to follow safety procedures.	Requires some reminders to follow safety procedures.	Follows safety procedures with only minimal reminders.	Routinely follows safety procedures.	
				Marks Obtained	

Instructor Name:_____ **Sign:** _____

LABORATORY SKILLS ASSESSMENT (Affective)

Total Marks: 40

Criteria (Max. Marks)	Level 1 0% ≤ S < 50%	Level 2 50% ≤ S < 70%	Level 3 70% ≤ S < 90%	Level 4 90% ≤ S ≤ 100%	Score
Introduction (5)	Very little background information provided or information is incorrect	Introduction is brief with some minor mistakes	Introduction is nearly complete, missing some minor points	Introduction complete and well-written; provides all necessary background principles for the experiment	
Procedure (5)	Many stages of the procedure are not entered on the lab report.	Many stages of the procedure are entered on the lab report.	The procedure could be more efficiently designed but most stages of the procedure are entered on the lab report.	The procedure is well designed and all stages of the procedure are entered on the lab report.	
Data Record (10)	Data is brief and missing significant pieces of information.	Data provides some significant information and has few critical mistakes.	Data is almost complete but has some minor mistakes.	Data is complete and relevant. Tables with units are provided. Graphs are labeled. All questions are answered correctly.	
Data Analysis (10)	Data is presented in very unclear manner.	Data is presented in ways that are not clear enough.	Data is presented in ways that can be understood and interpreted.	Data is presented in ways that best facilitate understanding and interpretation.	
Report Quality (10)	Report contains many errors.	Report is somewhat organized with some spelling or grammatical errors.	Report is well organized and cohesive but contains some grammatical errors.	Report is well organized and cohesive and contains no grammatical errors. Presentation seems polished.	
				Marks Obtained	

LABORATORY SKILLS ASSESSMENT (Cognitive)

Total Marks: 10

(If any) **Marks Obtained**	

Instructor's Signature:_____ **Date:**_____