

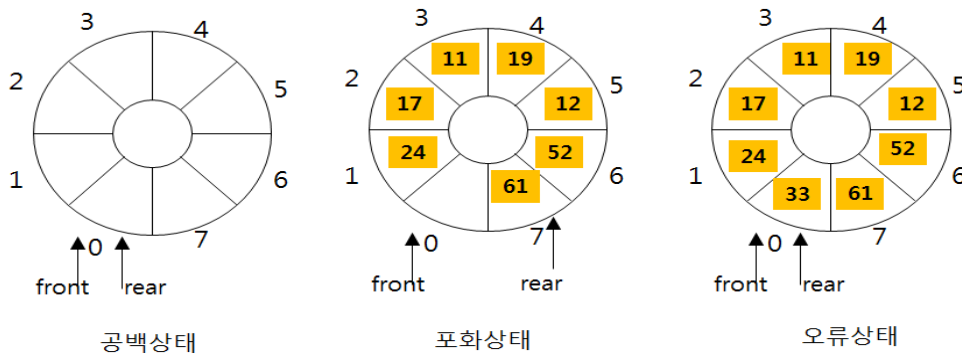
〈자료구조 실습〉 - 큐

※ 입출력에 대한 안내

- 특별한 언급이 없으면 문제의 조건에 맞지 않는 입력은 입력되지 않는다고 가정하라.
- 특별한 언급이 없으면, 각 줄의 맨 앞과 맨 뒤에는 공백을 출력하지 않는다.
- 출력 예시에서 □는 각 줄의 맨 앞과 맨 뒤에 출력되는 공백을 의미한다.
- 입출력 예시에서 ↳ 이 후는 각 입력과 출력에 대한 설명이다.

[문제 1-큐] 배열로 구성된 원형 큐에서의 삽입, 삭제 프로그램을 작성하고, 입력된 순서대로 삽입과 삭제가 이루어 졌을 경우, 원형 큐에 있는 원소 값을 차례대로 출력하시오.

- 문제의 원형 큐에서는 **포화 상태와 공백 상태를 구분**하기 위해 한 자리를 비워둠
 - front, rear, 배열의 초기 값은 0
 - rear의 값을 하나 증가시킨 후 데이터를 큐에 삽입 (출력 예시 1 참고). 또한, front의 값을 하나 증가시킨 후 front가 가리키는 데이터를 삭제함
 - front == rear이면 공백상태로 정의하고, front가 rear보다 하나 앞에 있으면 포화 상태로 정의 함
- ※ 단, 위 초기 값 및 조건은 하나의 구현 예시이고(교재마다 다를 수 있음), 다른 방법 정의를 사용해도 무방하나, **초기 상태에서 맨 처음 삽입되는 위치는 0번이 아니고, 1번이 되어야 함** (그렇지 않으면 문제의 입출력 예시와 다른 결과가 나올 수 있음)



○ 입출력 형식:

- 1) 첫 번째 라인 : 양의 정수 q (원형 큐의 크기)
 - ※ q 값에는 제한이 없다. 또한, 동적 할당을 사용하여야 함
- 2) 두 번째 라인 : 양의 정수 n (연산의 개수)
- 3) 세 번째 이후 라인: n개의 연산이 차례로 입력됨.
 - ※ 연산의 종류는 I (삽입), D (삭제), P (출력)
 - I 10 : 원형 큐에 원소 10을 삽입 함 (큐 원소는 양의 정수)
 - D : 원형 큐에서 데이터를 삭제 함. 또한, 해당 배열의 값을 0으로 함
 - P : **배열에 있는 모든 값들을** 차례로 화면에 출력 함 (입출력 예시 참조)

- ※ **overflow 발생 시** (즉, 포화상태에서 삽입연산이 발생하는 경우),
화면에 overflow 와 배열 값들을 모두 출력하고 프로그램 종료
- ※ **underflow 발생 시** (즉, 공백상태에서 삭제 연산이 발생하는 경우),
화면에 underflow 를 출력하고 프로그램 종료

