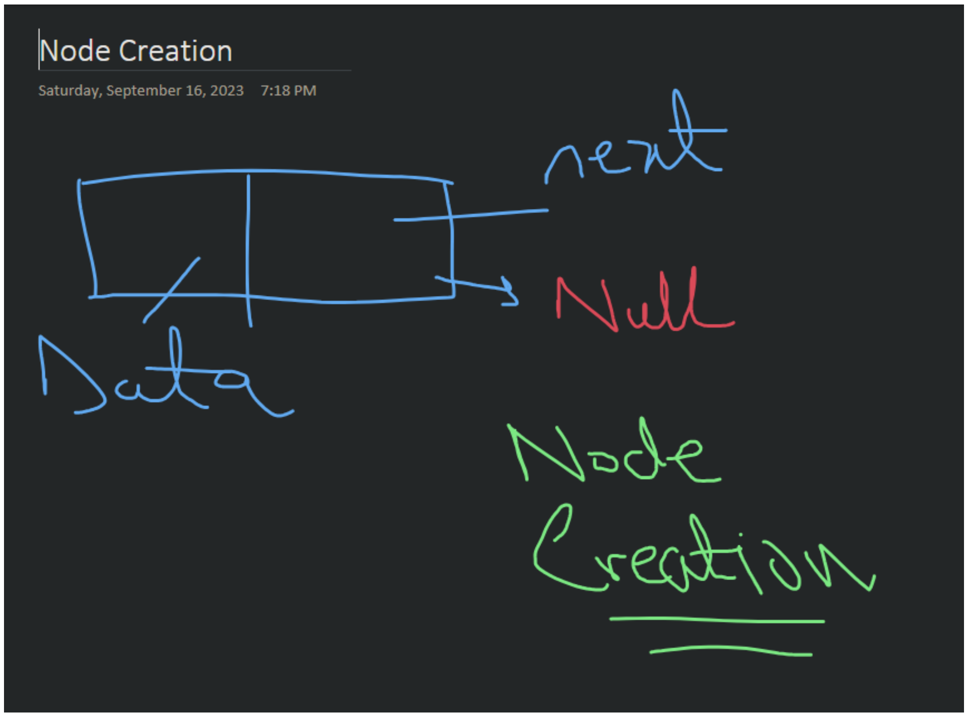
# ***<u>DS Diagram Notes:</u>***

## ***<u>Singly:</u>***

### ***<u>Node Creation:</u>***

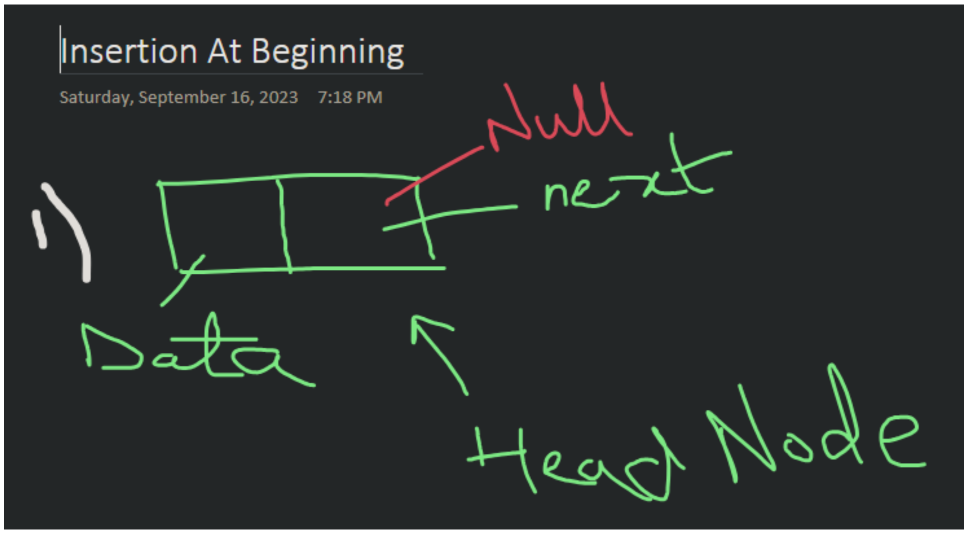
**-Node on its own has its next pointer point to nothing. There is no node connected to it.**

