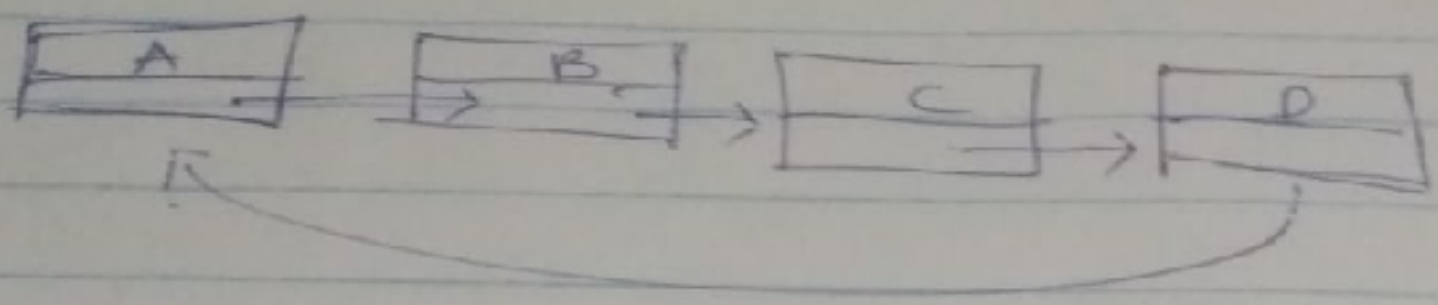
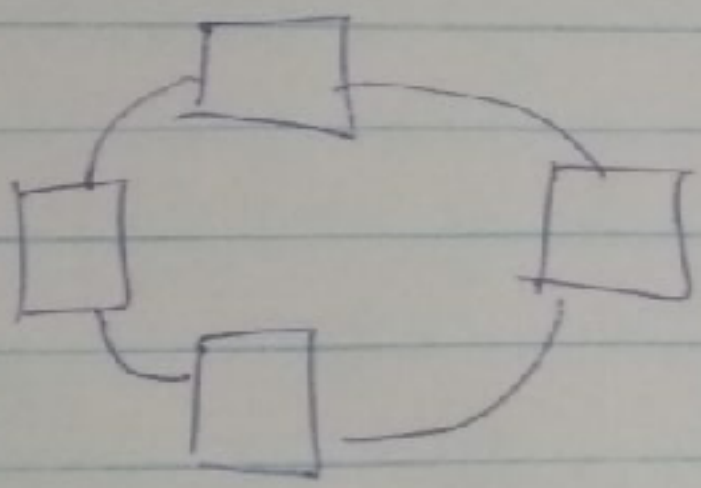
DAY - 8

Circular Linked List

→ just a simple list with the next pointer of the last node of a linked list connected to the first node.



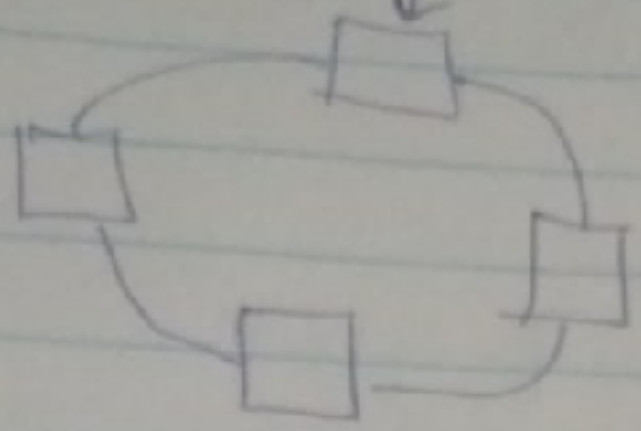
Later it becomes —



→ see, there is no start and end of a circular linked list;

→ but, will maintain a pointer to represent the last element of the linked list as shown in the picture given below.

Look → Last



* here, We have to keep the record of last element and not the first element.

Applications of Circular Linked List:-

→ In multiplayer game, a circular linked list can store the players and give each of them change one by one.

→ When computers share resources with each running application turn by turn and this is done with circular linked list.

→ used to create other data structures like "Queue".

Circula Linked List in python: —
Sample code: —

```
class Node:
    def __init__(self, data):
        self.data = data
        self.next = None
class CircularLinkedList:
    def __init__(self, key):
        z = Node(key)
        z.next = z
        self.last = z
    def insert_after(n, a):
        n.next = a.next
        a.next = n
    def insert_at_last(l, n):
        n.next = l.last.next
        l.last.next = n
        l.last = n
def delete(l, n):
    temp = l.last
    while temp.next != n:
        temp = temp.next
    if n == l.last:  # last node
        if n.next == n:  # only 1 node
            l.last = NULL
```

```
        else :  # more than one node and last node
            temp.next = n.next
            l.last = temp  # updating last pointer
    else:   # not last node
        temp.next
    del n
def traversal(l):
    temp = l.last
    a = str(temp.data) + "\t"
    temp = temp.next
    while temp != l.last:
        a = at str(temp.data) + "\t"
        temp = temp.next
    print(a)
```