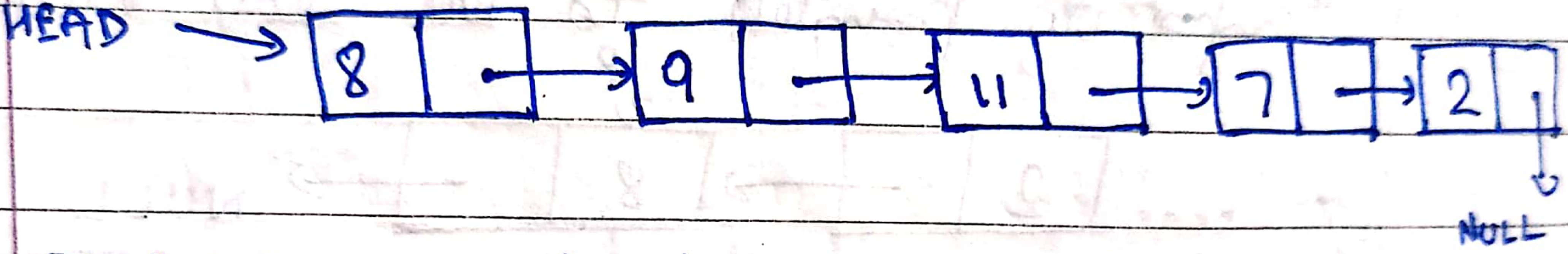
Deletion in a Linked list

→ Consider the following linked list:

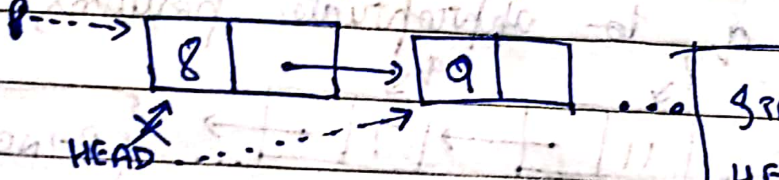


→ Deletion can be done for the following cases:

1) Deleting the first Node.
2) Deleting the node at an index.
3) Deleting the last node.
4) Deleting the first node with a given value.

→ The deletion is just like insertion. Both are done by rewiring the pointer connections, the only caveat being: we ~~used to~~ need to free the memory of the deleted node using free().

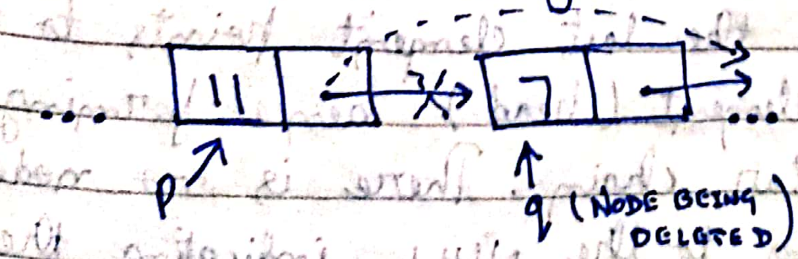
→ CASE 1: Deleting the first Node:



```
STRUCT NODE* P=HEAD
HEAD = HEAD→NEXT;
FREE(P);
```

→ CASE 2: Deleting the node at an index:

→ For deleting a given node, we first bring a temporary pointer p, before element to be deleted and q on the element being deleted.

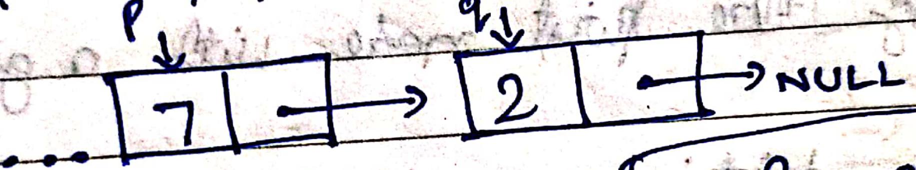


```
P→NEXT = q→NEXT
FREE(q);
```

→ CASE 3: Deleting the last Node:

Page No.
Date

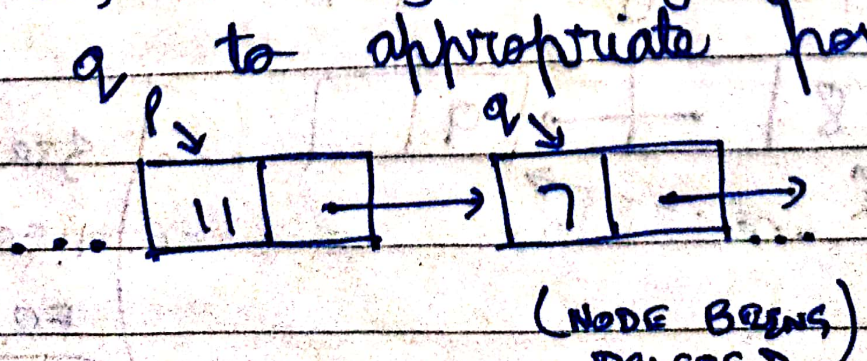
→ Last node can be deleted just like case 2 by bringing ~~up~~ p on second last element and q on last element.



```
P→NEXT = NULL
FREE(q)
```

→ CASE 4 : Delete the first node with a given value.

→ This can be done exactly like case 2 by bringing pointers p & q to appropriate positions.



```
P→NEXT = q→NEXT
FREE(q)
```