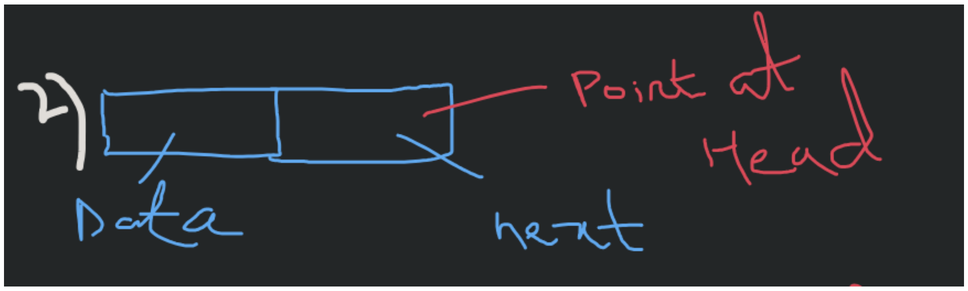


## ***Insertion at Beginning:***

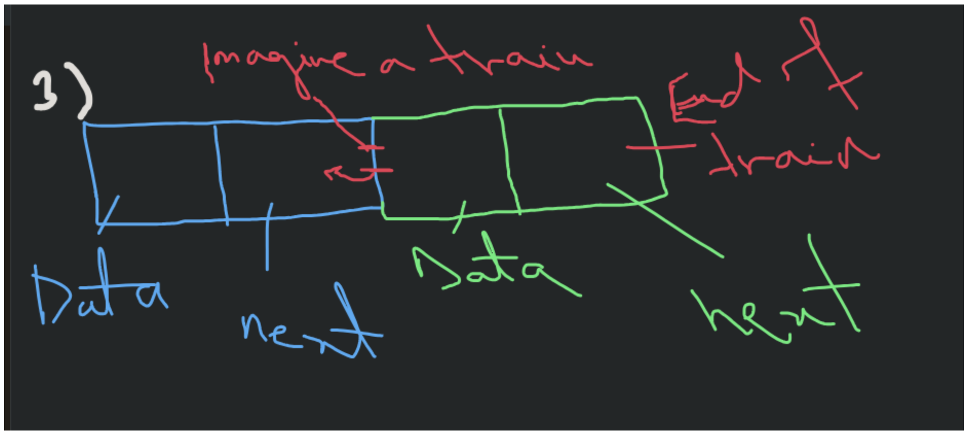
- **Node Creation.**

**-Node created which points at Head node.**



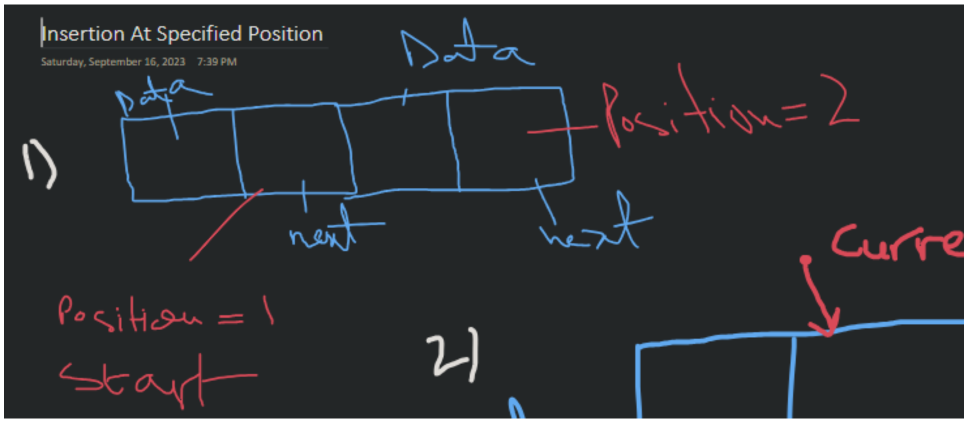


- **All put together.**

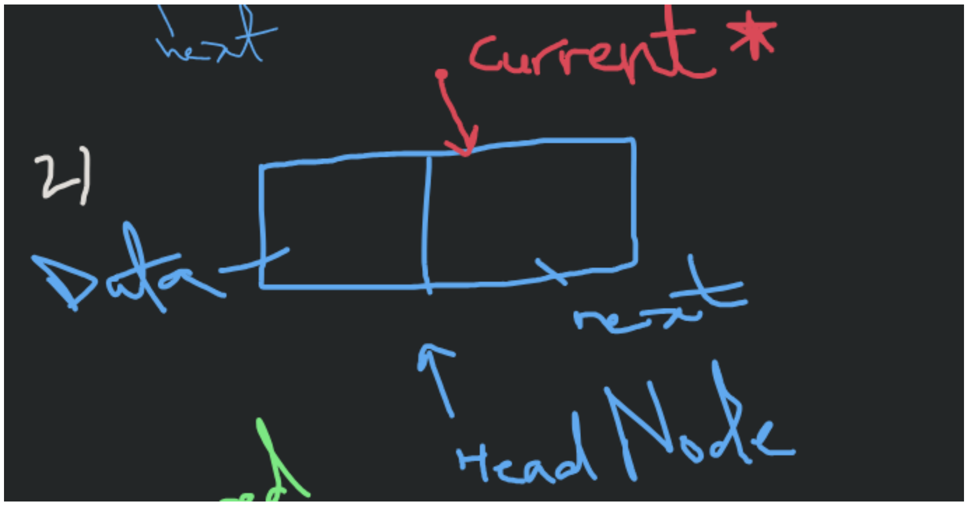


## ***Insertion At Specified Position:***

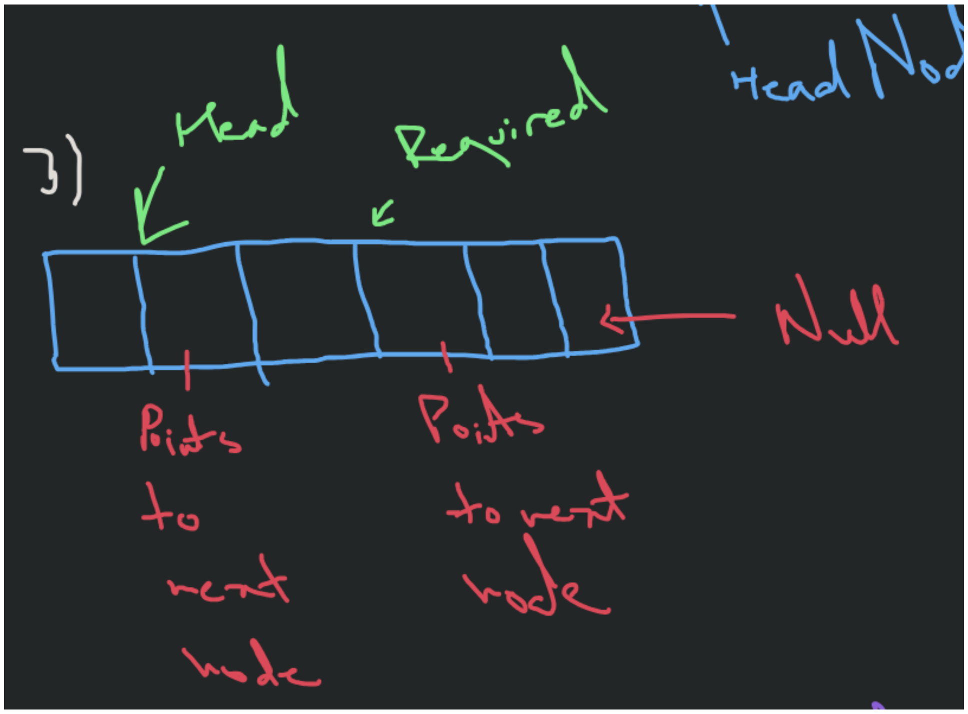
