* Theory

Why circular queue ?

In a normal queue data structure we can elements are inserts until queue becomes full. But once if queue becomes full. We can't the Next element until all element are deleted from the queue for eg consider the queue below

lets assume we start with a queue for sizes and insert the following element

element 10, 20, 30, 40, 50

Initial state (After Insertion)

Queue [10, 20, 30, 40, 50]
Front : 0
Rear : 4 (points to 50)

After Deleting Three : element

Now we will delete three elements from the queue (10, 20, 30)
step by step deletion

step 1) Dequeue - 1
Remove 10
update front : front = (front + 1) % size
→ front = 1
Queue state [20, 30, 40, 50]

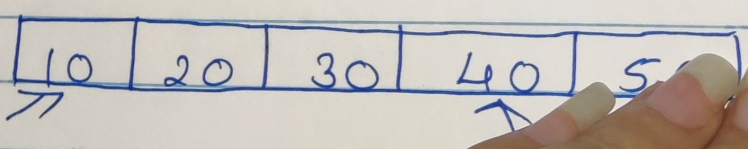
Step 2) Dequeue 2):
Remove 30
update front: front = (front + 1) % size
→ front = 3
- Queue state [-, -, 30, 40, 50]

Step 3) Dequeue 3):
Remove 20
- update front: front = (front) % size
→ front = 3
- Queue state: [-, -, -, 40, 50]

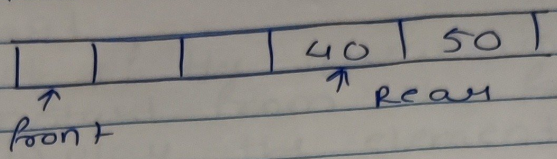
final state queue: [-, -, -, 40, 50]
- front 3 (points to 40)
- Rear 4 (points to 50)

* Why is a circular Queue
A circular queue can be defined as circular queue is a linear data structure are in which the operation are performed based on FIFO (first in first out) principal and the last position is connected to back to the first position make a circle

graphical representation of a circular queue is as follows



After deleting 10, 20, 30 the queue look like this



* Algorithm

1) Enqueue (value) : Inserting value into the circular queue

step1) start
step2) Check if the queue is full
step3) check if the queue is empty
step4) Insert the value
step5) update the queue
step6) end the function

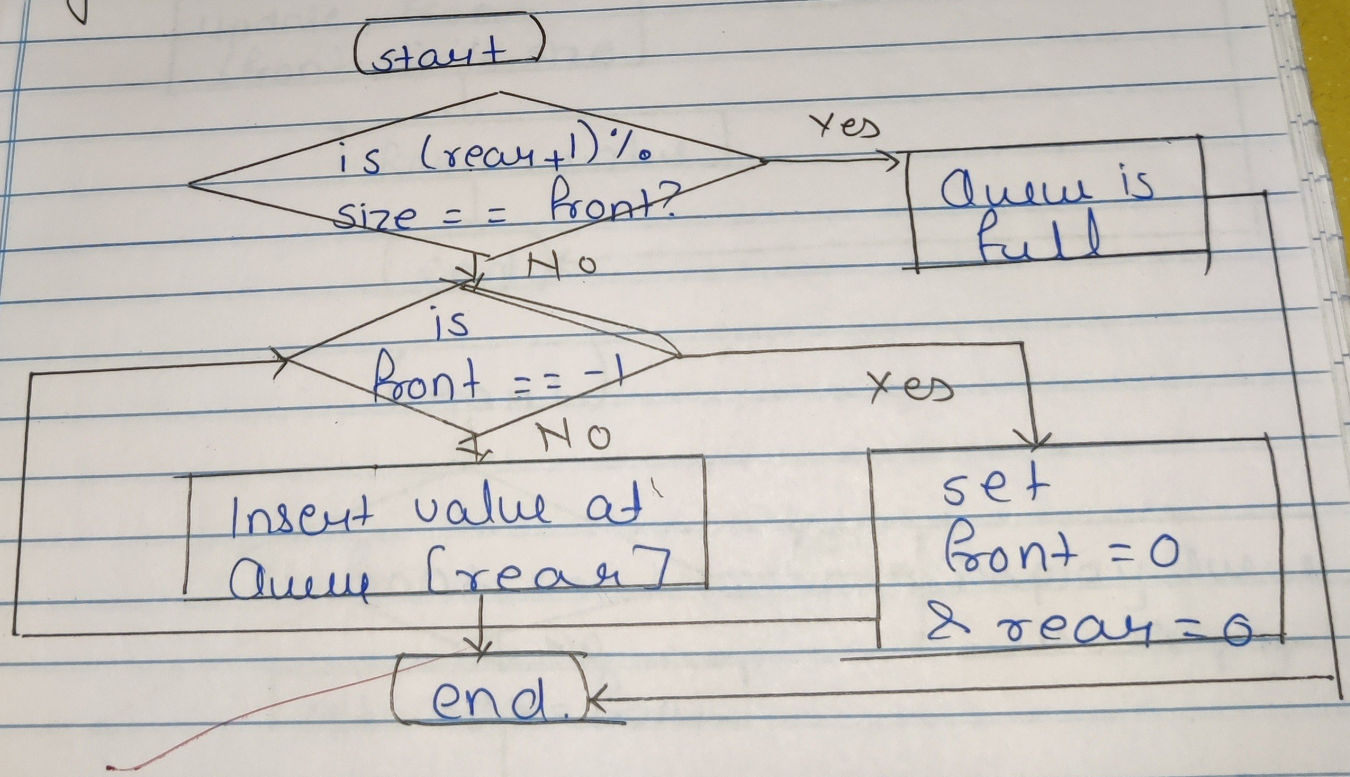
2) deque () - deleting a value from circular queue

