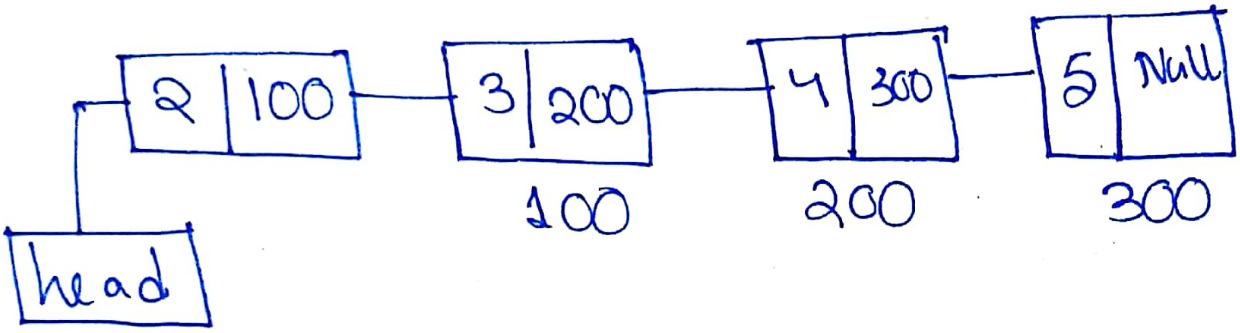
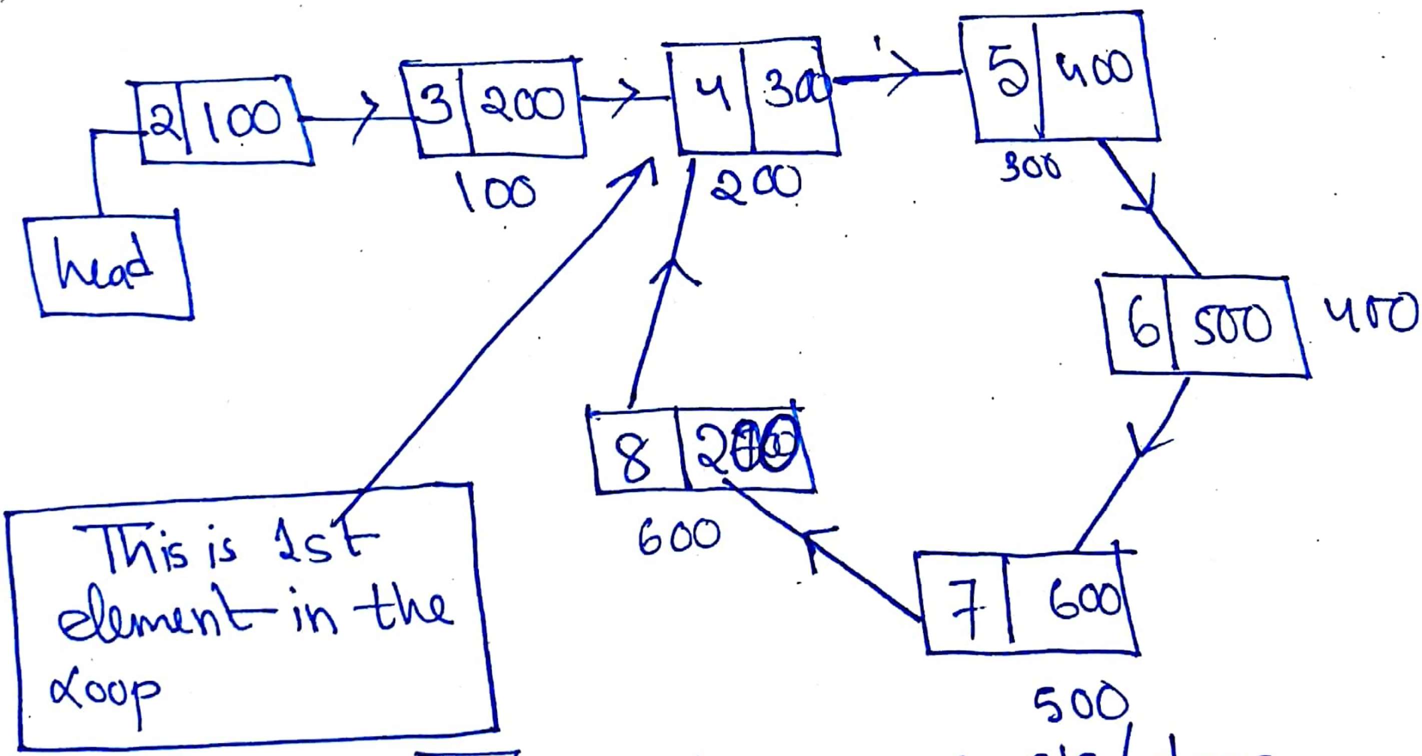
Floyd's Algorithm (Find a circle)