- **Head Node is at position 1 (Linked lists start from the first position.**


Insertion At Specified Position
Saturday, September 16, 2023 7:39 PM
Data
data
Position=2
1)
next
next
Curre
Position = 1
start
2)

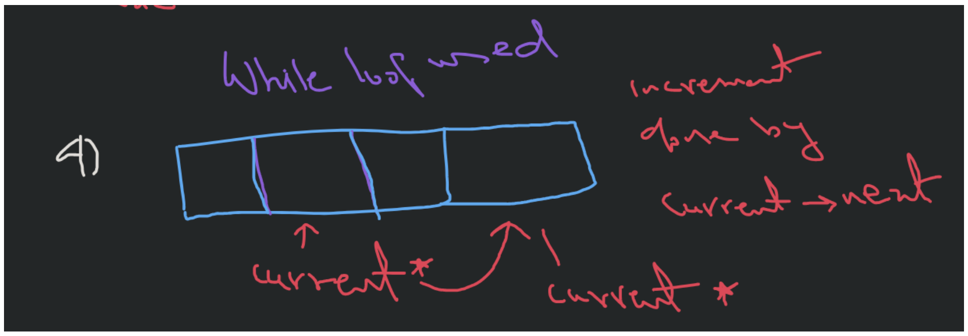
- **Current Node created which is at Head.**



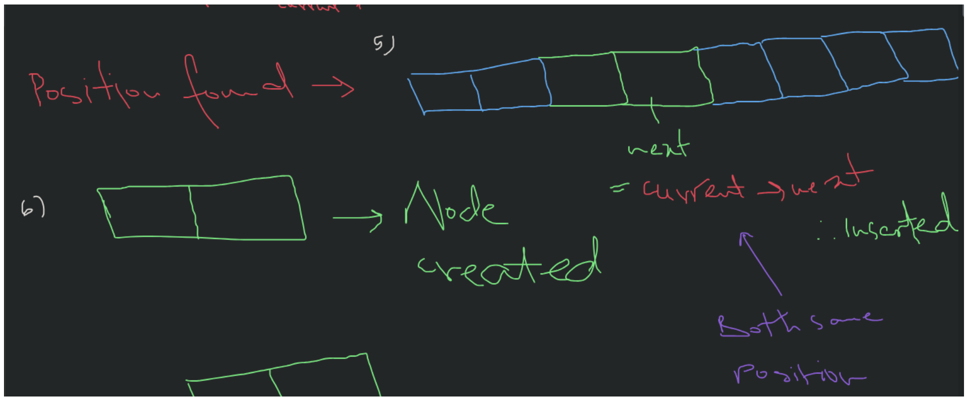
- **We must set the new node to Position 2.**



