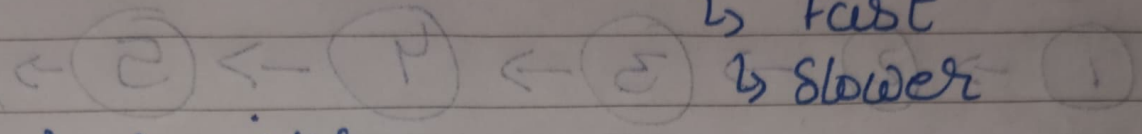
DETECT A CYCLE IN L.L

↳ TWO POINTER APPROACH
  ↳ Fast
  ↳ Slower



PreRequistes
↳
(I) Slow & fast at Head
(II) Slow move one step & will move first.
(III) fast will move two step.

→ if (slow == fast)
Node <--> Node

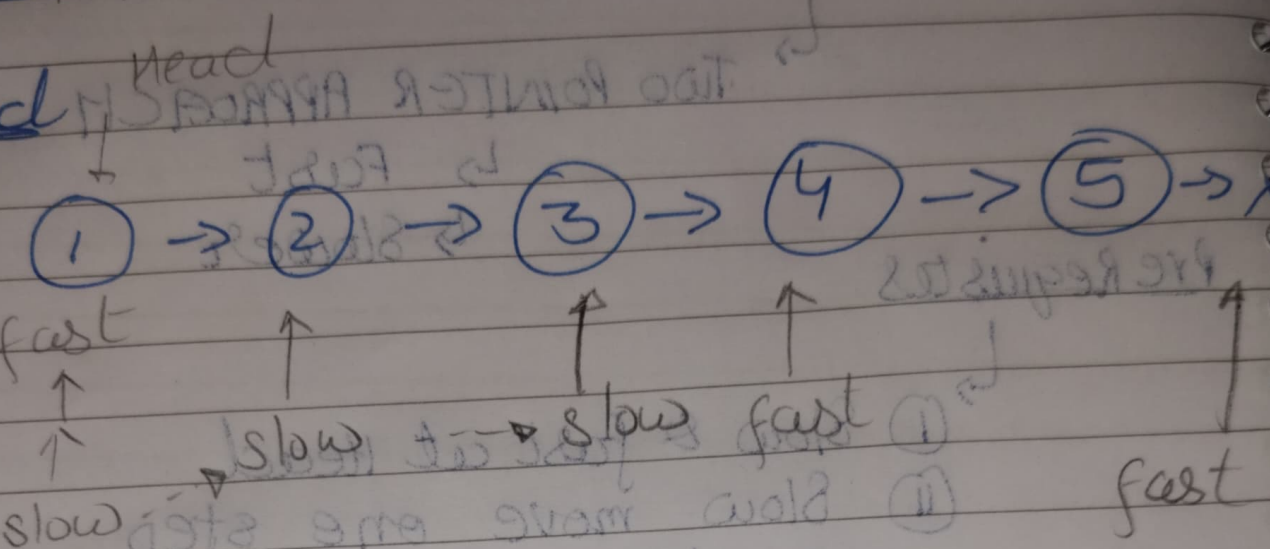
→ while condition ↳ Based on Linear Linked List with odd or Even Nodes

while (fast -> next != NULL && fast

While (fast != NULL && fast -> next != NULL)

