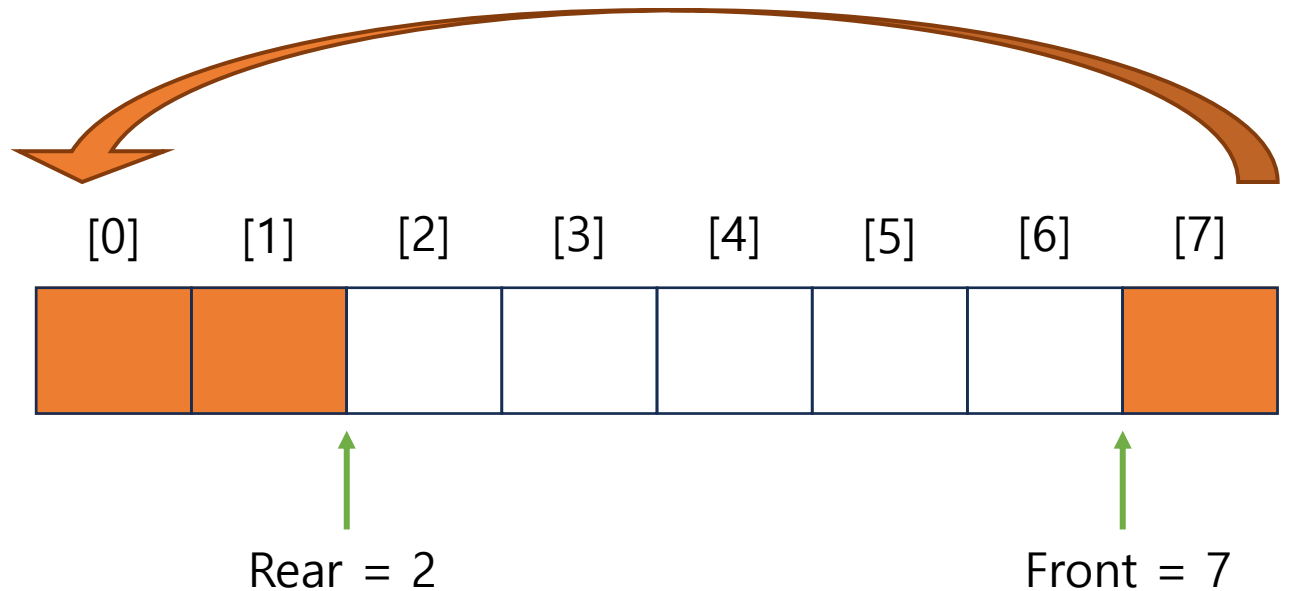
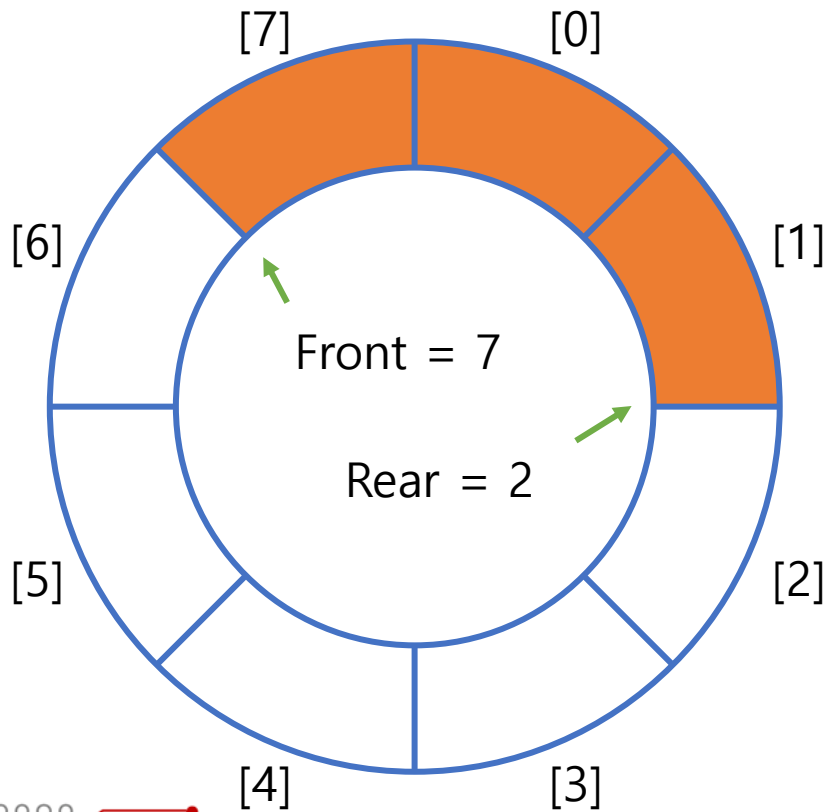


