

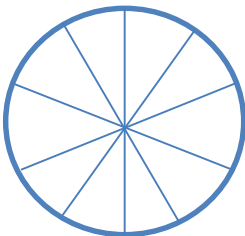
1. 데이터 1, 2, 3을 큐 Q에 넣었다가 꺼내어 출력하면 어떻게 되는가?

1-1. 위 작업을 enqueue와 dequeue으로 표현해 보시오.

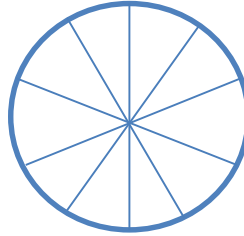
2. 데이터 10, 20, 30, 40, 50을 크기 7인 배열로 구현한 선형큐에 넣고 3개의 항목을 삭제하였을 때 큐의 상태를 나타내 보시오.(front와 rear를 표시한다.)

3. 배열로 구현된 크기가 10인 원형큐에서 front와 rear가 모두 0으로 초기화되었다고 가정하고 다음과 같은 연산 후에 front와 rear의 값은 얼마인지 큐의 상태를 그림으로 그리고 표시하시오.

enqueue(a),
enqueue(b),
enqueue(c),
dequeue(),
enqueue(d),
dequeue()



4. 크기가 10이고 front가 6, rear가 3인 원형큐에서 새로운 항목이 삽입되었을 경우, front와 rear의 값은 어떻게 되는가? 현재 큐의 데이터 항목 개수는? 큐의 상태를 그림으로 표시해보시오.



5. 다음과 같은 원형큐에서 (a)에서 (d)까지의 연산을 차례로 수행한다고 하자. 단계별로 수행 완료된 후의 큐의 상태를 그려라(front, rear 표시). 현재 큐의 상태는 다음과 같으며 []는 인덱스를 나타낸다.

[0]	[1]	[2]	[3]	[4]
	B	C		

- (a) enqueue(A)
- (b) enqueue(D)
- (c) dequeue()
- (d) enqueue(E)

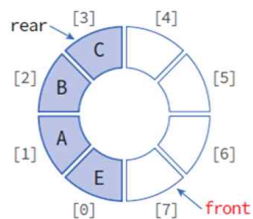
6. 덱은 다음과 같은 연산들을 수행한다.

`add_front(dq, e)`, `delete_front(dq)`,

`add_rear(dq, e)`, `delete_rear(dq)`

덱을 큐처럼 사용하려면 `enqueue()`, `dequeue()` 대신에 어떤 연산들을 호출하여야 하는가?

7. 현재 덱의 상태가 아래와 같다고 하자. 이 상태에서 `add_front(F)`, `add_rear(G)` 명령을 차례대로 실행했을 때 덱 상태는?



8. 크기 5인 덱에서 다음과 같은 연산을 차례대로 수행했다. 덱의 상태를 그림으로 표시해보시오.

`add_front(11)`

`add_rear(22)`

`add_front(33)`

`delete_front()`

`add_rear(44)`

`delete_front()`

`add_front(55)`

`delete_rear()`