step1) start
step2) check if the queue is empty
step3) store the value to Return
step4) check if the queue will be the empty
step5) Update the front pointer
step6) Return the stored value
step7) end

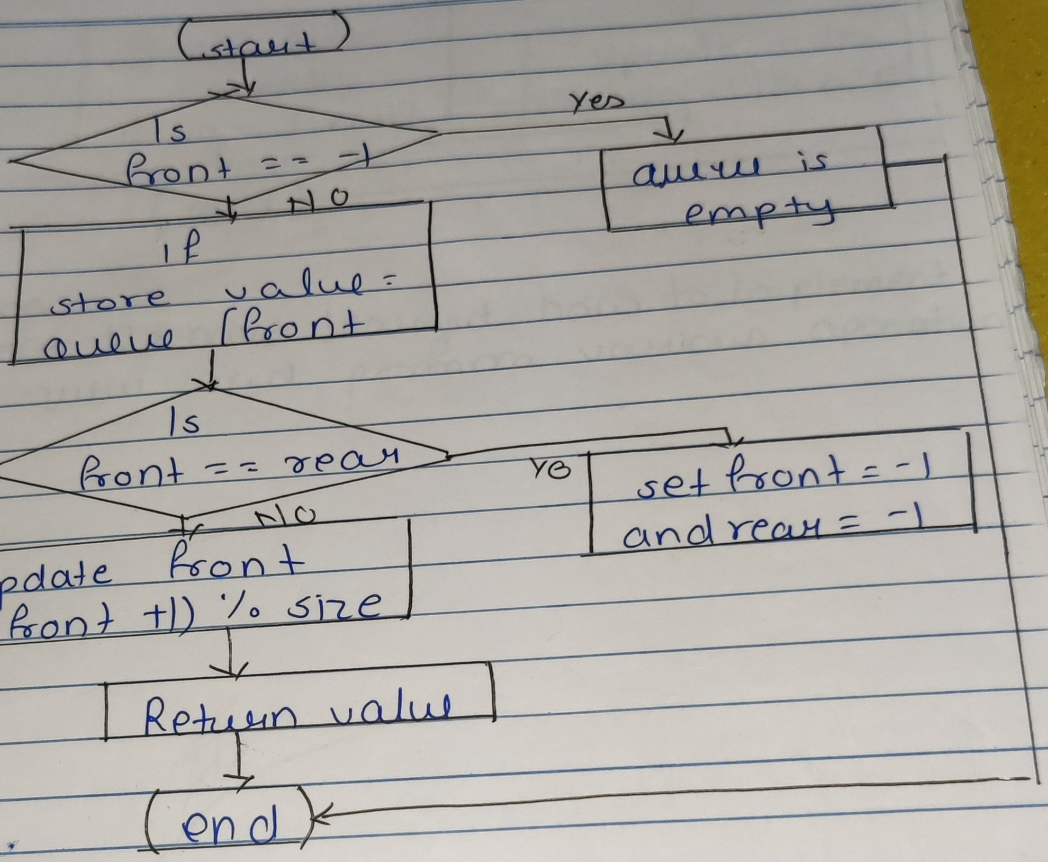
3) display() : display the element of a circular que

step1) start
step2) check if the queue is empty
step3) ~~D~~ start from the front
step4) Display the element
step5) print the last element After the loop print the element at queue [rear] to display the last element
step 6) end the function.