- **Traversing until the position.**



- **When position found, insert there.**

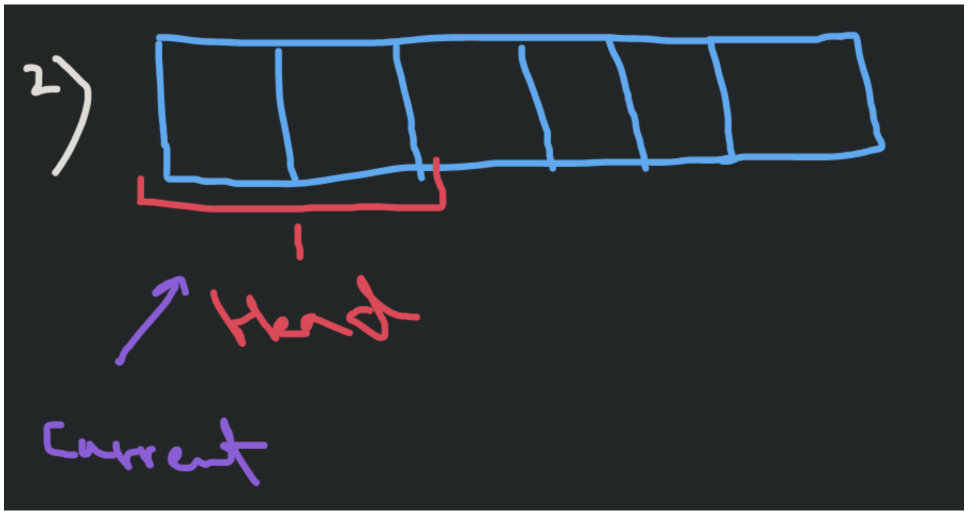


## ***Insertion At End:***

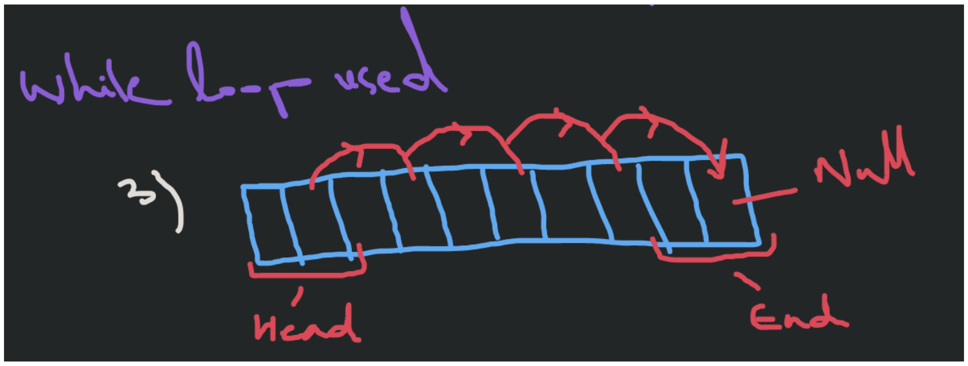
- **Create a node.**

1)
Node created

- **Make current node which is at head.**



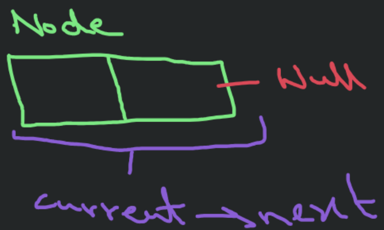
- **Current traversal until the end of the list (NULL)**



- **Insert at the position.**

4) ∴ current → next increments loop ⟹ Do until current → next = Null (Is at Tail)

