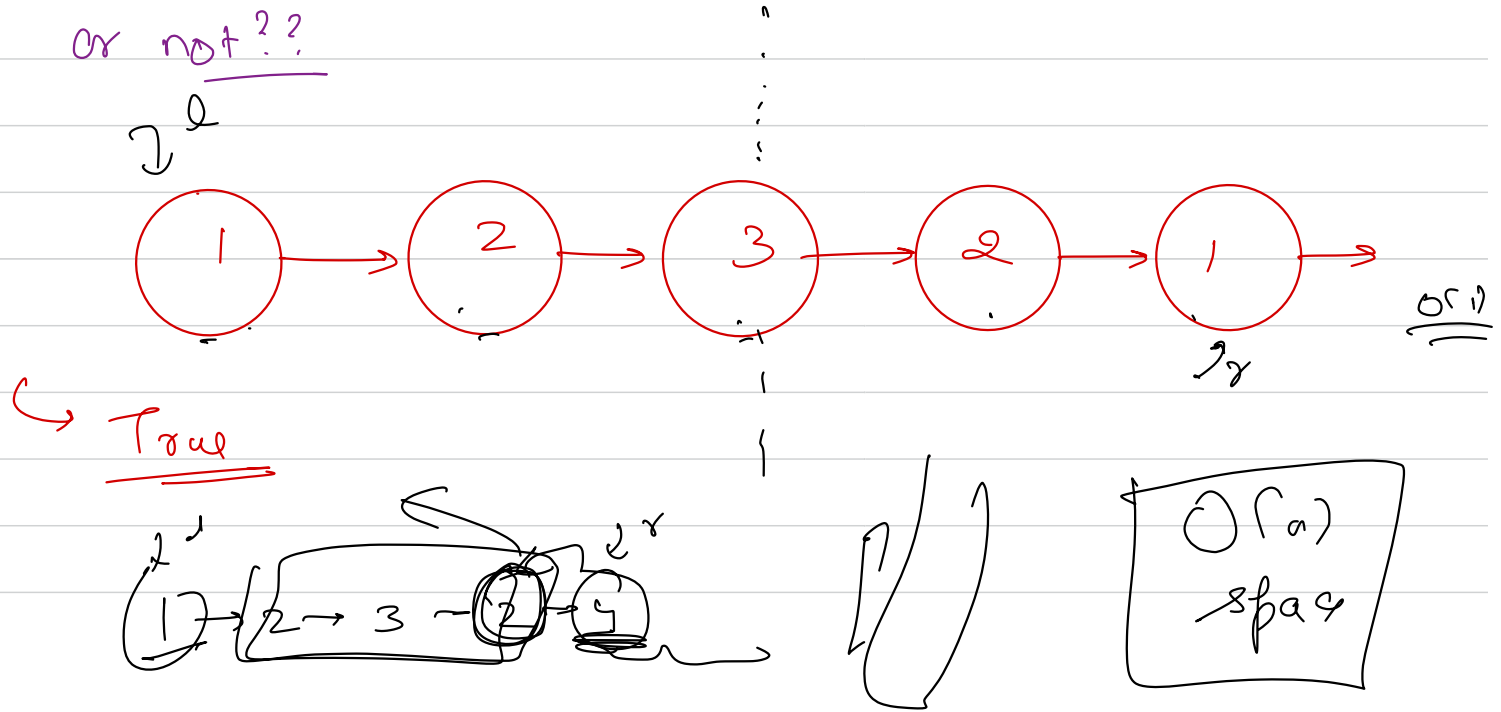


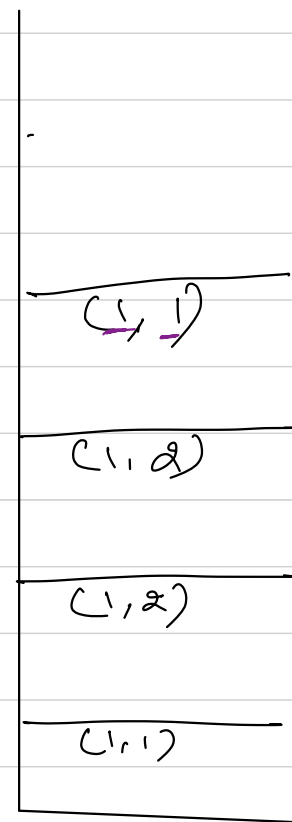
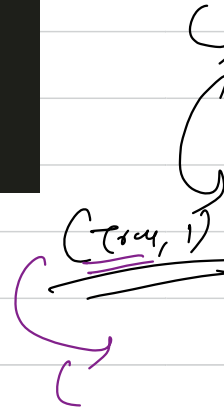
Q ⇒ Given a linked list with all the nodes containing some digits. Check whether the LL is palindromic or not??



```
def check_palindrome(left, right):
    if right != None:
        if not check_palindrome(left, right.next):
            return (False, left)
        if left.data != right.data:
            return (False, left)

        left = left.next
    return (True, left)
```