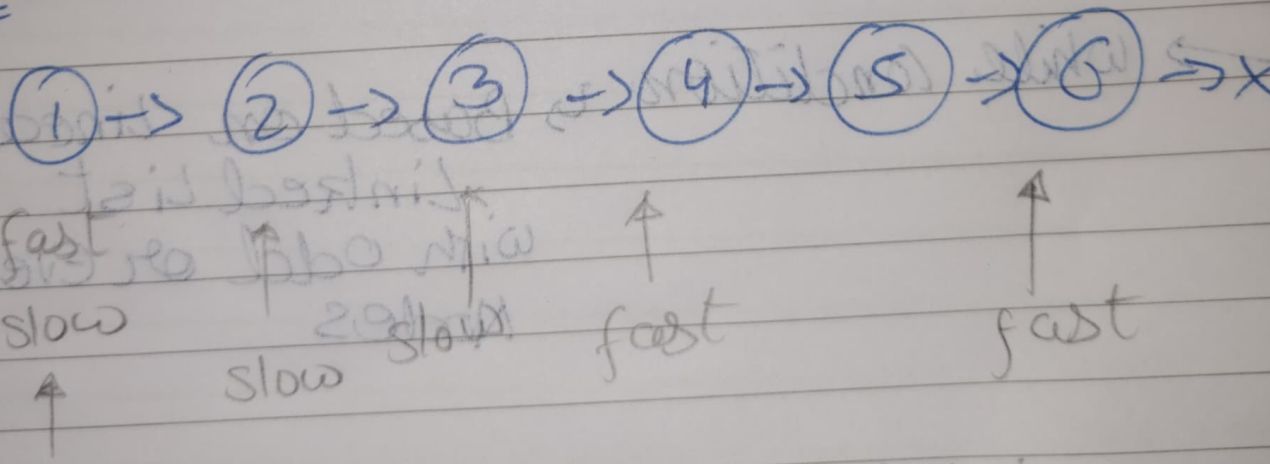
Linear LINKED LIST

Odd



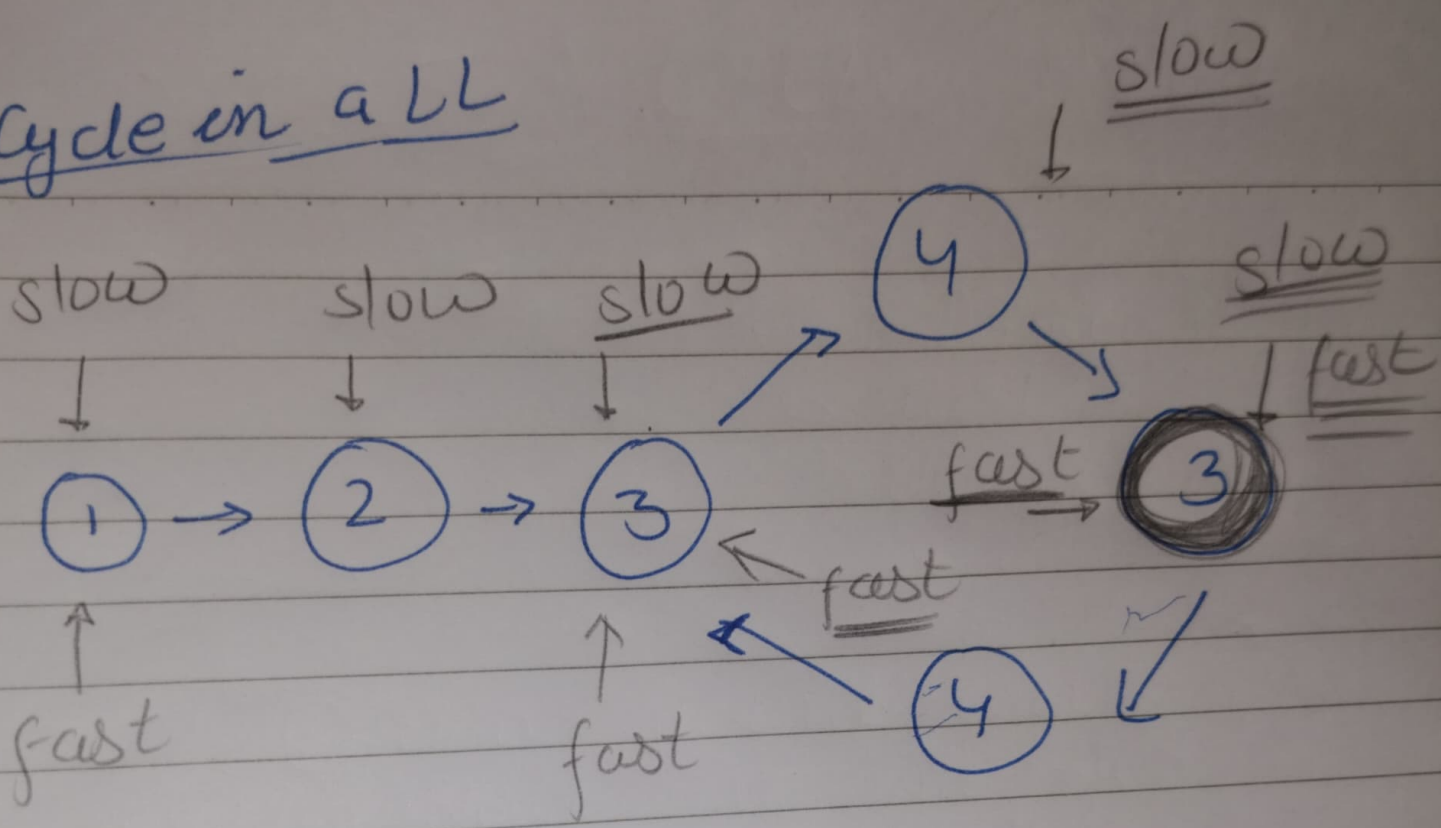

```
while( fast != NULL)
{
}
```

Even




```
while (fast -> next != NULL)
```

Cycle in a LL



→ if (slow == fast)

↳ yes cycle it is.