입력 예시 1

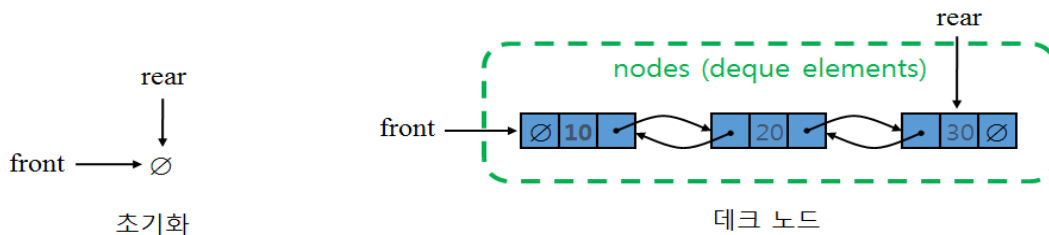
출력 예시 1

6	↳ q = 6	□ 0 10 20 0 0 0	↳ 3번째 연산(P)에 의한 출력
10	↳ n = 10	□ 0 0 20 30 40 0	↳ 7번째 연산(P)에 의한 출력
I 10	↳ 삽입	overflow□ 60 0 20 30 40 50	
I 20	↳ 삽입		↳ 10번째 연산(I 70)에서 overflow 발생
P	↳ 화면출력		
I 30	↳ 삽입		
I 40	↳ 삽입		
D	↳ 삭제		
P	↳ 화면출력		
I 50	↳ 삽입		
I 60	↳ 삽입		
I 70	↳ 삽입		

[문제 2-데크] 데크는 큐의 전단 (front)와 후단 (rear)에서 모두 삽입과 삭제가 가능한 자료구조이다. 이중연결리스트를 사용하여 아래에 정의된 데크 함수들을 구현하시오.

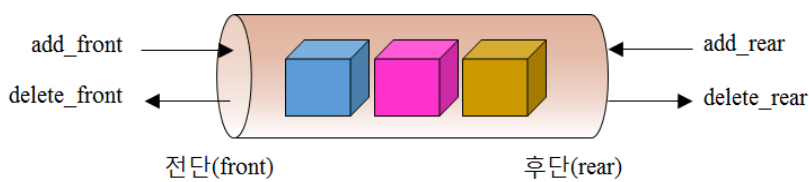
○ 본 문제에서는 헤드 노드와 테일 노드가 없는 이중연결리스트를 사용할 것

- 주의!! 연산 수행 도중에 원소가 모두 삭제되어 데크에 원소가 없는 경우에도, 아래 초기화 상태가 되어야 함



○ 데크 연산

- add_front(deque, X) : deque의 앞에 데이터 X를 추가함
- add_rear(deque, X) : deque의 뒤에 데이터 X를 추가함
- delete_front(deque) : deque의 앞에 있는 데이터를 반환한 다음 삭제함
- delete_rear(deque) : deque의 뒤에 있는 데이터를 반환한 다음 삭제함
- print(deque) : deque의 모든 데이터들을 전단부 부터 저장된 순서대로 출력함



- 입출력 형식:
- 1) 첫 번째 라인 : 연산의 개수 n
 - 2) 두 번째 이후 라인: n개의 연산이 한 줄에 하나씩 차례로 입력됨
 - 하나의 줄에는 연산의 종류, 추가인 경우 원소가 주어짐 (원소는 양의 정수로 표기)
 - 연산의 종류: 다음의 연산 이름이 대문자로 주어짐
- AF (add_front), AR (add_rear), DF (delete_front), DR (delete_rear), P (print)
- ※ underflow 발생 시, 화면에 underflow 를 출력하고 프로그램 종료

입력 예시 1

출력 예시 1

7	↳ 연산의 개수	□20 10 30	↳ 4번째 연산(P)에 의한 출력
AF 10	↳ add_front(deque, 10)	□10	↳ 7번째 연산(P)에 의한 출력
AF 20	↳ add_front(deque, 20)		
AR 30	↳ add_rear(deque, 30)		
P	↳ print(deque)		
DF	↳ delete_front(deque)		
DR	↳ delete_rear(deque)		
P	↳ print(deque)		

입력 예시 2

출력 예시 2

15	↳ 연산의 개수	□30 20 10 40 50	↳ 6번째 연산(P)에 의한 출력
AF 10	↳ add_front(deque, 10)	□10 40	↳ 10번째 연산(P)에 의한 출력
AF 20	↳ add_front(deque, 20)		
AF 30	↳ add_front(deque, 30)		
AR 40	↳ add_rear(deque, 40)		
AR 50	↳ add_rear(deque, 50)		
P	↳ print(deque)		
DF	↳ delete_front(deque)		
DF	↳ delete_front(deque)		
DR	↳ delete_rear(deque)		
P	↳ print(deque)		
DF	↳ delete_front(deque)		
DR	↳ delete_rear(deque)		
DR	↳ delete_rear(deque)		