



# 0x07 덱

☼ 상태	완료
👤 담당자	서연

## 덱이란? Double Ended Queue



덱은 큐와 스택을 합쳐놓은 자료구조로, 양 끝에서 삽입과 삭제가 모두 가능한 자료구조임.

## 덱의 성질

삽입, 삭제, 제일 앞/뒤 원소의 확인이 다  $O(1)$ 의 시간 복잡도를 가짐.

## 구현

### 배열을 이용한 구현

#### 1. 원소를 담을 큰 배열 한 개

#### 2. 앞쪽, 뒤쪽을 가리킬 변수 두 개

- head와 tail의 초기값이 0이 아니라 MX임
- 시작 지점을 배열의 중간으로 두어야 함.  $\Rightarrow$  양쪽에서 삽입과 삭제가 일어나기 때문에 양쪽으로 확장이 가능하게 됨.
  - 그래서 배열의 크기는  $2*MX+1$ 이고 head와 tail의 초기값은 MX임.

```

public class Main {
    // 배열 기반 덱(Deque)을 위한 변수들
    static final int MX = 1000005;
    static int[] dat = new int[2 * MX + 1];
    static int head = MX;
    static int tail = MX;

    // 덱의 맨 앞에 요소를 추가
    public static void push_front(int x) {
        dat[--head] = x;
    }

    // 덱의 맨 뒤에 요소를 추가
    public static void push_back(int x) {
        dat[tail++] = x;
    }

    // 덱의 맨 앞 요소를 제거
    public static void pop_front() {
        head++;
    }

    // 덱의 맨 뒤 요소를 제거
    public static void pop_back() {
        tail--;
    }

    // 덱의 맨 앞 요소를 반환
    public static int front() {
        return dat[head];
    }

    // 덱의 맨 뒤 요소를 반환
    public static int back() {
        return dat[tail - 1];
    }

    // 테스트 함수

```

```
public static void test() {
    push_back(30); // 30
    System.out.println(front()); // 30
    System.out.println(back()); // 30
    push_front(25); // 25 30
    push_back(12); // 25 30 12
    System.out.println(back()); // 12
    push_back(62); // 25 30 12 62
    pop_front(); // 30 12 62
    System.out.println(front()); // 30
    pop_front(); // 12 62
    System.out.println(back()); // 62
}

// 메인 함수
public static void main(String[] args) {
    test();
}
}
```