

Node has 3 factor

\* Creating node class :

-Create class or struct.

-Enter node’s factors.

\*Creating Linked List:

- Write abilities.

- Define tail and head pointers.

\*Add operation

->At the beginning

1- Create a new node.

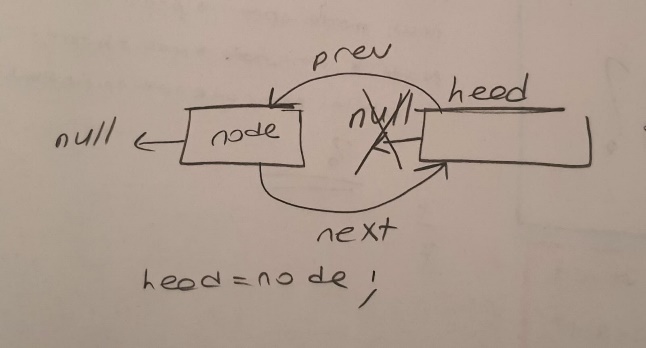
2- Add node’s value.

3-Put the NULL previous of node.

4- Look is list empty?

--- if yes : new node is HEAD and next of node is NULL.

--- if no : next of node is HEAD and head’s prev is node. So head is node.



->At the end

1-Create a new node.

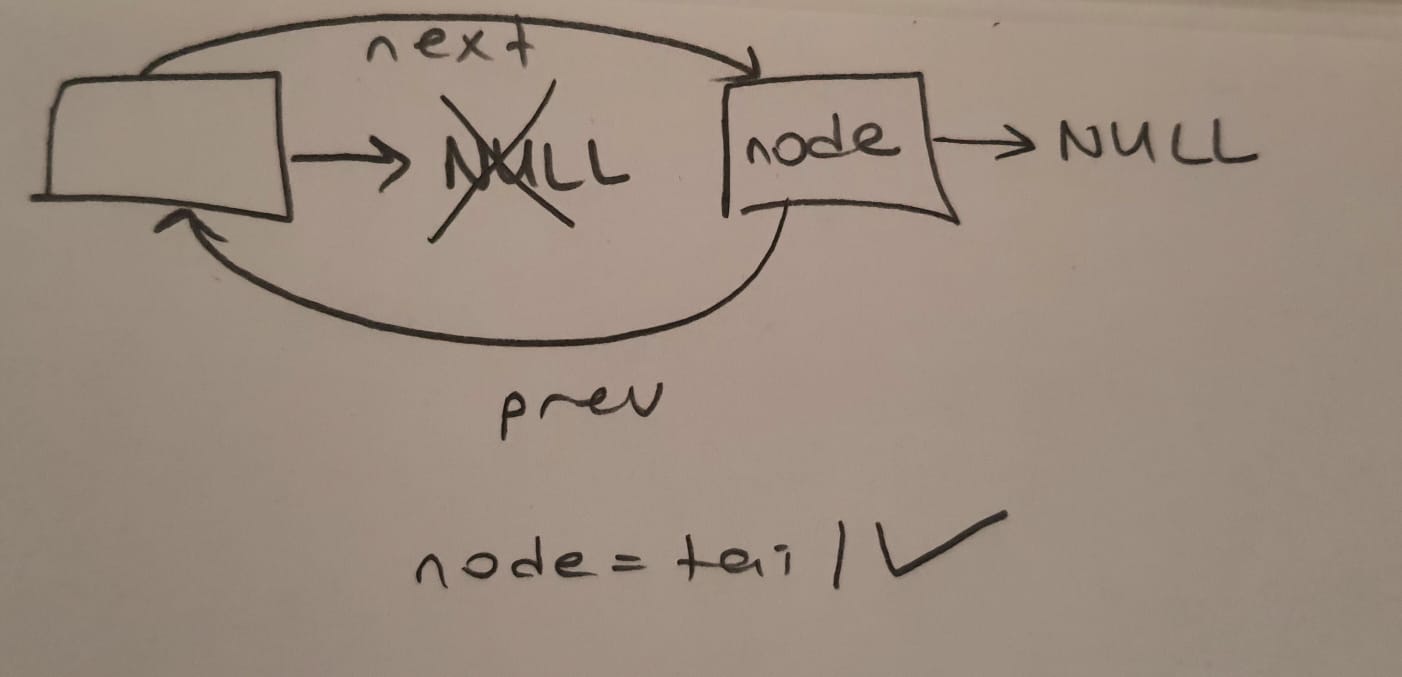
2-Add node’s value.