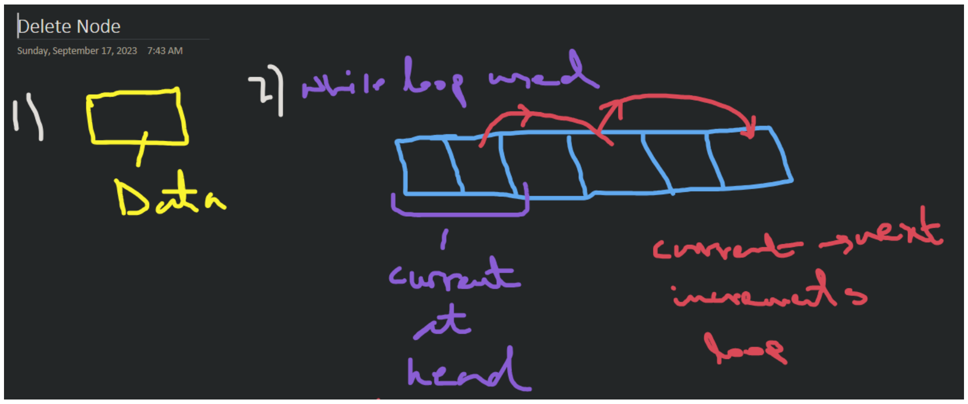
5) ∴



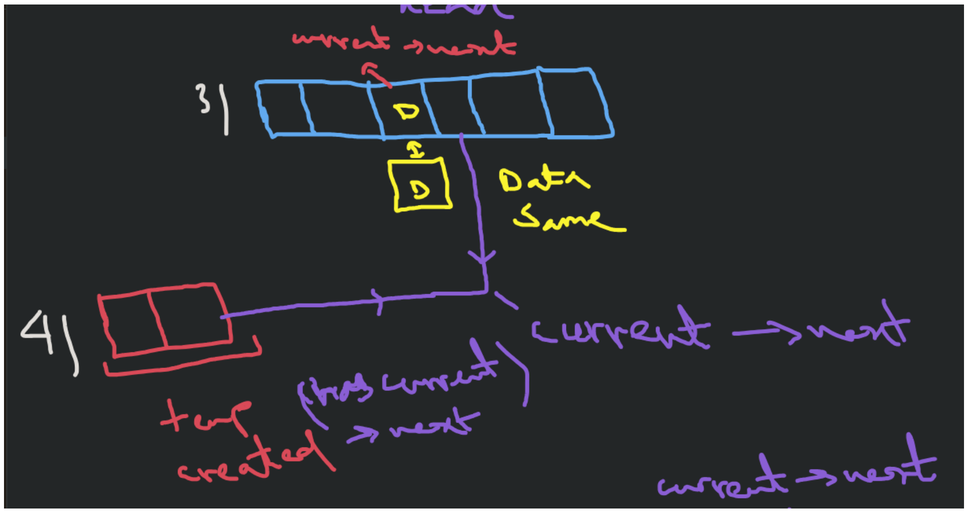
∴ Is now end Node

## ***Delete Node:***

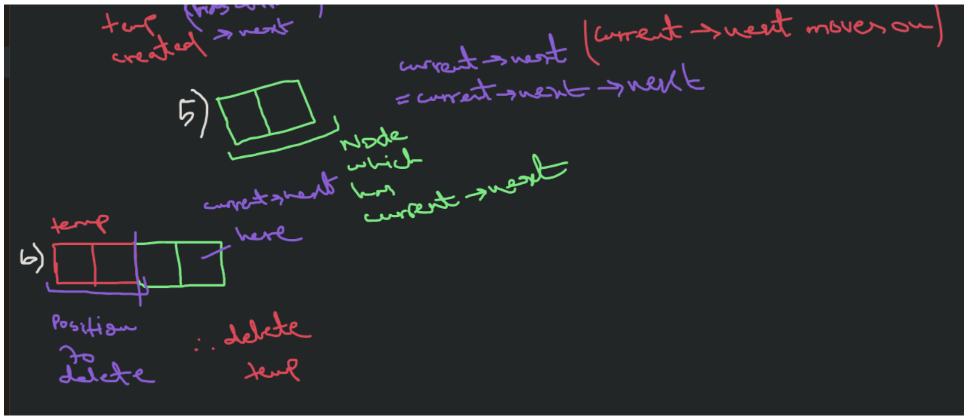
- ***Current traversal until end.***


Delete Node
Sunday, September 17, 2023 7:43 AM
1) Data
2) while loop used
current at head
current → next increments loop

- **Match Node's data with the data given.**
- **If found, Temp node created to go next to the Data holding node.**

