# Queue

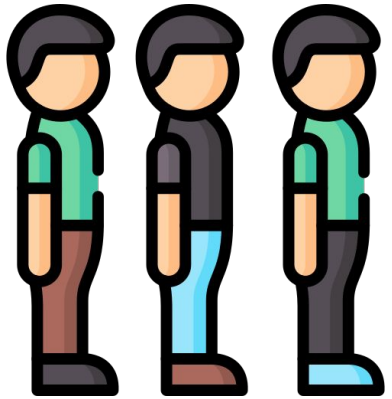
By: Yehia M. Abu Eita

# Outlines

- Introduction
- Enqueue operation
- Dequeue operation
- Queue applications
- How to implement a queue

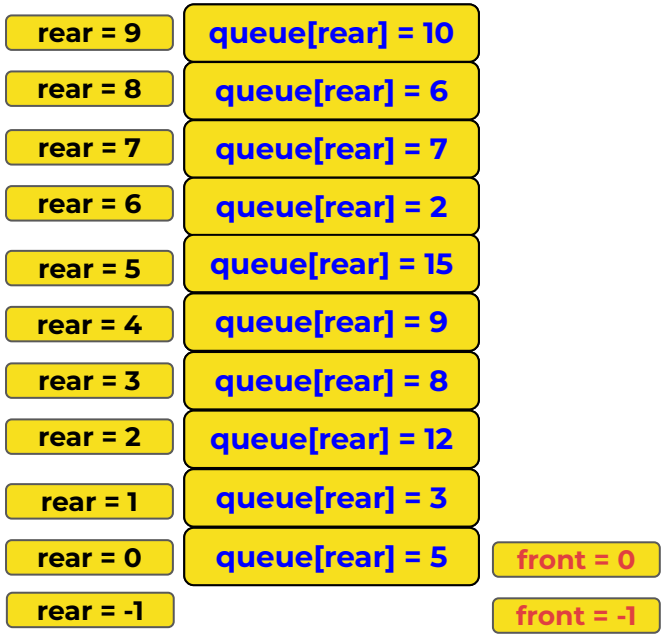
# Introduction

- Queue is a linear data structure that follows **F**irst **I**n **F**irst **O**ut principle.
- A real-life example of a queue is human queue.
- **Inserting** in a queue is called **enqueue**.
- **Enqueue** means **adding** at the **rear** of the queue.
- **Removing** from a queue is called **dequeue**.
- **Dequeue** means **removing** from queue **front**.
- Any queue is characterized by its **size, front** and **rear**.



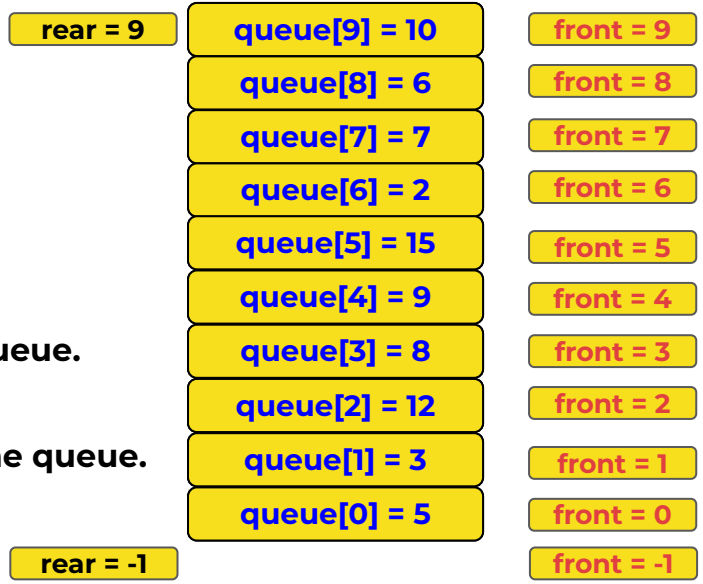
# Enqueue operation

- Enqueue means **adding** at the **rear** of the queue.
- A queue is represented by an array of 10 integers.
- The **front and rear are -1** if the queue is **empty**.
- The **rear is 9** if the **queue is full**.
- **Steps** to enqueue data into the queue:
  - **Check if the queue is full.**
  - **If full, print queue is full error.**
  - **If not full and is empty, increment both rear and front.**
    - **Store data into the queue array.**
  - **If not full and not empty, increment rear.**
    - **Store data into the queue array.**



# Dequeue operation

- Dequeue means **removing** from queue **front**.
- **Steps** to **dequeue** data from the queue:
  - Check if the queue is empty.
  - If empty, print queue is empty error.
  - If not empty and the last element, read data from the queue.
    - Set rear and front to -1.
  - If not empty and not the last element, read data from the queue.
    - Increment the front.



# Queue applications

- Operating System uses queues for job scheduling.
- To handle congestion in the networking queue can be used.
- Sending an email, it will be queued.
- Server while responding to request
- Uploading and downloading photos, first kept for uploading/downloading will be completed first (Not if there is threading)
- While switching multiple applications, windows use circular queue.
- A circular queue is used to maintain the playing sequence of multiple players in a game.

# How to implement a queue

- Declare a **global or local variable** that defines the **queue and its size**, it's usually a structure with **three members**, array and  two integers.
- Implement **enqueue** and **dequeue** functions as the **main functions** of the queue.
- Implement **isEmpty**, **isFull**, **printQueue**, **getQueueFront** and **getQueueRear** functions as a **helper** and **utility** functions.
- Also queues **can be implemented using linked-lists**.

# How to implement a queue

- Use the following **prototypes** as a guide to implement a queue:
  - `typedef struct queue{int elements[QUEUE_SIZE]; int front; int rear}ST_queue_t; // Type`
  - `void createEmptyQueue(ST_queue_t *queue); // Setting queue front and rear to -1`
  - `int enqueue(ST_queue_t *queue, int data);`
  - `int dequeue(ST_queue_t *queue, int *data);`
  - `int printQueue(ST_queue_t *queue);`
  - `int getQueueFront(ST_queue_t *queue);`
  - `int getQueueRear(ST_queue_t *queue);`
  - `int isFull(ST_queue_t *queue);`
  - `int isEmpty(ST_queue_t *queue);`

# Summary

- Now you familiar with the queue data structure.
- Remember that enqueue operation occurs only if the queue is not full.
- Remember that dequeue operation occurs only if the queue is not empty.