1 → 2 → 2 → 1

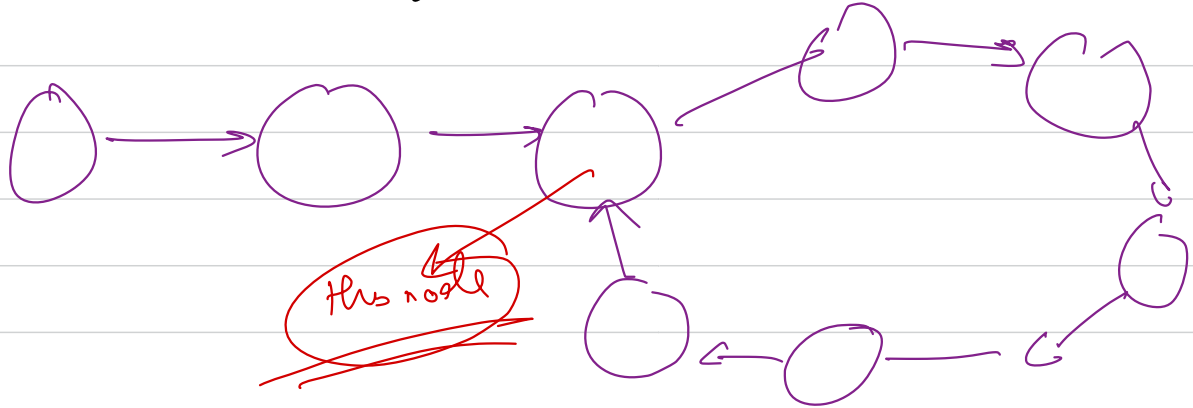


Q.2)

Given an unsorted linked list. Sort the linked list using merge Sort.

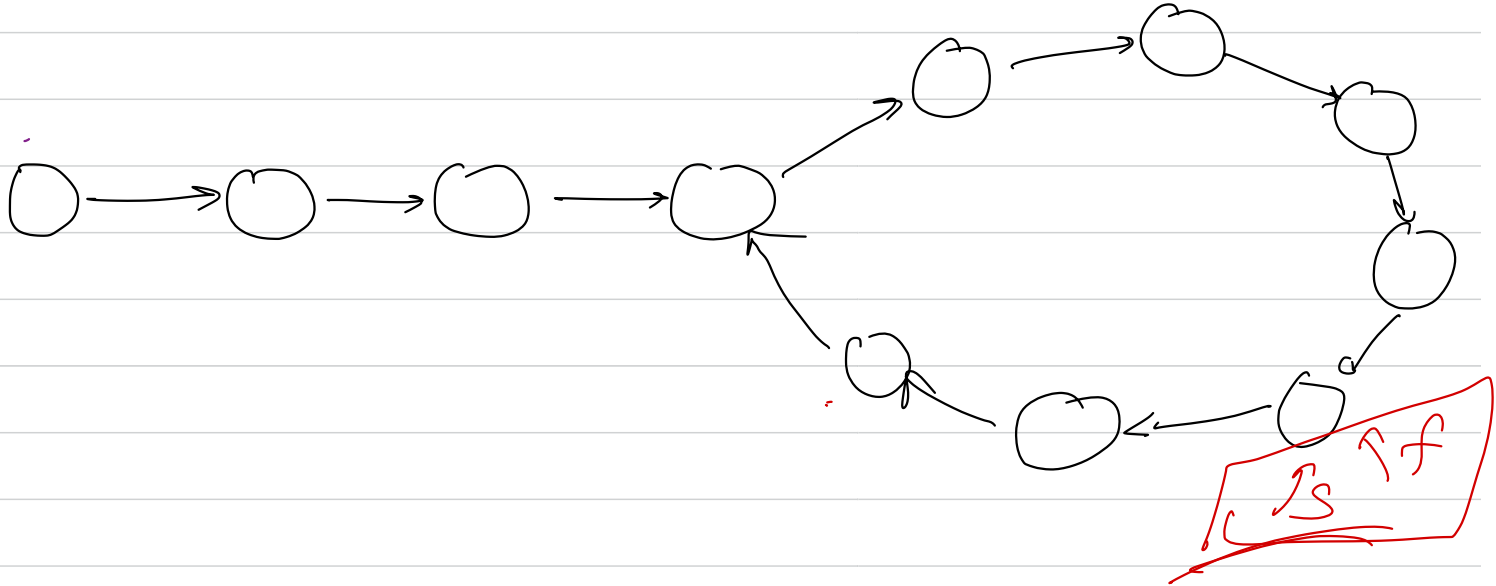
Q. 1

Given a singly linked list such that the last node has the address of some node from the ll in the 'next' data member. Find this value of 'next' of last node.