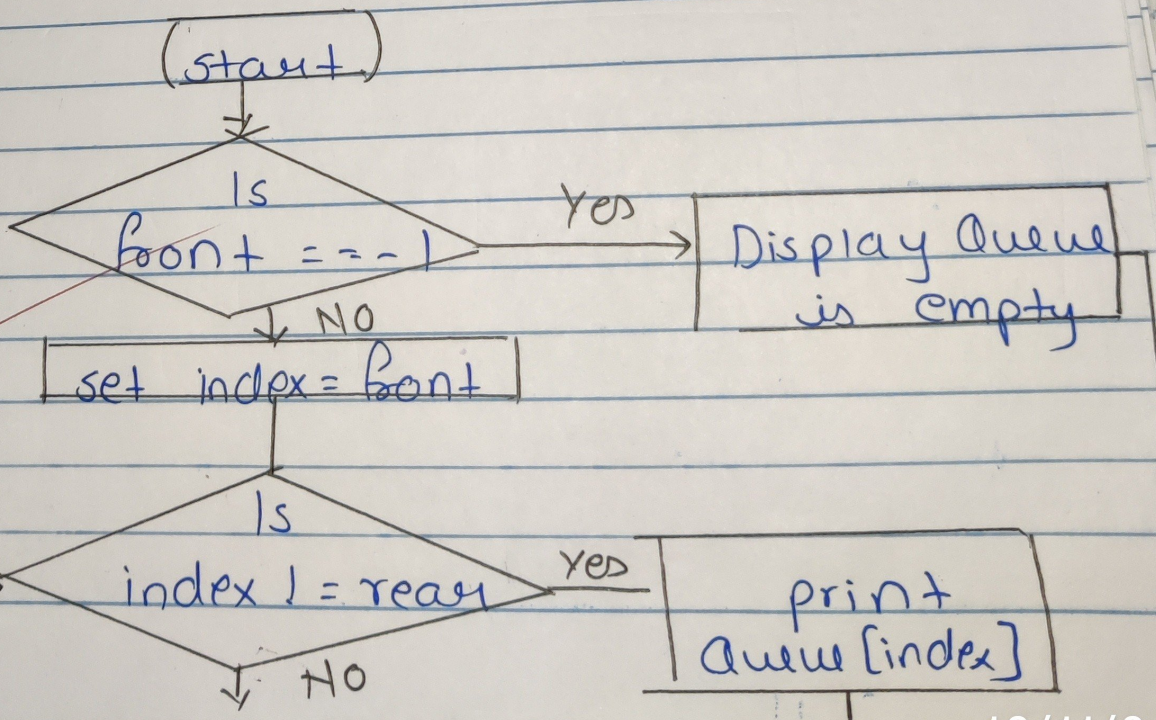
* flow chart

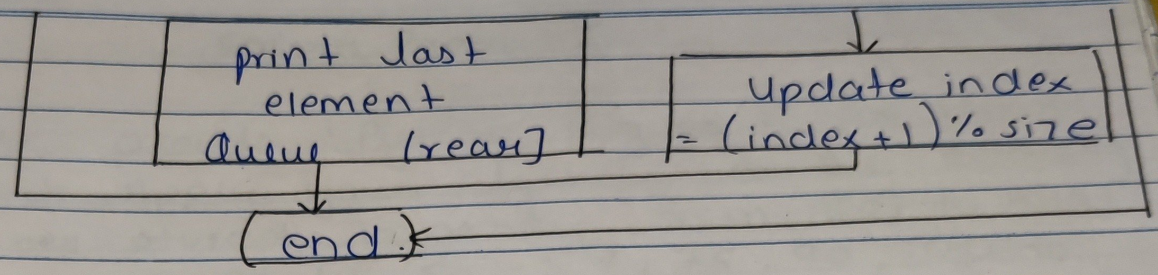


2)



3)





* Conclusion

Hence We have learned how to implement circular queue and perform various operatio on it.