Singly Linkedlist :-



Singly circular linked list :



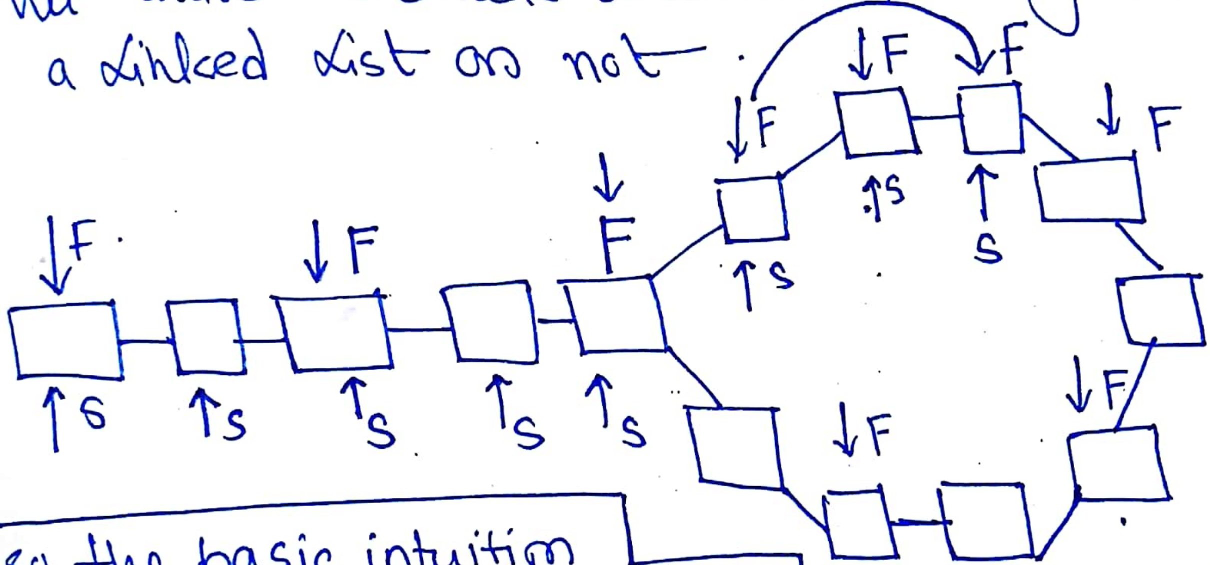
This will be my cycle/loop in the linked list.

How Floyd's algo works / correctness discussion

Here we are using Fast / slow pointer or people called it Hare / Tortoise algorithm.

So the problem statement :-

We have to check whether the cycle exists in a Linked List or not.



So the basic intuition

I can notice that if there exist loop the first & slow will meet and no node→next = NULL.

