

## 9주차 자료구조/알고리즘 실습 - 덱

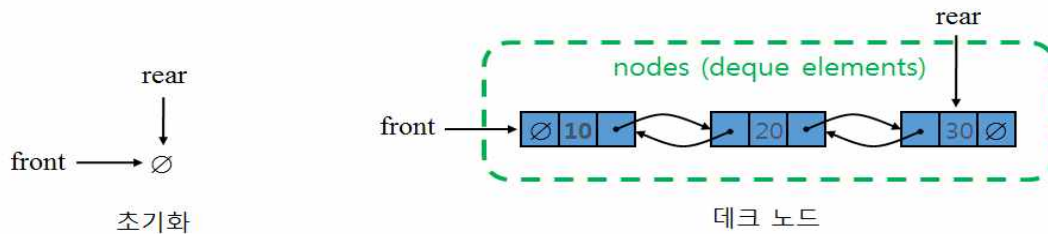
### ※ 입출력에 대한 안내

- 특별한 언급이 없으면 문제의 조건에 맞지 않는 입력은 입력되지 않는다고 가정하라.
- 입출력 예시에서 □는 출력되는 공백을 의미한다.
- 입출력 예시에서 ↳ 이후는 각 입력과 출력에 대한 설명이다.

[ 문제 1-덱 ] **덱**은 큐의 전단(front)과 후단(rear)에서 모두 삽입과 삭제가 가능한 자료구조다.  
헤드 노드와 테일 노드가 없는 이중연결리스트를 사용하여 아래에 정의된 덱 함수들을 구현하시오.

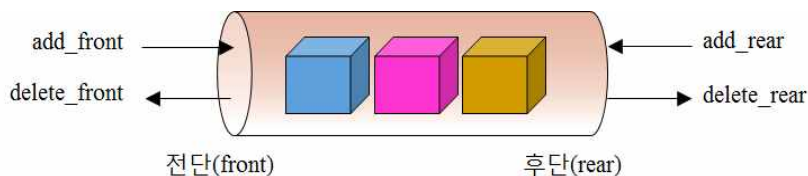
#### ○ 초기 상태

- 주의 : 연산 수행 도중 원소가 모두 삭제되어 덱이 비는 경우에도, 아래 초기 상태가 되어야 함.



#### ○ 덱 연산

- **add\_front(X)** : deque의 앞에 원소 X를 추가
- **add\_rear(X)** : deque의 뒤에 원소 X를 추가
- **delete\_front(deque)** : deque의 앞에 있는 원소를 삭제
- **delete\_rear(deque)** : deque의 뒤에 있는 원소를 삭제
- **print\_deque(deque)** : deque의 모든 원소들을 전단부터 후단까지 차례로 출력.



○ **입출력 형식:**

- 1) 첫 번째 라인 : 연산의 개수 **n**
- 2) 두 번째 이후 라인: **n**개의 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\_deque)

※ underflow 발생 시, 화면에 underflow를 출력하고 프로그램 종료.

입력 예시 1

7	↳ 연산의 개수
AF 10	↳ add_front(10)
AF 20	↳ add_front(20)
AR 30	↳ add_rear(30)
P	↳ print(deque)
DF	↳ delete_front()
DR	↳ delete_rear()
P	↳ print(deque)

출력 예시 1

□20 10 30	↳ 4번째 연산(P)에 의한 출력
□10	↳ 7번째 연산(P)에 의한 출력

입력 예시 2

15	↳ 연산의 개수
AF 10	↳ add_front(10)
AF 20	↳ add_front(20)
AF 30	↳ add_front(30)
AR 40	↳ add_rear(40)
AR 50	↳ add_rear(50)
P	↳ print(deque)
DF	↳ delete_front()
DF	↳ delete_front()
DR	↳ delete_rear()
P	↳ print(deque)
DF	↳ delete_front()
DR	↳ delete_rear()
DR	↳ delete_rear()

출력 예시 2

□30 20 10 40 50	↳ 6번째 연산(P)에 의한 출력
□10 40	↳ 10번째 연산(P)에 의한 출력
underflow	↳ 13번째 연산(DR)에서 underflow발생. 실행을 종료함