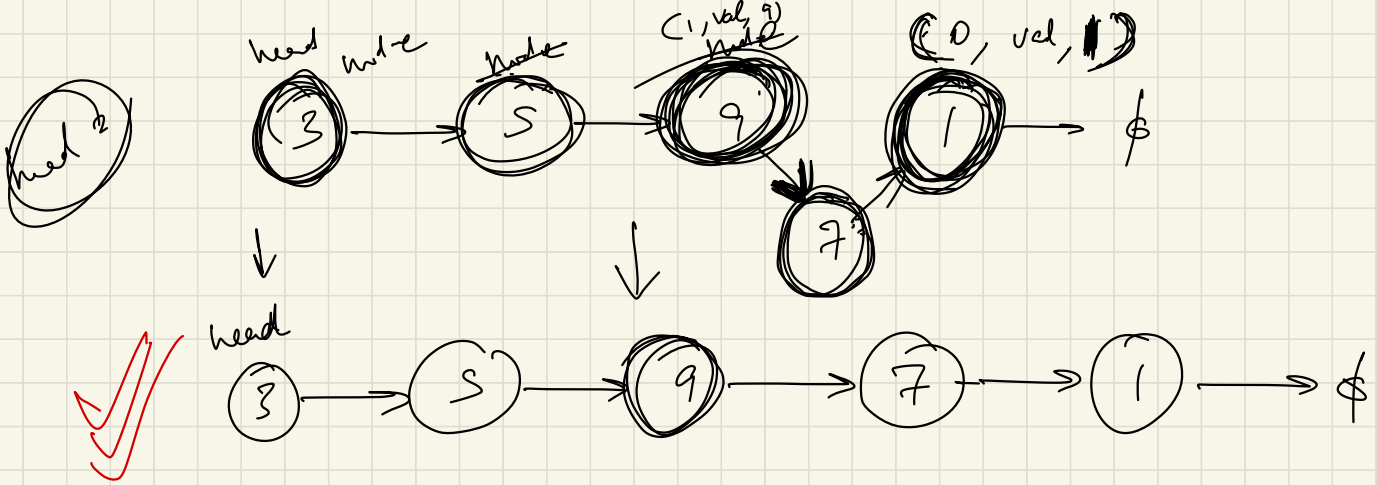
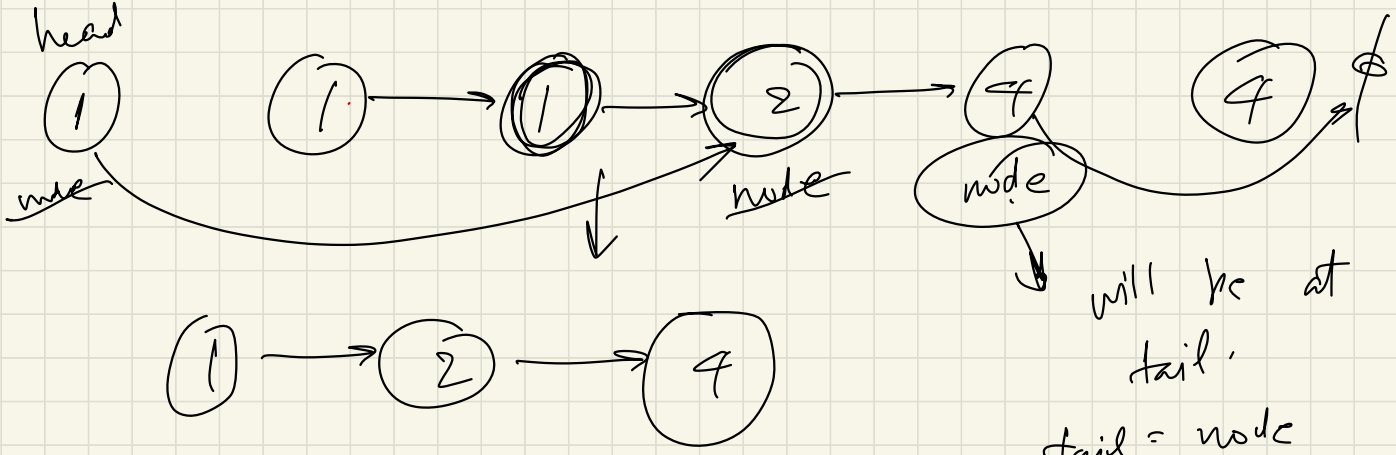


(1) void return type & make changes in ll

(2) node return type that returns the list node to change the structure.

