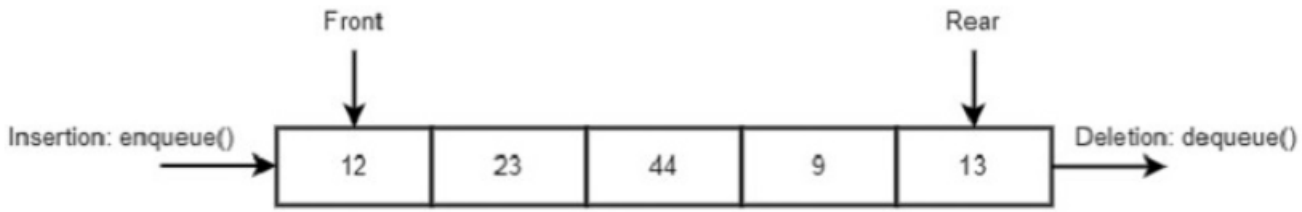
# Queues

# What is Queue?

- A linear data structure where elements are stored in the FIFO (First In First Out) principle



# Operations

- Enqueue (Insertion)
- Algorithm

*1. If count == queue_size, return "Queue Overflow"*

*2. If front == -1, set front = 0 (initial condition)*

*3. Set rear = (rear + 1) % queue_size*

*4. Set queue[rear] = value*

*5. Increment count by 1*

| | | | |
|---|---|---|---|
| E | b | c | d |

*Q_s=4*

*Count=0 ,1,2,3*

*Front=-1,0*

*Rear=-1 (-1+1)%4 =0  (0+1)%4=1, 1+1)%4=2, 2+1)%4=3,  3+1)%4*

# Operations

- Dequeue (Deletion)
- Algorithm
  1. *If count == 0, return "Queue Underflow"*
  2. *Store queue[front] in a variable (optional, for returning)*
  3. *Set front = (front + 1) % queue_size*
  4. *Decrement count by 1*
  5. *If count == 0, set front = rear = -1 (queue is now empty)*

# References


- Queue: https://www.geeksforgeeks.org/queue-data-structure/
- Queue Operations: https://www.geeksforgeeks.org/basic-operations-for-queue-in-data-structure/

- Applications of queue: https://www.geeksforgeeks.org/applications-advantages-and-disadvantages-of-queue/