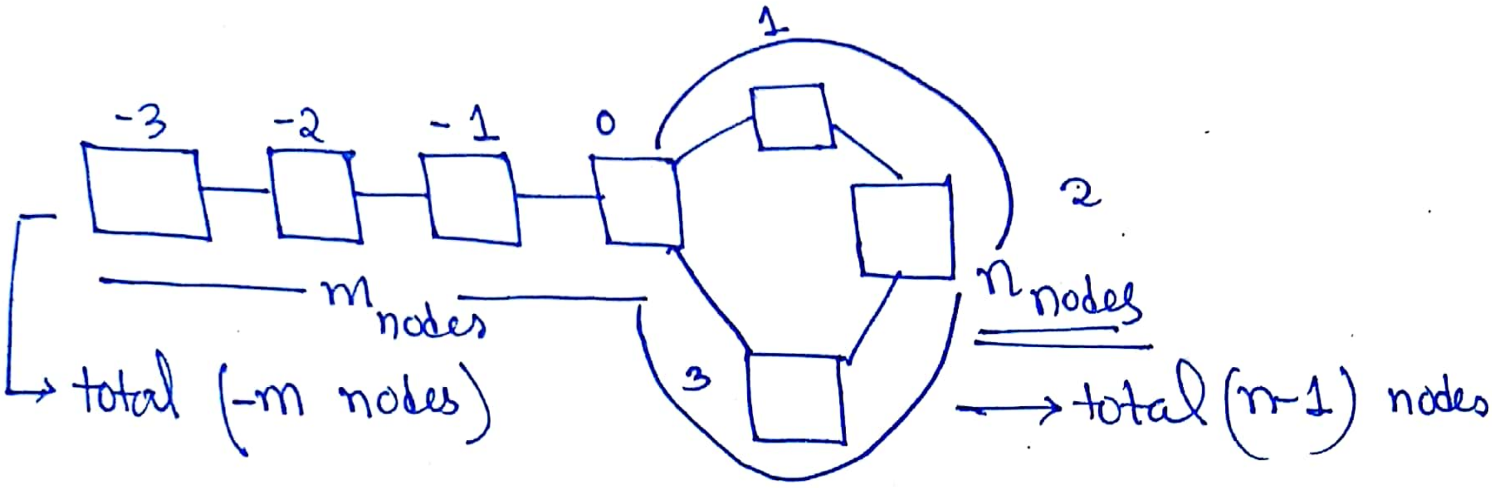
Here slow will be moving 1 steps but fast will certainly move to 2 steps.

fast = fast→next→next;

slow = slow→next;

Let at the begining they are pointing to the same which is head of LL.

Let's try to prove why this works :-



Let's assume the total numbers of node in tail is m and

total numbers of nodes in the cycle is n

$n \in \omega$: $\omega \Rightarrow$ whole number

$+m \in \omega$: $\omega \Rightarrow$ "

$\boxed{m} = \boxed{n}\boxed{c} + \boxed{r}$

nodes in tail

no of nodes in cycle

This is an integer

$\boxed{r = m \% n}$

now there can be 4 possibilities :-

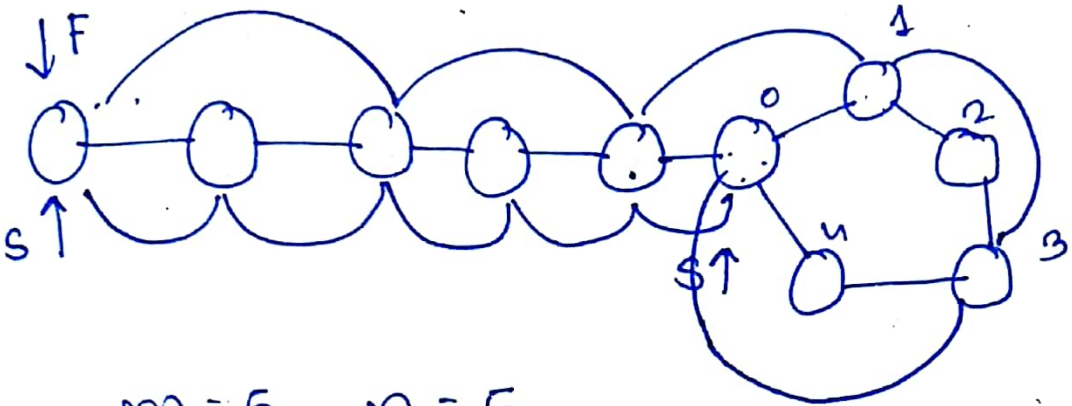
1> $m > n$

2> $m < n$

3> $m = n$.