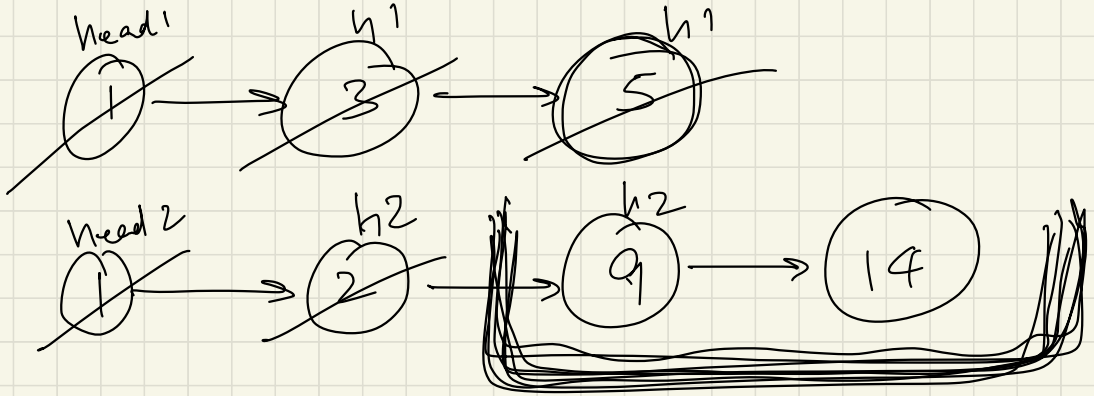
Q:



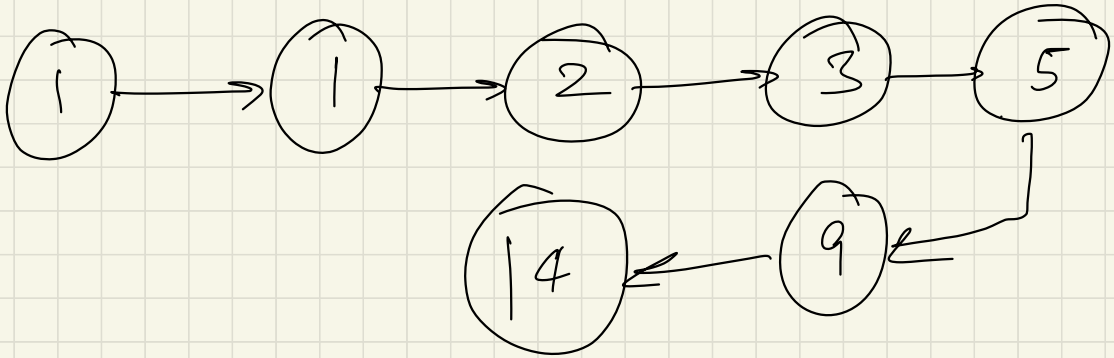
Q

list 1 :

list 2 :



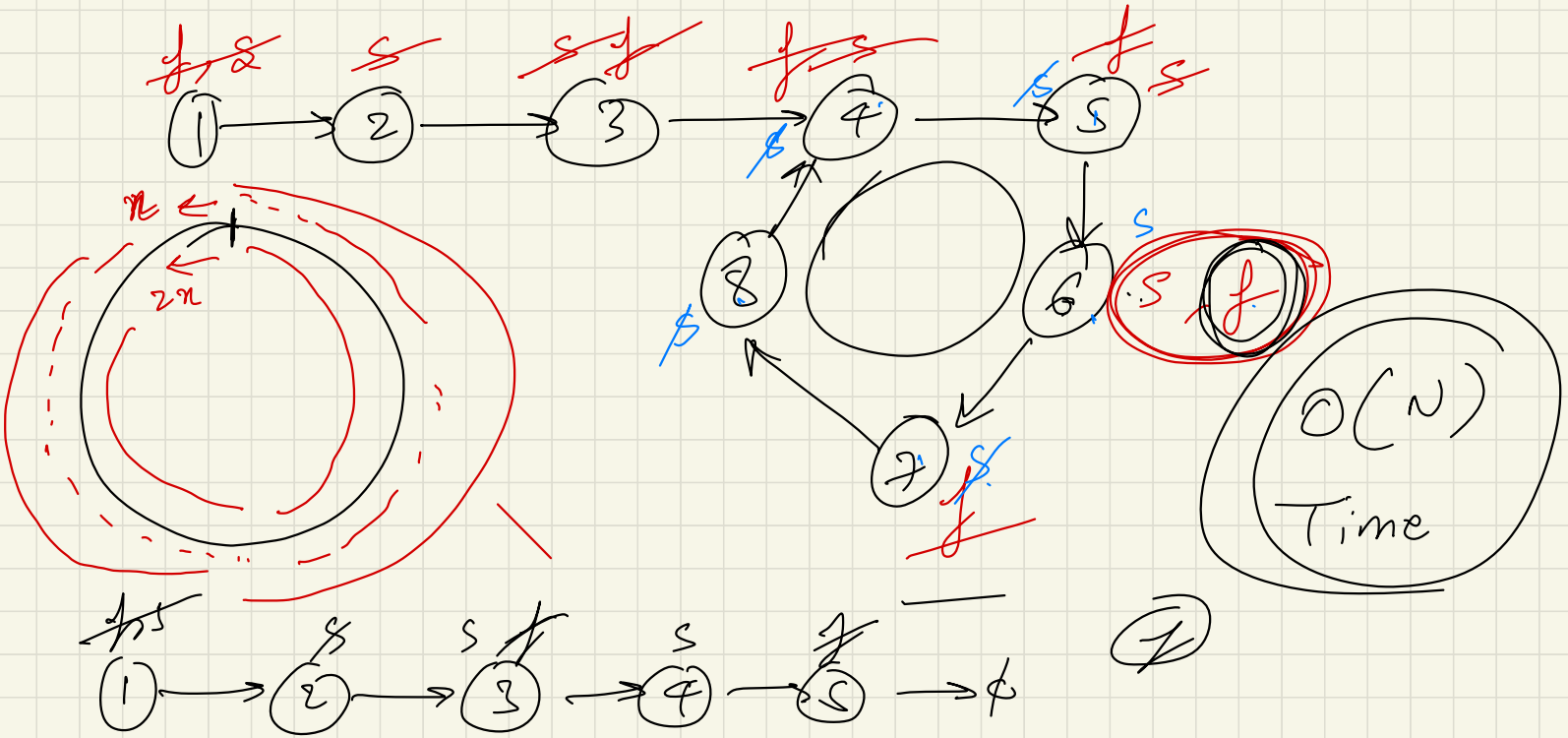
ans :