3-Put the NULL: next of node.

4-Look is list empty?

İf yes : node is head.

İf no : previous of node is tail, next of tail is node. So tail is node.



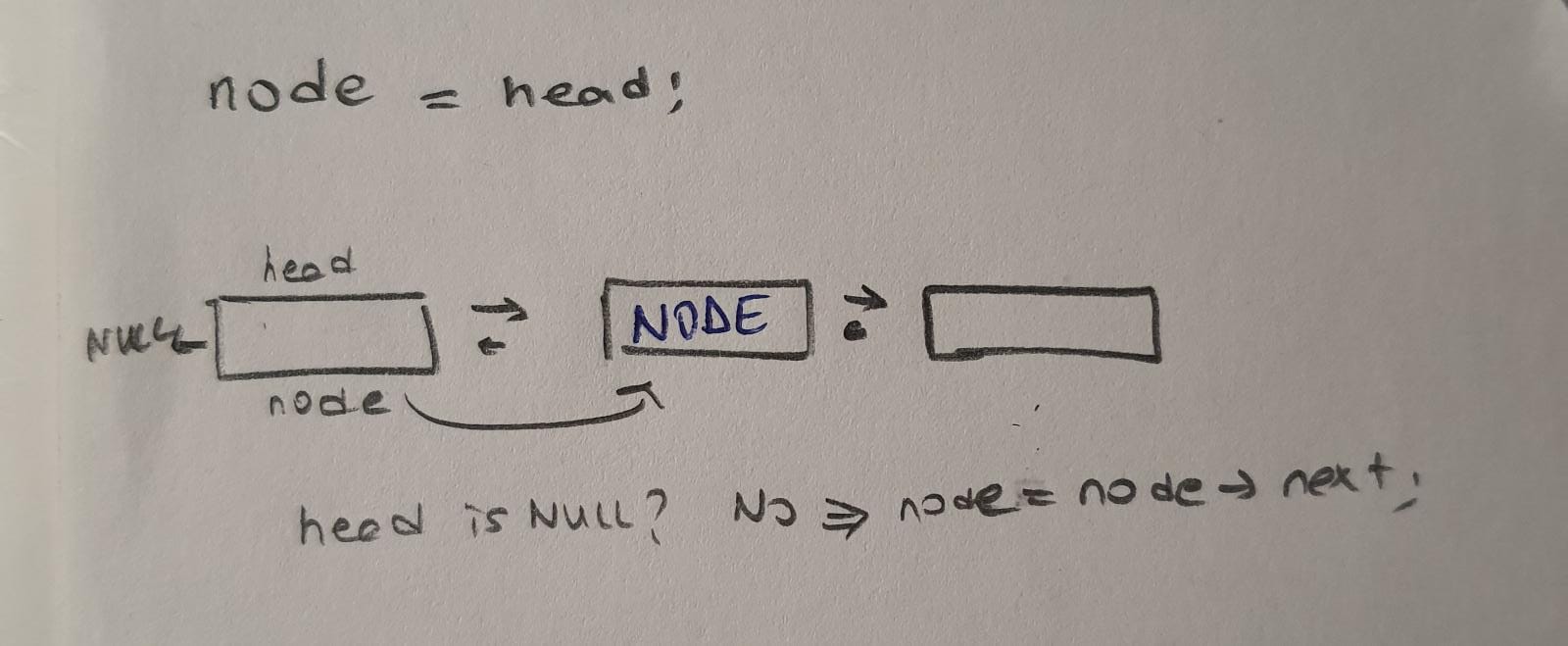
->Second position

1-Create a new node.

