(12) Array to DLL

- create head node
- traverse through the array from 1 to n creating new node for each element.
- & initially current is initialized to head node as & when we create new nodes we initialize current→next to newnode & new nodes prev as current & then move the current ahead.
& finally return the head.

__/__/____

(13) Deletion of head in DLL

Edge cases
- if list is empty
- if list has only 1element

- just move the head to next node by keeping a copy of the head
- now delete new heads prev
- copy of heads next should be deleted
- then delete copy of head.

(14) Deletion of tail in DLL

Edge cases.
- if list is empty
- if list has only 1 node.

- traverse till the last but one element then break out
& delete currents next
& set currents next to NULL

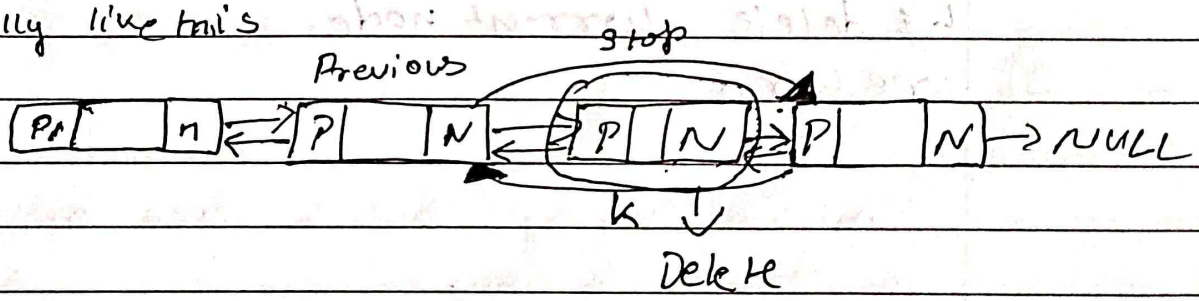
__/__/____

(15) Kth pos deletion in DLL

Egdge cases
- if K<=0 or list is empty
- if K=1 - you move the head to next node
  - also only if head != nullptr only then remove its prev

- traverse through the array till last eden node while keeping track of previous node & node count
- now if node count == K then
  - ↳ prev nodes next to current nodes next
  - ↳ connect current nodes next nodes prev to current nodes previous node
  - ↳ then delete current & return head.

basically like this:



Edge case
- if K = tail value
- then just connect previous nodes next to NULL
- Delete current & return head.

__/__/____

(16) Delete given node

Edge cases

- if given node is null just return null
- if given node is head
  - └ check if there is a node after head
    if yes then set that next nodes prev to null
  - └ delete the head
  - └ return.
- if given nodes is tail then
  - └ connect tails previous node to null
  - └ Delete node
  - └ return.
- if its a midlle node then
  - └ connect previous to currents front node
  - └ connect front nodes prev to previous
  - └ & delete current node.
  - return