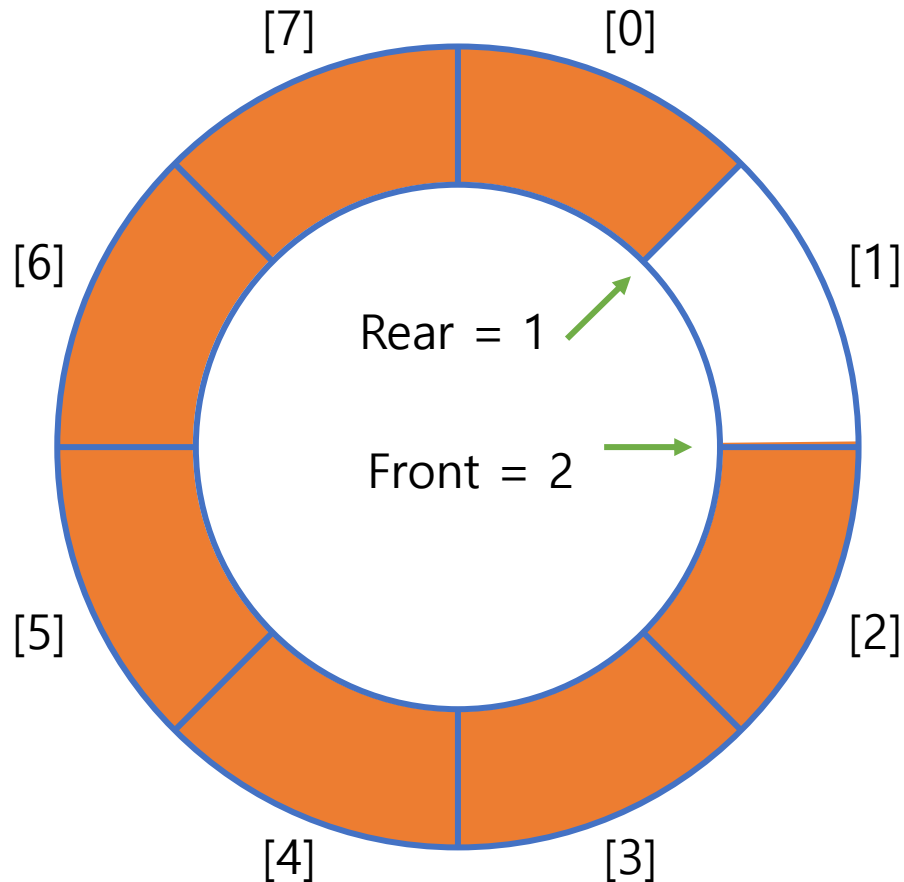
Circular Queue

- ✓ queue는 FIFO(First In First Out) 모델이다
- ✓ queue의 배열의 처음과 끝을 이어 붙여 원형으로 표현한 것이 Circular Queue 이다



Circular Queue

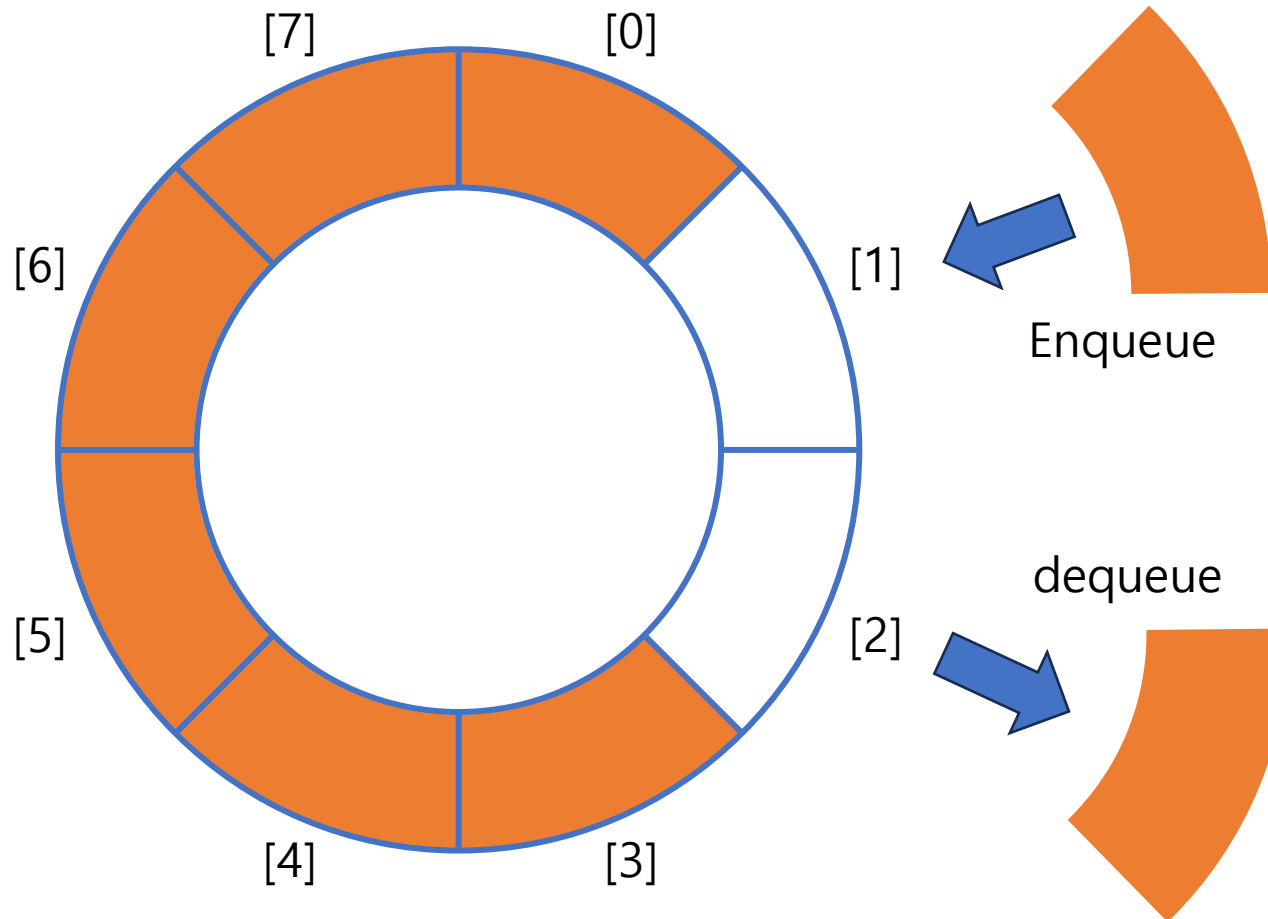
- ✓ Circular Queue는 Front와 Rear를 통해 데이터의 시작과 끝을 알 수 있다.
- ✓ queue 의 데이터 길이는 **배열 길이 - 1** 이다