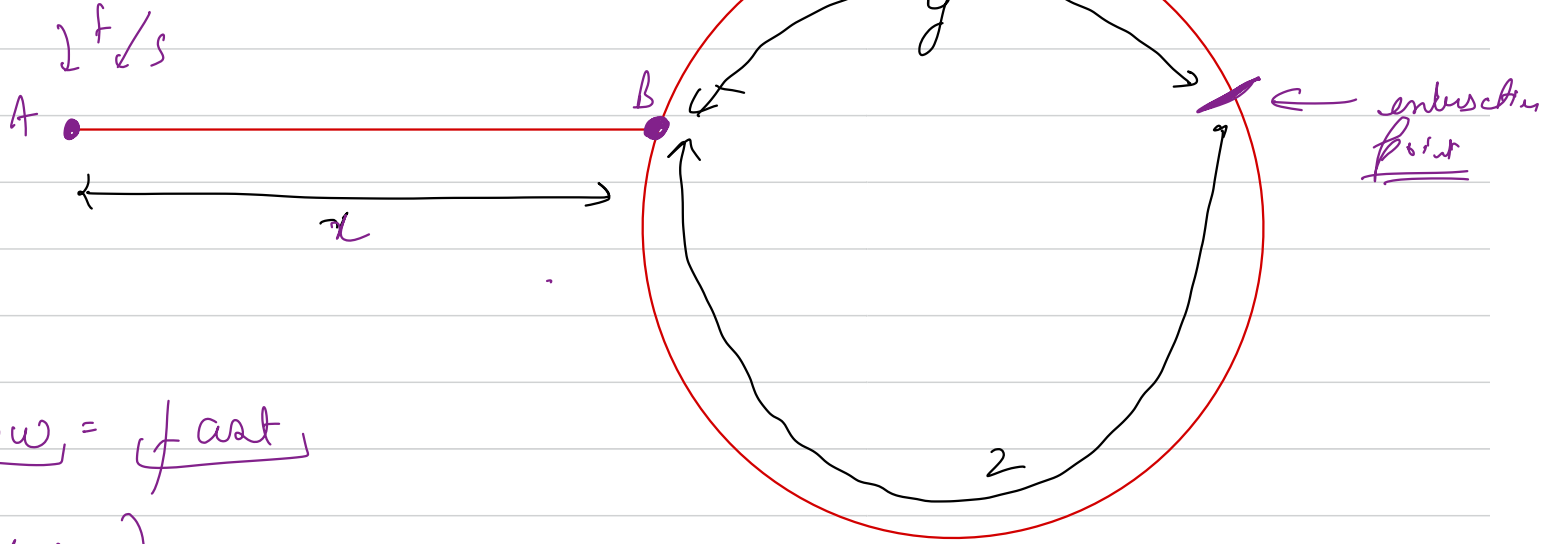


Floyd - cycle - deletion algorithm.

cycle is present



$$C - y = 2$$



$$2 \times \text{slow} = \text{fast}$$

$$2x(x+y) = x + nc + y$$

$$x + y = nc$$

$$A \rightarrow B \rightarrow \text{slow} \rightarrow x \quad \text{fast} \rightarrow 2x \rightarrow \begin{matrix} (x + x) \\ \downarrow \quad \downarrow \\ \text{to B} \quad \text{cycle} \end{matrix}$$

$$C \rightarrow \text{length of cyl}$$

$$C = 2 + y$$

$$n \rightarrow \text{no of times fast ptr completed cycle}$$

$C \rightarrow$ by y cycle

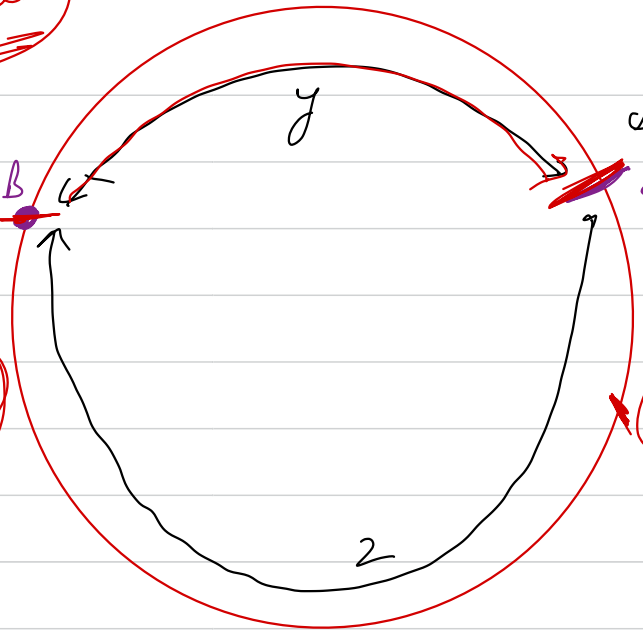
slow

A

x

$$x + y + aC = 2(x + y + bc)$$

$S \rightarrow x$ $f \rightarrow (n + x)$
 \downarrow \downarrow
 AB cycle



slow

entw. point

C

after $\rightarrow C - y$ iterations \rightarrow slow

$$y + 2(C - y) = 2C - y$$

slow

fast

$$x + y + ac = 2x + 2y + 2ac$$

$$x + y = (\underline{a} - \underline{2b}) C$$

multiple of C

$$x + y = K \times C$$

$$\begin{array}{lcl}
 x+y & & x+y+z+y \\
 2(x+y) & = & (x+2y+z) \\
 2x+2y & = & x+2y+z
 \end{array}$$

$$\boxed{x = z}$$

~~course~~



4 different ways