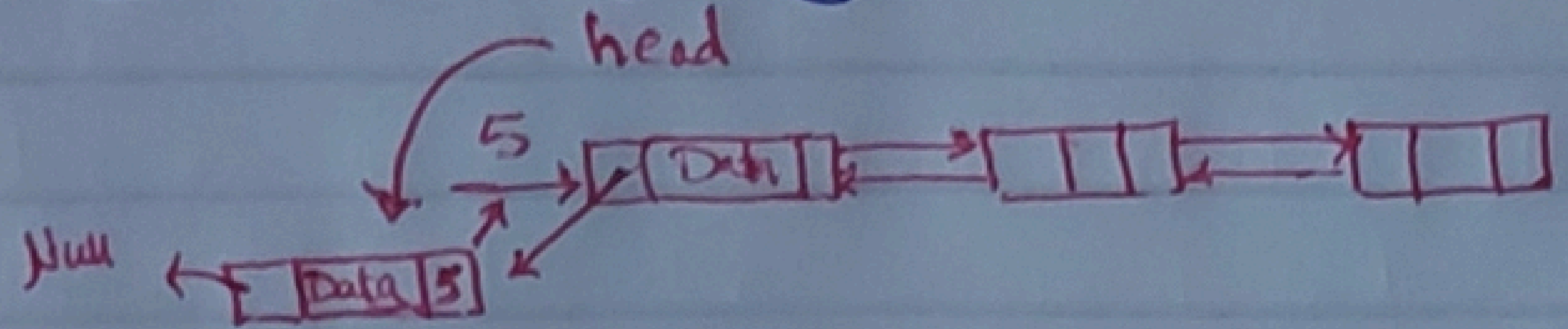
Double linked list assignment

Name: Muhammad Sami Ahmad , Sec: 2 , BN: 15

*) Insert from beginning:



make a newnode and make it point to "1st Node old" and make the previous of The "1st Node old" point to The newnode and make head on The newnode

*) Insert from Middle:

We will make a new pointer "Curr" and traverse the linked list till Curr points to The node which we will place the newnode after.

Newnode → next = Curr → next.

Newnode → prev = Curr.

Curr → next → prev = Newnode

Curr → next = Newnode.

*) Insert from End:

1) With tail: we have a pointer named tail on last node , So

Newnode → prev = tail , Newnode → next = null , Tail → next = newnode

Tail = Newnode.

2) Without tail: We will make a new pointer named Curr and traverse till last node and do Same operations as before.

Newnode → prev = Curr.

NewNode → Next = Null

Curr → next = newnode.

*) Insertion from Front and End (with tail) is The easiest, then Comes end without tail then middle.

middle > End no tail > Front, End (tail)

*) Deletion from Front:-

move the head to the next node, make the deleted node next points to Null , and delete it from memory.

head = head → next
head → prev = Null

*) Delete from middle:-

We will make a new pointer "Curr" traverse through list till the node we want to delete "Curr" Will be on it.

Curr → prev → next = Curr → next
Curr → next → prev = Curr prev , then delete "Curr" from memory.

This new pointer aim is to Connect list again after deletion of that node to not lose other data.

*) Delete from End:-

With tail: Tail = Tail → prev , Then delete that node from memory.
Tail → next = Null.

Without tail: We will make pointer "Curr" traverse till last node to be deleted then do Same processes.

Curr → prev → next = Null , delete Curr from memory.

*) Deletion from Front and End With tail (easiest)
Then Comes Deletion from Middle, End Without tail

Deletion Front , End with tail < Deletion Mid, without tail