2-Check is list empty?

İf yes : node-> next : NULL , node -> prev = NULL, head is node.

İf no : node is = next of node: once a time.Because i want to put the data on second position.



\*Remove operations

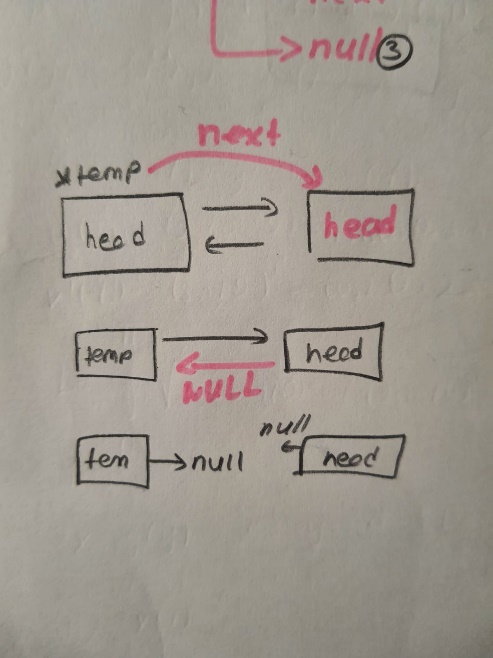
->Front

1- Check is list empty?

if yes: list is empty.

if no : check has list one data = head = NULL

if no : head is next of temp , prev of head is NULL ! and next of temp is NULL.



->End

1-Check is list empty?

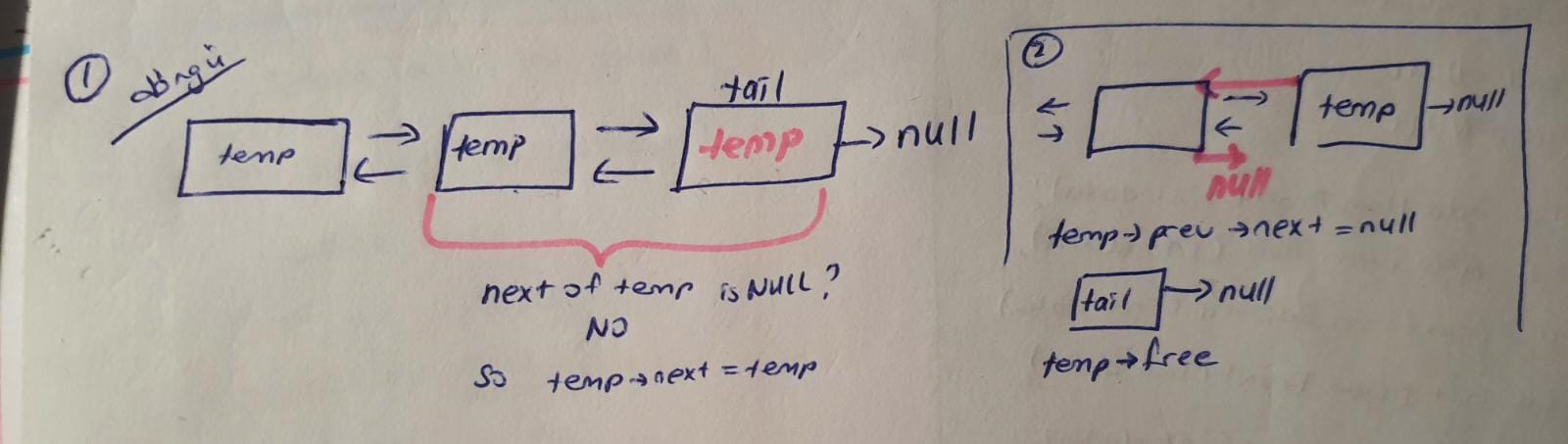
if yes : list is empty!