4> cycle doesn't exist

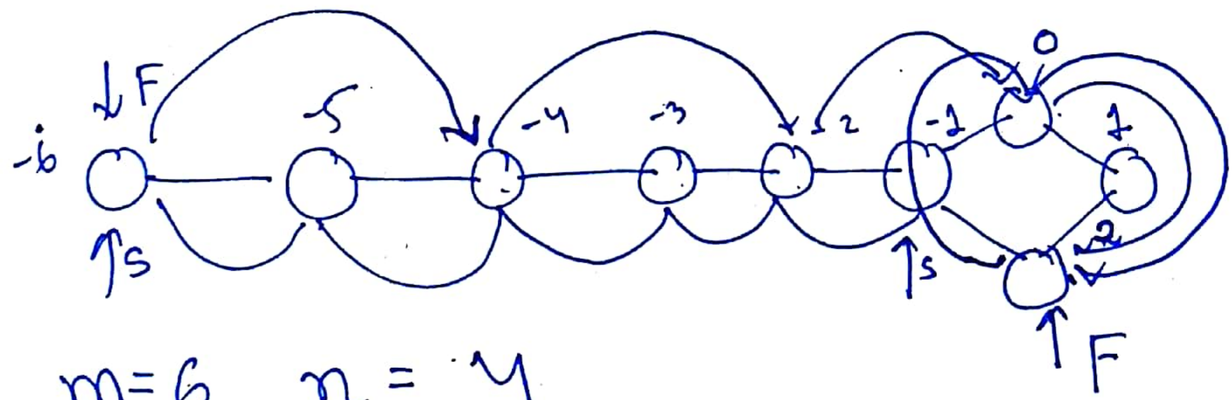
~~case~~ m=n



m=5, n=5

now:- $r = m \% n = 5 \% 5 = 0$.

m>n:-



m=6, n=4

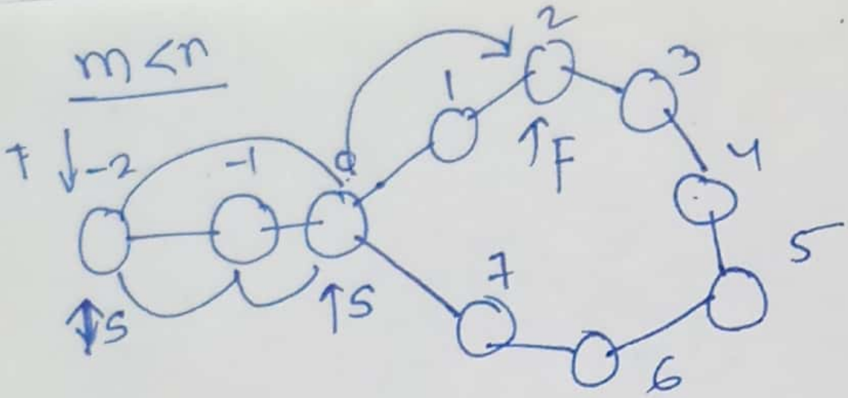
$r = m \% n = 2$

which is same as the F is pointing now.

$\boxed{r = m \% n}$

Correct

m < n



$m = 2$.
$n = 8$.

$m \% n = 2/8 = 2$

$\boxed{r = 2}$ which is correct we can see

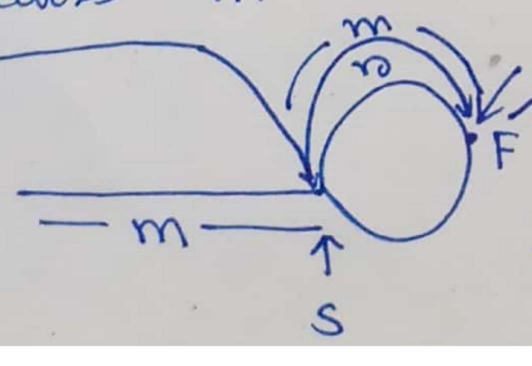
The condition for existance of cycle :-

first and slow will meet atleast once.

Proof :- after m steps →

1> The slow will be at node 0.
2> The first " " " " (r)

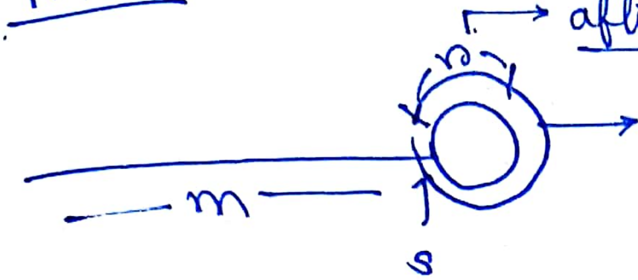
when first covers '2m' number of nodes when slow covers 'm' numbers of nodes.



according to (2).

