Smart Coding & Interview Series
Top-20 Basic Program
(Queue Applications)

First, understand the solution building strategies and coding for the problems in LIVE/VIDEO session and then you apply those strategies discussed in LIVE/VIDEO session to solve the following problems. Use your favourite language(C/C++/Java/C#/Python/Scala) for coding.

1) Queue Implementation with Circular Array: : Implement queue by circulant array. You need to support the following methods:
1. CircularQueue(n): initialize a circular array with size n to store elements
2. boolean isFull(): return true if the array is full
3. boolean isEmpty(): return true if there is no element in the array
4. void enqueue(element): add an element to the queue
5. int dequeue(): pop an element from the queue

Example

Input: *CircularQueue(5)*
isFull()
isEmpty()
enqueue(1)
enqueue(2)
dequeue()
Output: ["false","true","1"]
Source:https://www.lintcode.com/problem/implement-queue-by-circular-array/description

2) Queue using Stacks: Implement the following operations of a queue using stacks.
- push(x) -- Push element x to the back of queue.
- pop() -- Removes the element from in front of queue.
- peek() -- Get the front element.
- empty() -- Return whether the queue is empty.

Example

```
MyQueue queue = new MyQueue();

queue.push(1);

queue.push(2);

queue.peek();  // returns 1

queue.pop();   // returns 1
```

```
queue.empty(); // returns false
```

Source: *https://leetcode.com/problems/implement-queue-using-stacks/description/*

3) Stack using Queues: Implement the following operations of a stack using queues.
- push(x) -- Push element x onto stack.
- pop() -- Removes the element on top of the stack.
- top() -- Get the top element.
- empty() -- Return whether the stack is empty.

Example:

```
MyStack stack = new MyStack();

stack.push(1);

stack.push(2);

stack.top();   // returns 2

stack.pop();   // returns 2

stack.empty(); // returns false
```

Source: *https://leetcode.com/problems/implement-stack-using-queues/description/*