

# 스택 수열

$N=8$

1번 수열: 4

2번 수열: 3

4
3
2
1

+

-

+

+

3번 수열: 6

6
5
2
1

+

-

+

4번 수열: 8

5번 수열: 7

6번 수열: 5

7번 수열: 2

8번 수열: 1

8
7
5
2
1

+

-

+

-

-

-

$N=5$

1번 수열: 1

2번 수열: 2

2
1

+

-

+

-

3번 수열: 5

4번 수열: 3

5
4
3

+

-

+

+

X

↓

NO

### 1. 로직에 쓸 Stack 생성, N 입력받기

```
Stack<Integer> stack = new Stack<>();
```

```
BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
```

```
int N = Integer.parseInt(br.readLine());
```

### 2. 출력값 값을 StringBuilder 생성

```
StringBuilder sb = new StringBuilder();
```

### 3. 수열 만들기

```
int[] arr = new int[N];
```

```
for(int i=0; i<N; i++){
```

```
    arr[i] = Integer.parseInt(br.readLine());
```

```
}
```

### 4. 로직에 필요한 자열씩 선인(stack에 대해 삽입)

```
int num = 1;
```

### 5. 수열이 만족되지 않는 경우 시스템 종료시키기 위해 boolean 추가

```
boolean flag = true;
```

## 6. 수열 값 찾기 시작

```
for (int i=0; i<N; i++){
```

## 7. 수열 값이 지정된 값보다 큰 경우 찾아볼 때까지 삽입

```
if (arr[i] >= num){  
    while (arr[i] > num){  
        stack.push(num++);  
        sb.append(" +\n");  
    }  
}
```

## 8. 수열 값과 peek() 같다면 pop

```
if (arr[i] == stack.peek()){  
    stack.pop();  
    sb.append(" -\n");  
}
```

## 9. 다른 경우 수열 탐색 X

```
} else {  
    System.out.println("NO");  
    flag = false;  
    break;  
}
```

```
{  
}
```

## 10. flag가 true일 때만 sb 출력

```
if (flag){  
    System.out.println(sb.toString());  
}
```