∴ $m = r \ \forall \ m < n$

$m > n$:-



after same round

first will cover a couple of round as here $m > n$

$m \% n = r > 1$.

so for this case first pointer already cover same rounds so now $r = m - n$.

$m = 15$

$n = 4$

$m = nc + r$

$15 = 4 \times 3 + \underline{3} \rightarrow r$.

After slow moves $n - r$ steps.

Position of turtle / slow $= n - r$.

" " hare / fast $= [2(n-r) + r] \% n$

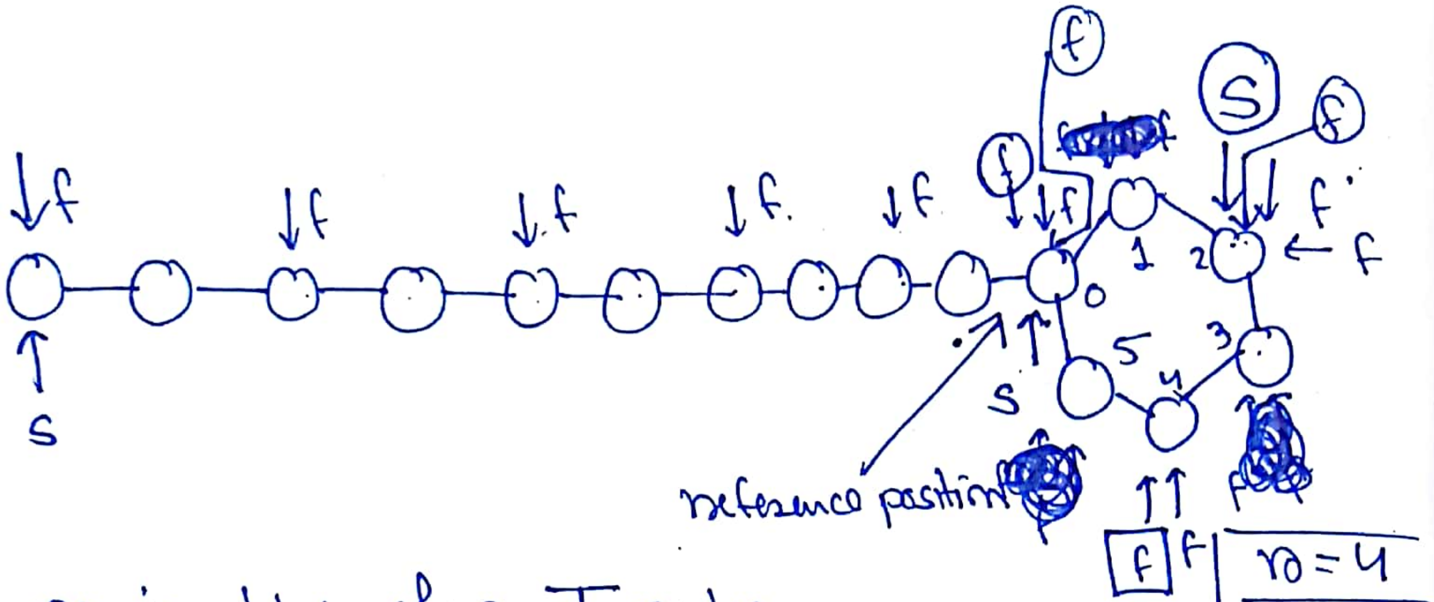
$= (2n - r) \% n$

$$= n \overline{\smash{)}\,2n - r\,}\, 1$$
$$\quad -n$$
$$\quad \overline{n - r}$$

Position of hare = Position of turtle.

(LHS = RHS.) So we have a cycle.

T = 10   C = 6.



1> moving the slow T nodes.

2> check the steps of first and move it by 2XT and check the position

now T%C = 10% 6 = 4 = r

3> Let's move turtle from reference position to C-r nodes = 2 nodes

now pos of slow is 2nd position

now " " fast " 4th "

4> Let's move the first 4 → 0 position

0 → 2 "

∴ pos of slow = pos of fast

after reaching of slow to the reference node and moving it to C-r position.

first & slow will point the same.