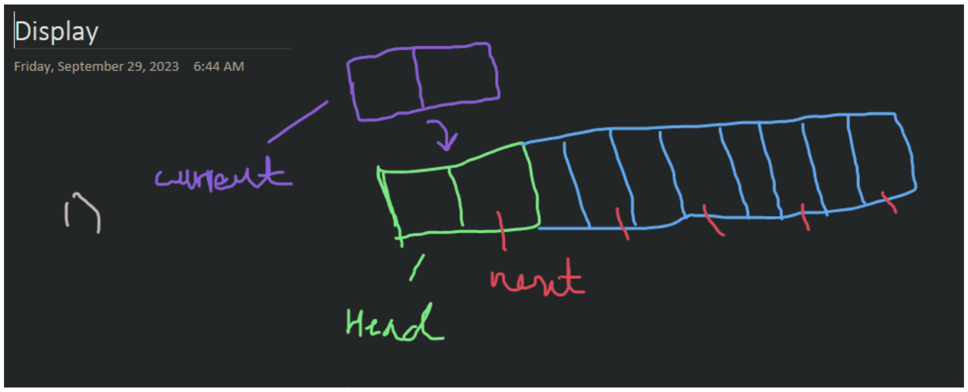


- **Current moves on and Temp holding the data is deleted.**

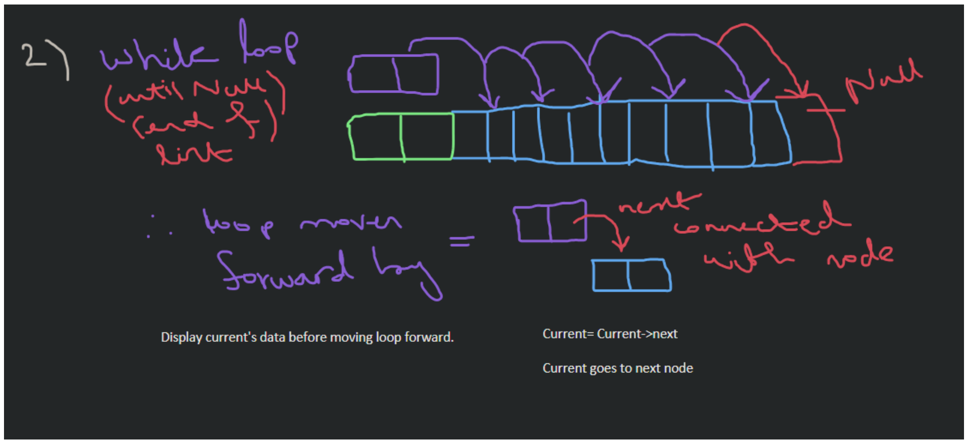


## ***Display List:***

- **Current node created pointing at the Head.**



- **Current traverses until the end using a While loop**
- **In the while loop, access the current node's data and print it.**



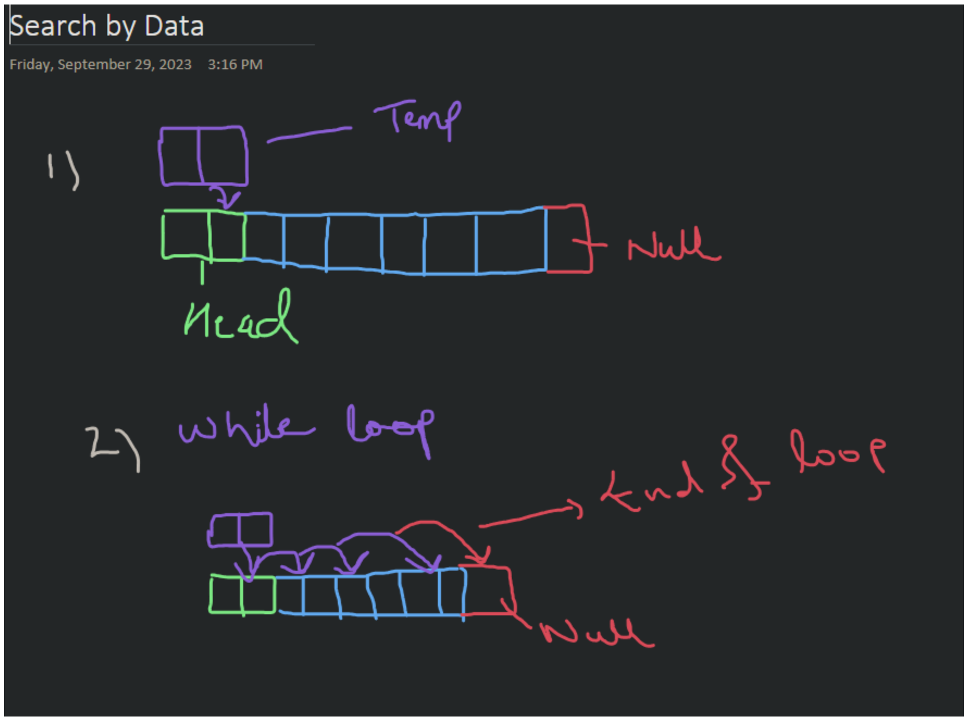
Display current's data before moving loop forward.