- ✓ queue 의 시작과 끝을 표시하기 위해 마지막 한 칸은 비워 두어야 한다. (Front 와 Rear 가 같아지면 데이터가 없는 상황과 동일하기 때문)
- ✓ Front 가 Rear + 1 인 경우 queue 는 full 상태이다.

Circular Queue

✓ queue를 사용하기 위해 데이터를 저장하는 함수와 삭제하는 기능이 필요하다



✓ Enqueue : queue에 데이터를 추가
Rear 값을 1 증가시킨다

✓ Dequeue : queue에 데이터를 가져오고 삭제
Front 값을 1 증가시킨다

Circular Queue

✓ Enqueue 와 Dequeue 함수의 Prototype

```
1  #include <stdio.h>
2
3  #define QUEUE_SIZE 8
4
5  int enqueue(int queue[], int front, int* rear, int value);
6  int dequeue(int queue[], int* front, int rear, int* value);
7  void printqueue(int queue[], int front, int rear);
8
9  int main(void) {
10
11     int queue[QUEUE_SIZE];
12     int front = 0;
13     int rear = 0;
14     int input;
15
16 > while(1) { ...
46 }
47
```

3가지 함수에 대한 함수인자는 다음과 같다.

Queue 에 데이터를 저장하는 함수

enqueue(int queue[], int front, int* rear, int value)

Queue 에서 데이터를 가져오는 함수

dequeue(int queue[], int* front, int rear, int* value)

현재 Queue 의 내용을 출력하는 함수

printqueue(int queue[], int front, int rear)

Queue 배열과 front, rear 변수는 main 함수 내에 선언하고, 함수인자로 값을 넘겨 사용한다.

Circular Queue

✓ 프로그램 동작 예시

```
C:\Users\admin\Downloads\circularqueue.exe

=====
Enter number to select mode
1: Enqueue 2: Dequeue 3: Print Queue 4: Exit >> 3
===== Print Queue =====
Queue Empty

=====
Enter number to select mode
1: Enqueue 2: Dequeue 3: Print Queue 4: Exit >> 1
Enter number >> 10
Enqueue Successfully

=====
Enter number to select mode
1: Enqueue 2: Dequeue 3: Print Queue 4: Exit >> 3
===== Print Queue =====
num:0 value:10

=====
Enter number to select mode
1: Enqueue 2: Dequeue 3: Print Queue 4: Exit >> 2
Dequeue Successfully
value : 10

=====
Enter number to select mode
1: Enqueue 2: Dequeue 3: Print Queue 4: Exit >> _
```

Enqueue

Dequeue

Circular Queue

✓ 프로그램 동작 예시

Queue 가 Full 상태일 때 Enqueue 동작

```
=====
Enter number to select mode
1: Enqueue 2: Dequeue 3: Print Queue 4: Exit >> 1
Queue Full
```

Queue 가 Empty 상태일 때 Dequeue 동작

```
=====
Enter number to select mode
1: Enqueue 2: Dequeue 3: Print Queue 4: Exit >> 2
Queue Empty
```

- ✓ Enqueue 가 불가능하거나 Dequeue 가 불가능한 상황에서 예외 처리가 필요하다
- ✓ Enqueue 동작 수행 전 Queue 가 Full 상태인지 확인
 $((rear+1)\%8 == front)$
- ✓ Dequeue 동작 수행 전 Queue 가 Empty 상태인지 확인
if (rear < front)
 rear += QUEUE_SIZE
if (front < rear) { ... }