10주차 과제(덱 실습)

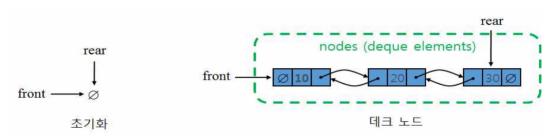
※ 입출력에 대한 안내

- 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-데크] 데크는 큐의 전단(front)과 후단(rear)에서 모두 삽입과 삭제가 가능한 자료구조다. <u>헤드 노드와 테일 노드가 없는 이중연결리스트</u>를 사용하여 아래에 정의된 데크 함수들을 구현하시오. (이중연결리스트로 구현이 어렵다면, 원형 데크로 구현)

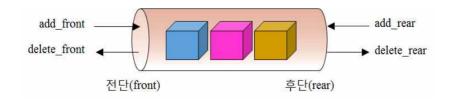
○ 초기 상태

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



○ 데크 연산

- add_front(X) : deque의 앞에 원소 X를 추가
- add_rear(X) : deque의 뒤에 원소 X를 추가
- delete_front(deque) : deque의 앞에 있는 원소를 반환한 다음 삭제
- delete_rear(deque) : deque의 뒤에 있는 원소를 반환한 다음 삭제
- print(deque) : deque의 모든 원소들을 전단부터 후단까지 차례로 출력.



○ 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 underflow를 출력하고 프로그램 종료.

입력 예시 1

출력 예시 1

7	→ 연산의 개수	□20 10 30	↦ 4번째 연산(P)에 의한 출력
AF 10	\mapsto add_front(10)	□10	↦ 7번째 연산(P)에 의한 출력
AF 20	\mapsto add_front(20)		
AR 30	\mapsto add_rear(30)		
P	\mapsto print(deque)		
DF	\mapsto delete_front()		
DR	\mapsto delete_rear()		
Р	\mapsto print(deque)		

입력 예시 2

출력 예시 2

15	→ 연산의 개수	□30 20 10 40 50 → 6번째 연산(P)에 의한 출력
AF 10	\mapsto add_front(10)	□10 40
AF 20	\mapsto add_front(20)	underflow → 13번째 연산(DR)에서
AF 30	\mapsto add_front(30)	underflow발생. 실행을 종료함
AR 40	\mapsto add_rear(40)	
AR 50	\mapsto add_rear(50)	
Р	\mapsto print(deque)	
DF	\mapsto delete_front()	
DF	\mapsto delete_front()	
DR	\mapsto delete_rear()	
P	\mapsto print(deque)	
DF	\mapsto delete_front()	
DR	\mapsto delete_rear()	
DR	<pre> → delete_rear()</pre>	

※ 제출 시 유의사항

- 학번-이름-문제번호.py 파일에 작성한 파이썬 코드를 제출 (문제가 1개인 과제의 경우 문제번호는 생략 가능)
- 한 개의 PDF 문서 파일에 코드 실행 결과를 캡처하여 제출
- 파일들을 압축하지 않고 제출
- 직접 작성한 파이썬 스크립트를 import 하여 사용할 수 없음