Current= Current->next

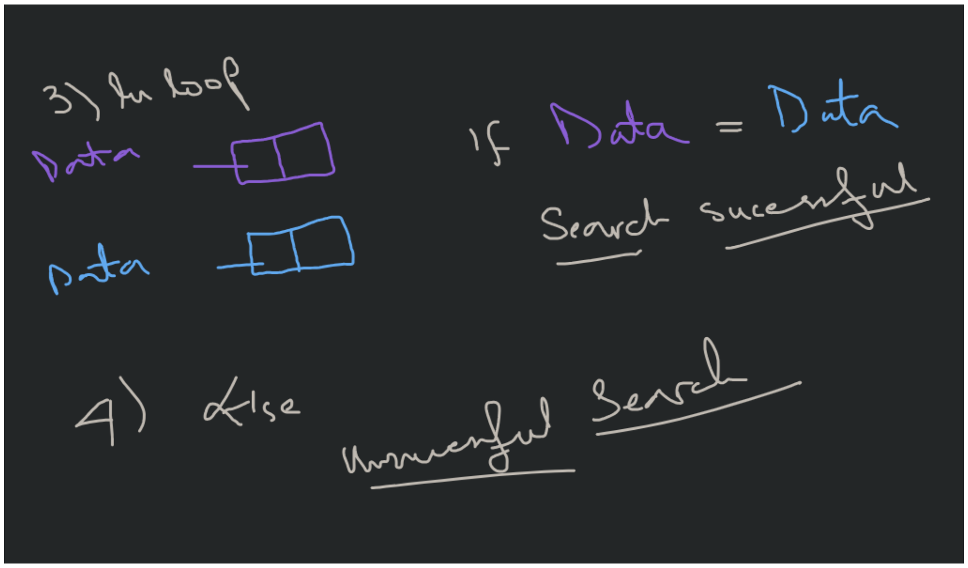
Current goes to next node

## ***Search by Data:***

- **Temp node created pointing at head which traverses until end of list.**

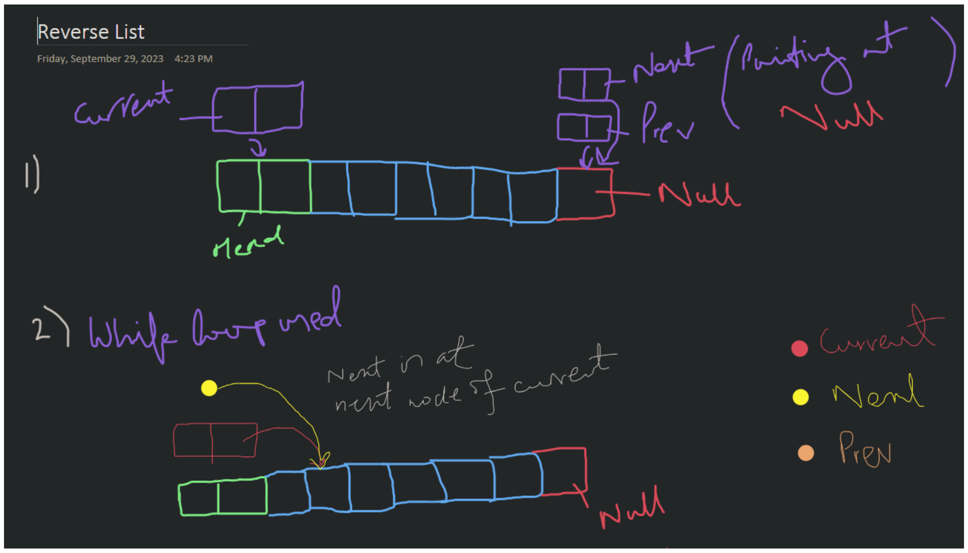


- In the loop, use if condition to search for data in node.
- If a node contains the data we want, the search is successful.
- Else, if the loop ends and the data was not found, the search was unsuccessful.