---

fast & slow pointer:

- Cycle detection
- Finding a node in cycle, etc.


n
2n
O(N)
Time

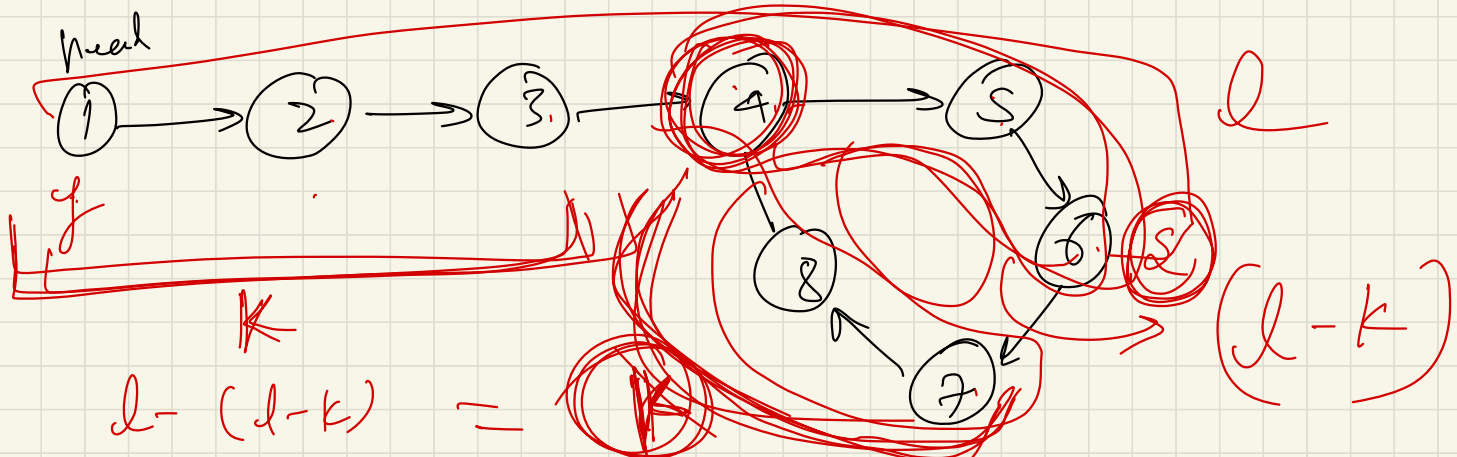
2 possibilities:

① $f \quad s \longrightarrow f - s$

② $f - s \longrightarrow f, s$ → in 2 steps total they will meet

---

Q



$l - (l-K) = K$

① Find length of cycle ✓

② move S ahead by length of cycle times.

③ move S & f one by one, it will meet at start.

