Data *Structures*
Queue Homework 1

Mostafa S. Ibrahim
Teaching, Training and Coaching since more than a decade!

Artificial Intelligence & Computer Vision Researcher
PhD from Simon Fraser University - Canada
Bachelor / Msc from Cairo University - Egypt
Ex-(Software Engineer / ICPC World Finalist)

Problem #1: Deque

- Deque is a **Double ended queue** where you can add/remove from either rear or front. It is not FIFO anymore, but provides great flexibility
- Change the **circular queue** to include
    - void enqueue_rear(int value)   [same old code]
    - **void enqueue_front(int value)**
    - int dequeue_front()   [same old code]
    - **int dequeue_rear()**
- Front/Rear meanings shouldn't change.
- O(1) time complexity for all methods

```cpp
Deque dq(6);

dq.enqueue_front(3);
dq.enqueue_front(2);
dq.enqueue_rear(4);
dq.enqueue_front(1);
dq.display();      // 1 2 3 4
cout<<dq.dequeue_rear()<<"\n";    // 4
dq.display();      // 1 2 3
cout<<dq.dequeue_front()<<"\n";  // 1
dq.display();      // 2 3
cout<<dq.dequeue_rear()<<"\n";    // 3
cout<<dq.dequeue_front()<<"\n";  // 2
dq.enqueue_rear(7);
dq.display();      // 7
```

Problem #2: Implement a stack using a single queue

- Let's practice aggregations
- Using Linked-list queue, implement stack functionalities
  - You may add a few extra functionalities to the queue code to make your stack code proper
- What is the time complexity of your push/pop?
- Don't implement display

```
5  class Stack {
6  private:
7      Queue q;
8      int added_elements { };
```

```
Stack stk(3);
stk.push(10);
stk.push(20);
stk.push(30);

while (!stk.isEmpty()) {
    cout << stk.peek() << " ";
    stk.pop();
} // 30 20 10
```

"Acquire knowledge and impart it to the people."

"Seek knowledge from the Cradle to the Grave."