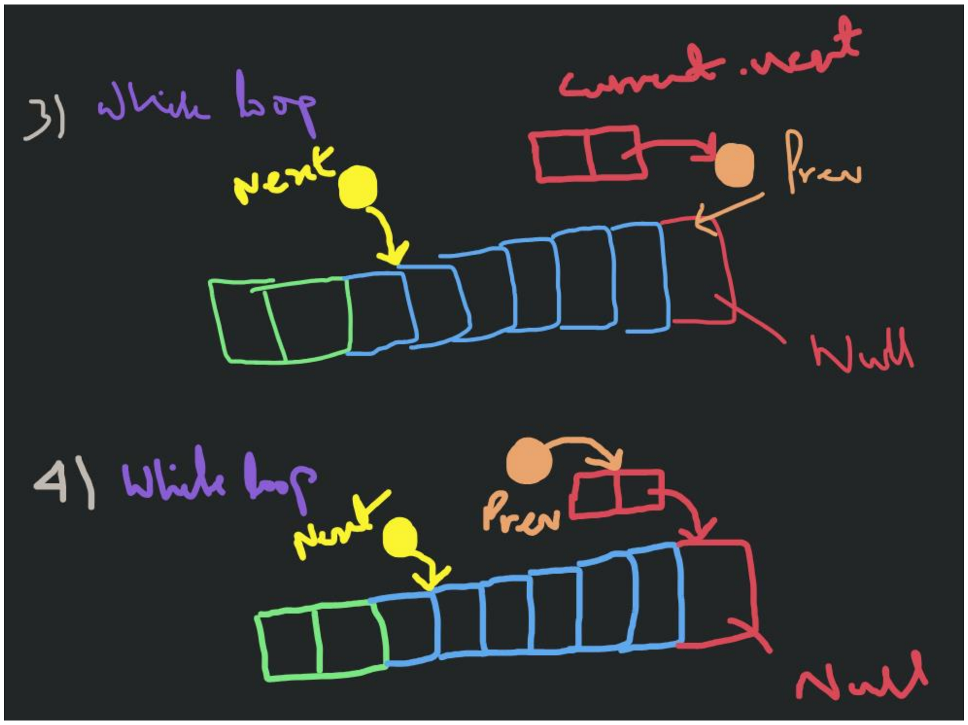


## ***Reverse List:***

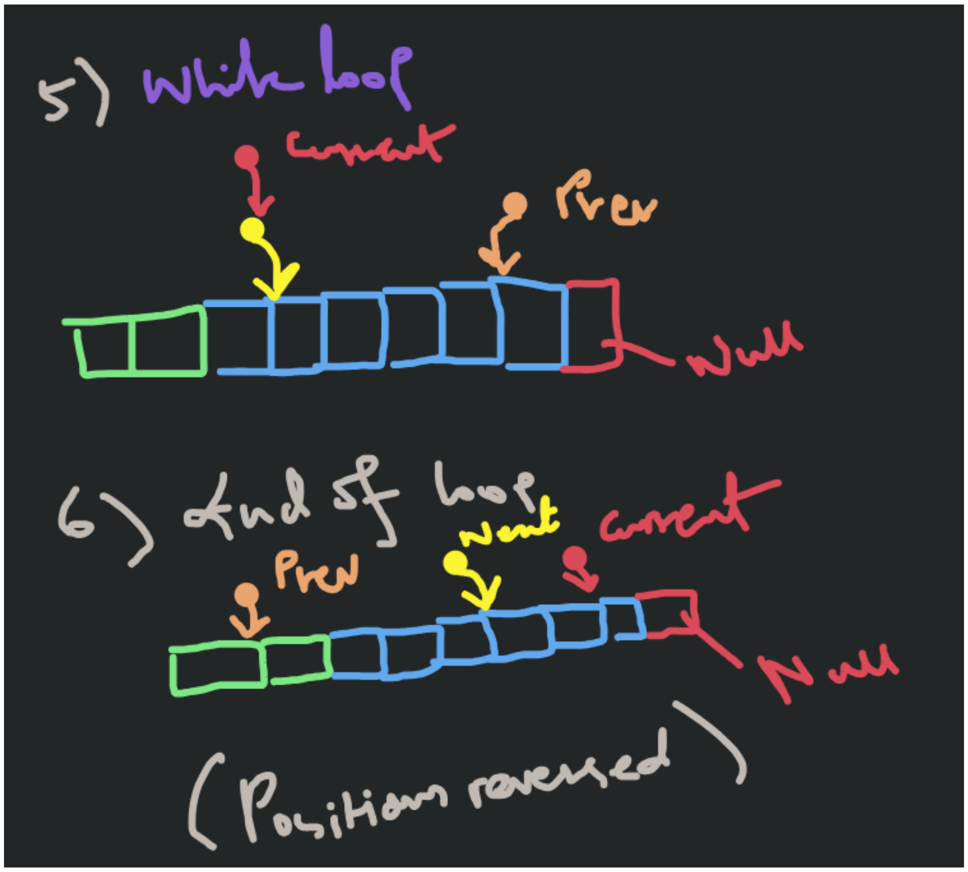
- **Current Node made at head.**
- **Next and Prev node made at end of list.**
- **Next is right ahead of Current Node.**



- **Current now just behind Prev, at last node.**
- **Prev points at Current.**



- Current points at Next
- So, Prev will move behind from the end of the list to the head of the list.
- Current will go from head of the list to the end of the list.
- Next will go from second node to second last node.



- All of the functions can be combined to use in each other.
- Recursion can be used in the functions to shorten the code.
- Sorting and other functions can be made by using the logics of these functions, like traversing and searching by data to be used in searching by position.

Muhammad Raza Khan

22K-4355