Q: Google Question

Ex: 12 ⇒ $1^2 + 2^2 = 1 + 4 = 5$

$5^2 = 25$

$2^2 + 5^2 = 4 + 25 = 29$

$2^2 + 5^2 = 4 + 81 = 85$

8^2 + 9^2 = 89

$8^2 + 9^2 = 64 + 81 = 145$

$1^2 + 4^2 + 5^2 = 42$

$4^2 + 1^2 = 20$

$2^2 + 0 = 4$

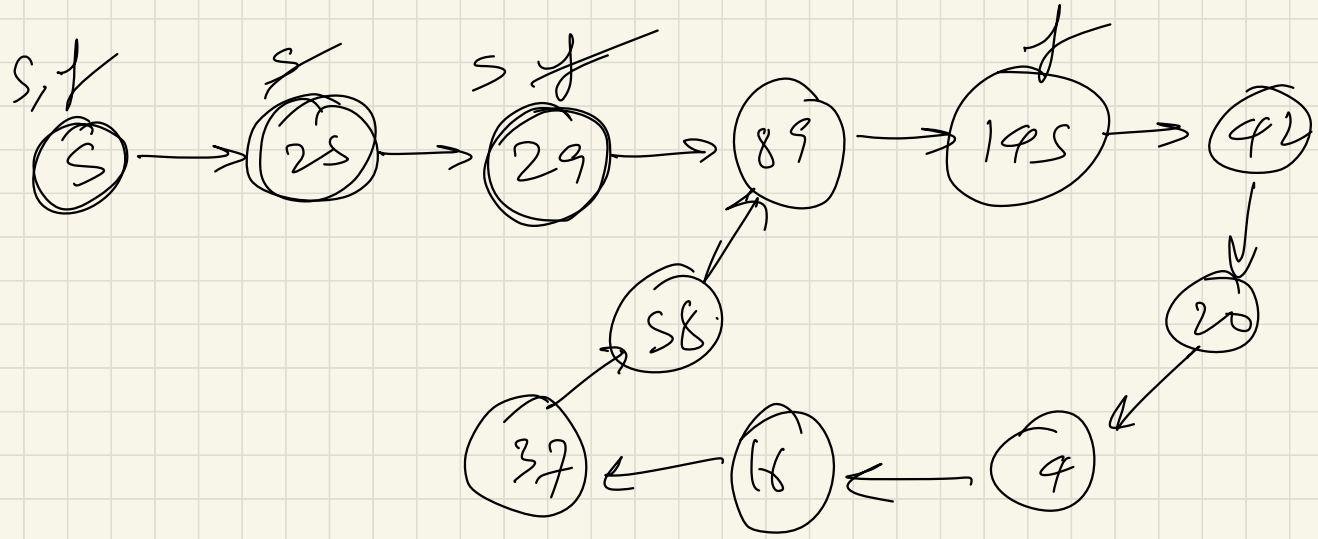
$4^2 = 16$

$1^2+6^2 = 37$

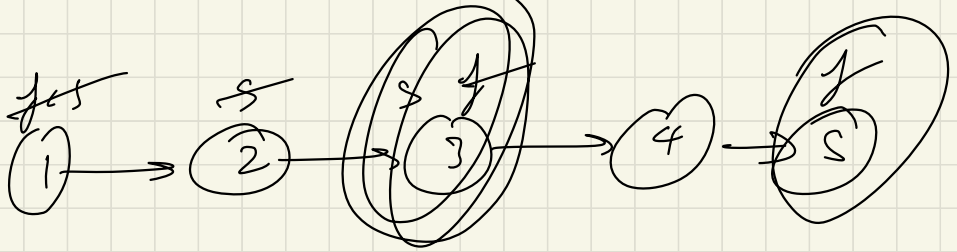
$3^2+7^2 = 9+49 = 58$

$5^2+8^2 = 25+64 = 89$