İf no: \*temp = head

check has one data?

İf yes: head is NULL

İf no : while next of temp isn’t NULL, temp is next of temp.And temp->prev-> next is NULL, so we delete last data.

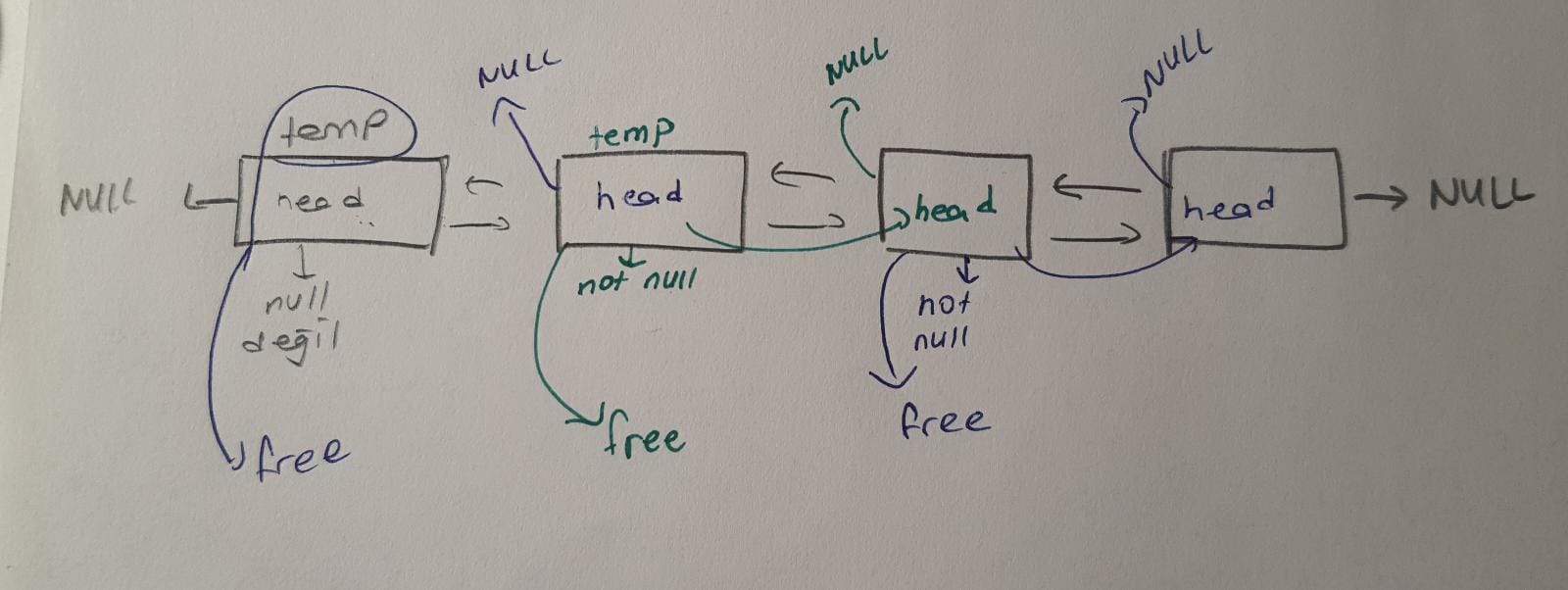


->All

1-Check list is it empty ?

2-Check has list one data?

3-If list has more than one data : while head is not NULL, head is next of head,prev of head is NULL.



\*Get

-front

1- \*temp = head

2- Check is list empty?

3- Check has list one node?

4- If list has more than one nodes : if temp-> prev == NULL print node.

-tail

1- \*temp = head

2- Check is list empty?

3- Check has list one node?

4- If list has more than one node :

while next of node is NOT NULL temp is next of temp

until next of temp is NULL