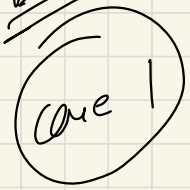
$8^2+9^2 = \ldots$

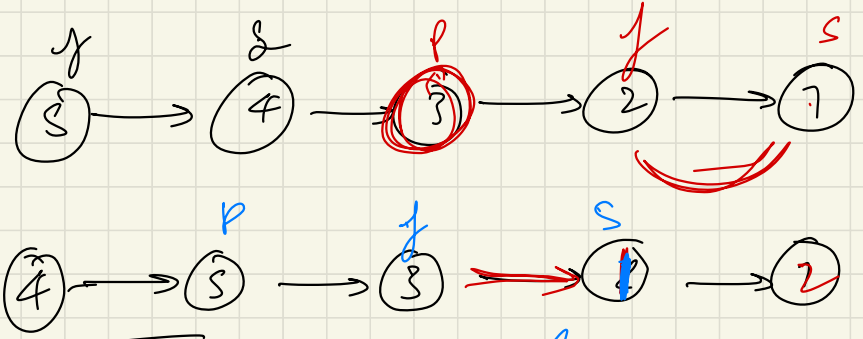


Q: middle



BS

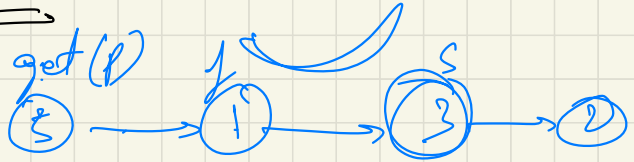




head = s

f.next = s.next

s.next = f



p.next = s

f.next = s.next

s.next = f

case 2

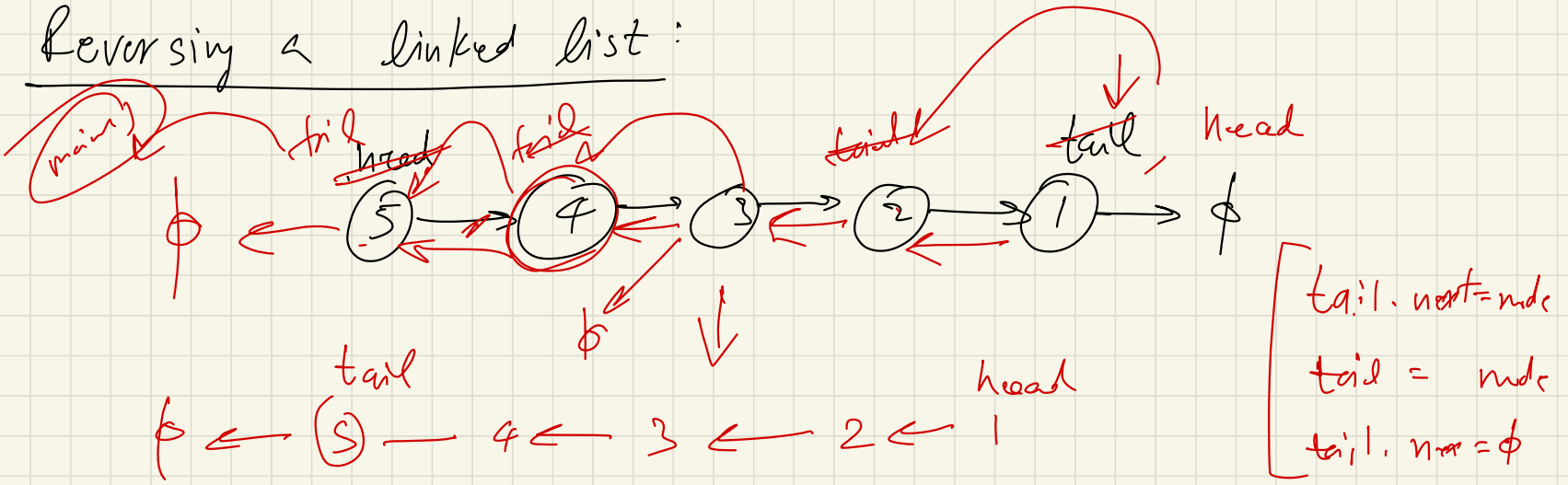
p.next = s

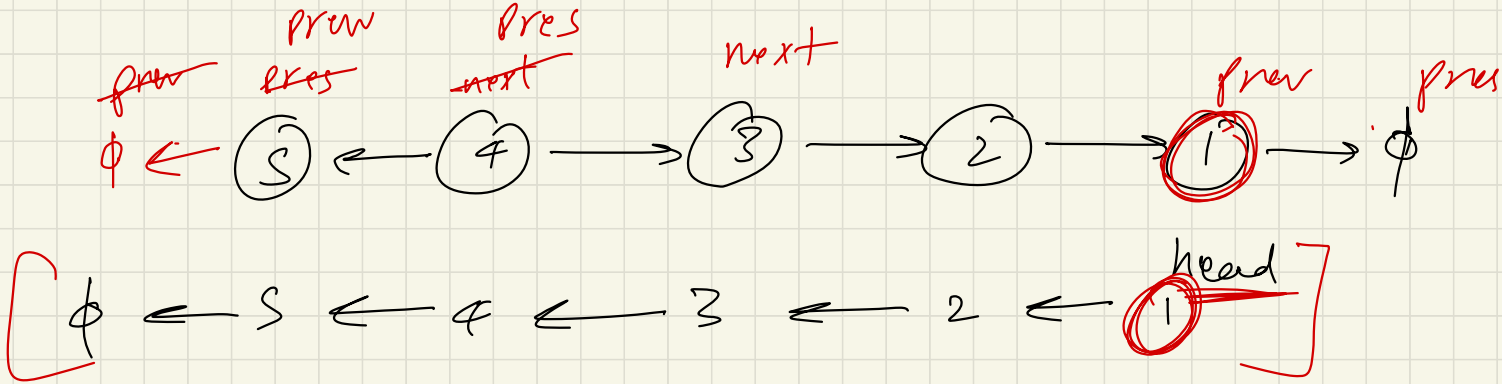
tail = f

f.next = ϕ

s.next = tail

Reversing a linked list:





while present != ϕ:

pres.next = prev

prev = pres

pres = next

next = next.next // note: ϕ pointer check

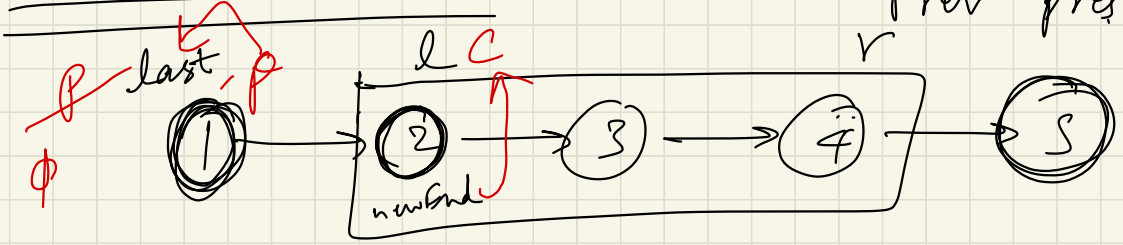
In the end:

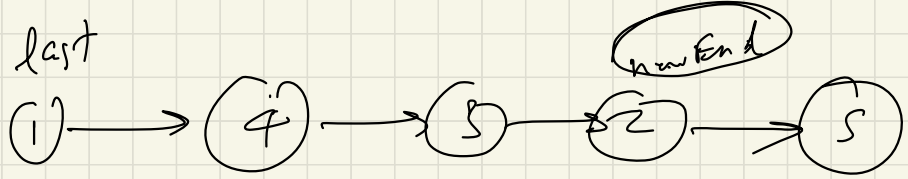
head = prev.

Q Reverse a LL - II

$l = 2$

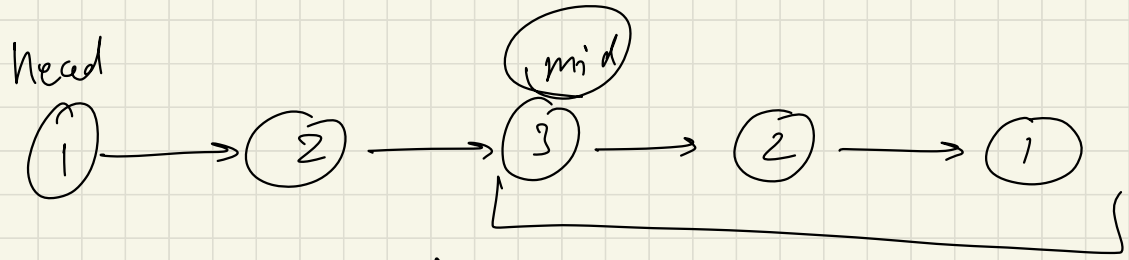
$r = 4$

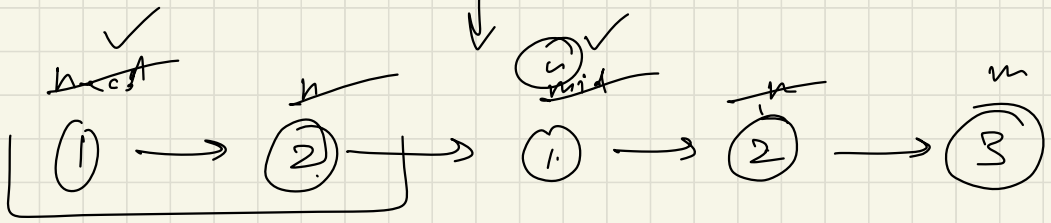


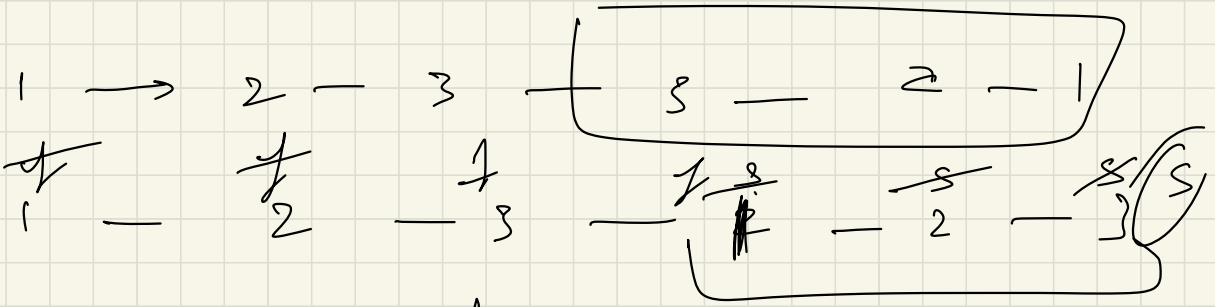


last.next = p / if last = p :
head = p

Q:

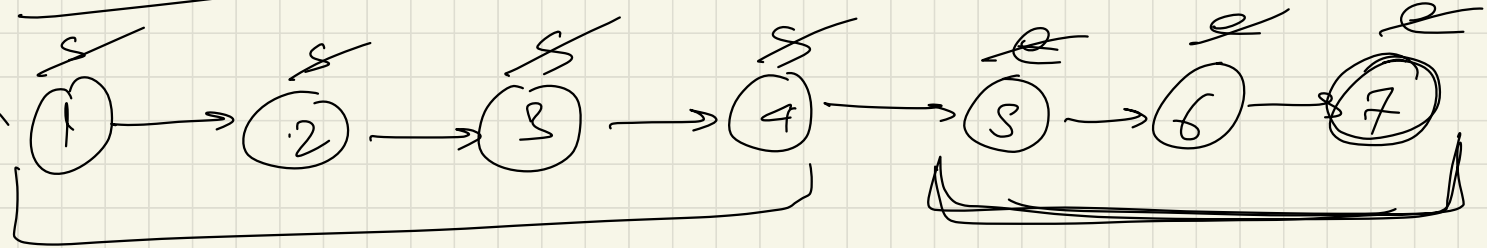


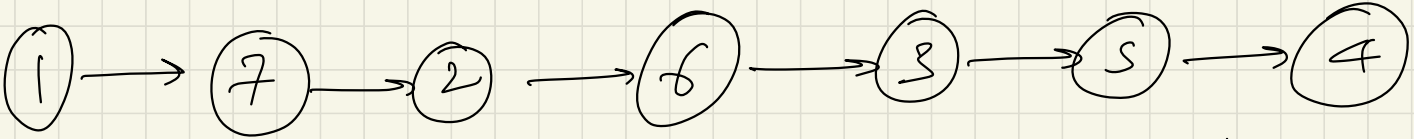


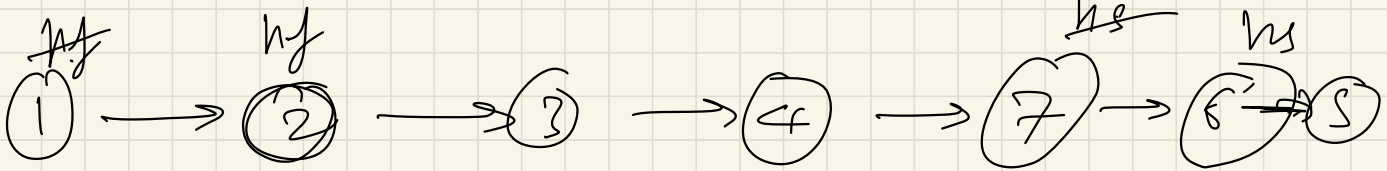


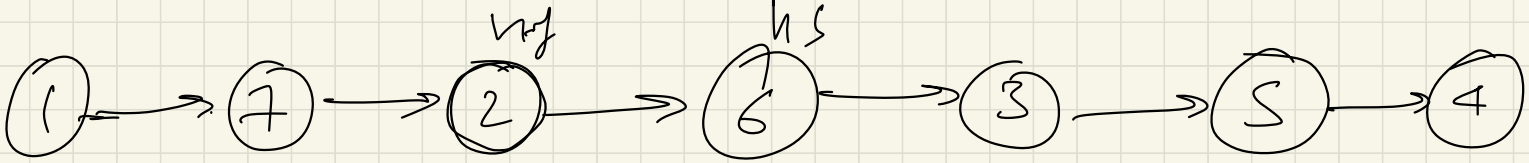
re-reverse the second half.

Q: Pre-order list:









✓ temp = nf.next

nf.next = ns

nf = temp

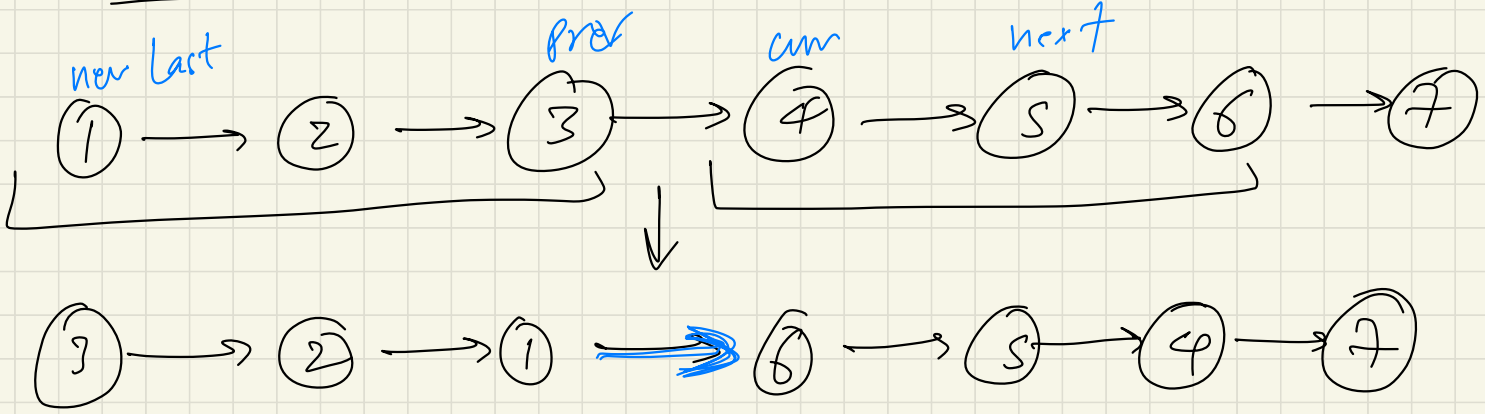
✓ temp = ns.next

ns.next = nf

ns = temp

Q: Rev every k- node :-

k = 3

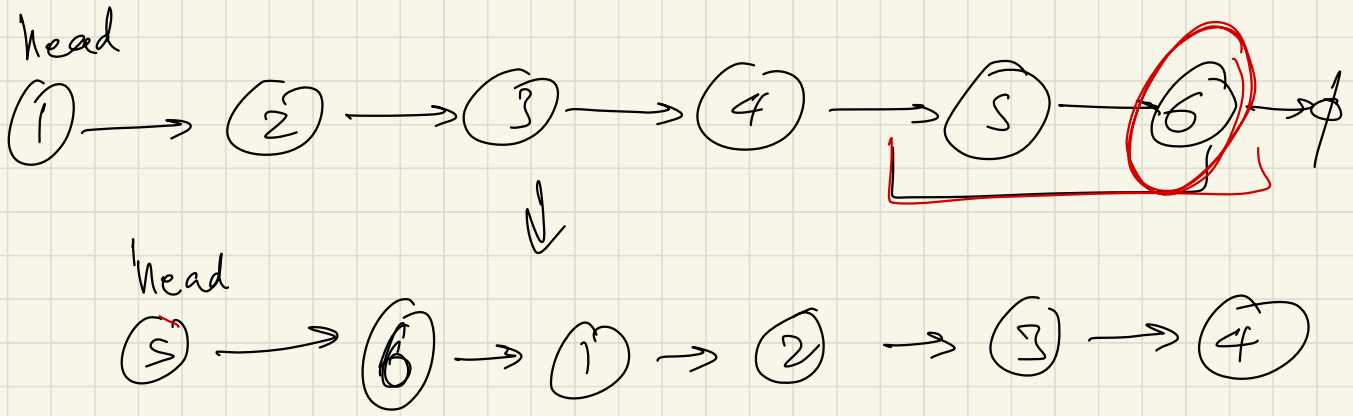


run a for loop k times till curr != ϕ

new last . next = current

Q:

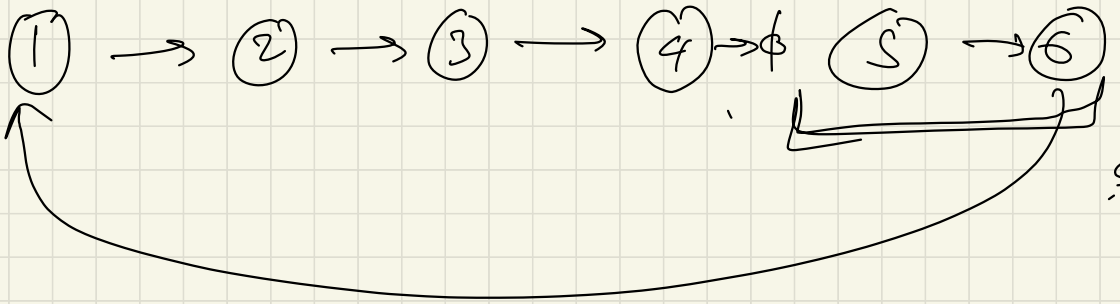
k = 3



① original last node . next → original head

② new head = start of sub list

③ new tail = node before



$k = 2$

$skip = l - k$

$l = 6$

head = newLast.next

newLast.next = ϕ

rotations = $k \% l$