

Chapter. 01
기본 자료구조

| 핵심 유형 문제풀이

FAST CAMPUS
ONLINE
유형별 문제풀이

강사. 나동빈

Chapter. 01

기본 자료구조(핵심 유형 문제풀이)

I 혼자 힘으로 풀어 보기

문제 제목: 스택 수열

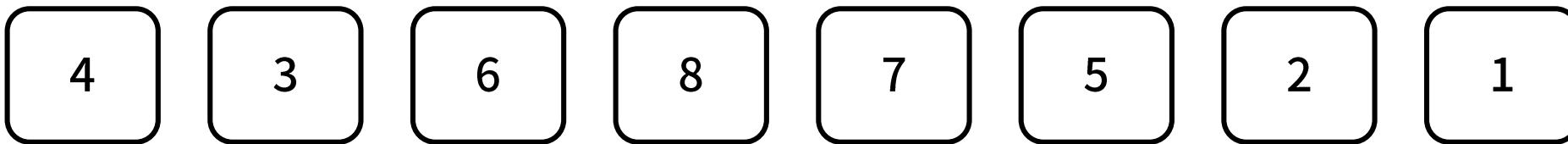
문제 난이도: 하(Easy)

문제 유형: 스택, 그리디

추천 풀이 시간: 30분

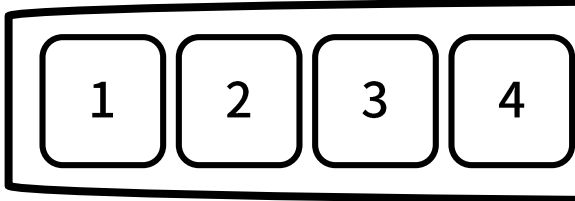
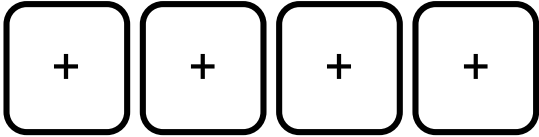
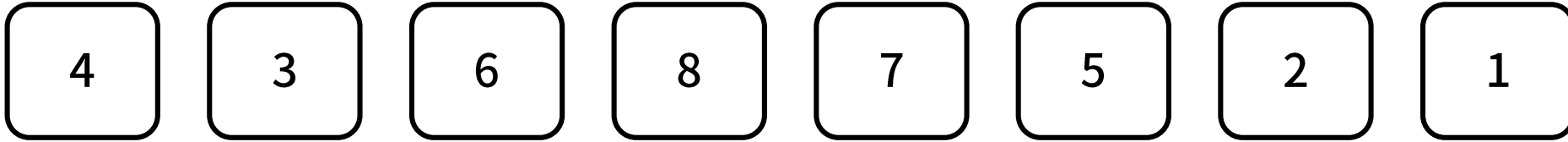
I 문제 풀이 핵심 아이디어

1. 스택에 원소를 삽입할 때는, 단순히 특정 수에 도달할 때까지 삽입하면 됩니다.
2. 스택에서 원소를 연달아 빼낼 때 내림차순을 유지할 수 있는지 확인합니다.



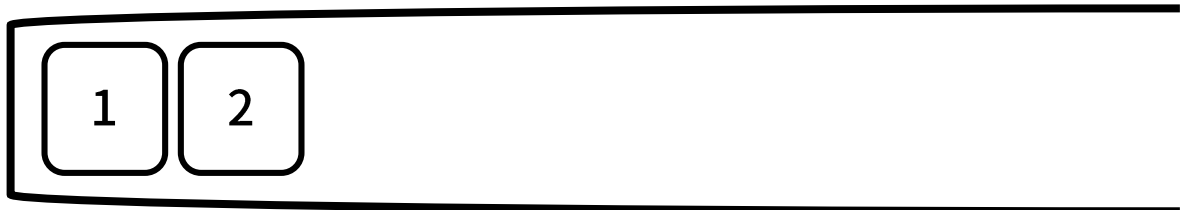
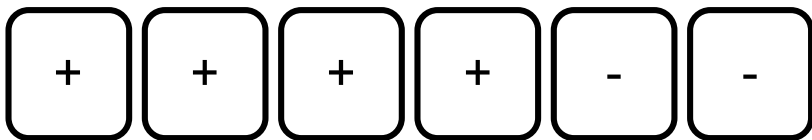
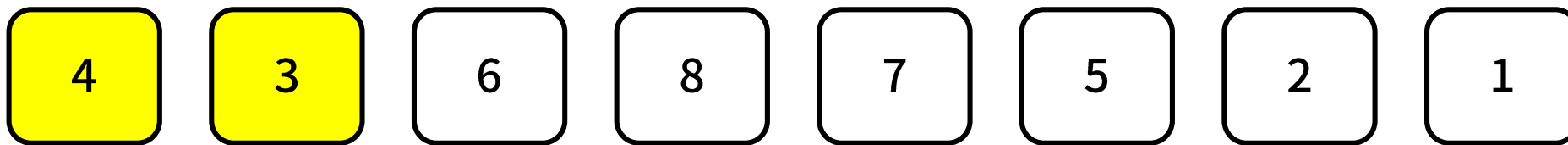
I 문제 풀이 핵심 아이디어

1. 스택에 원소를 삽입할 때는, 단순히 특정 수에 도달할 때까지 삽입하면 됩니다.
2. 스택에서 원소를 연달아 빼낼 때 내림차순을 유지할 수 있는지 확인합니다.



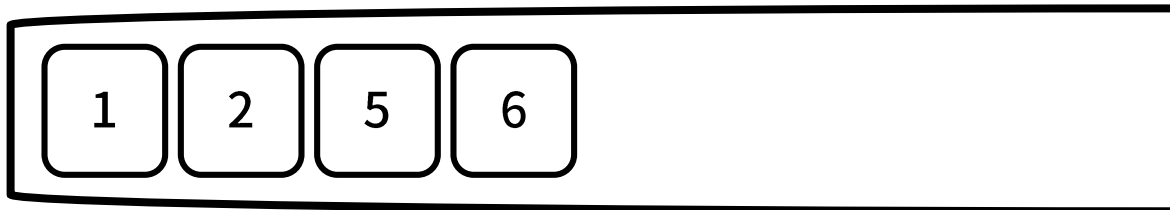
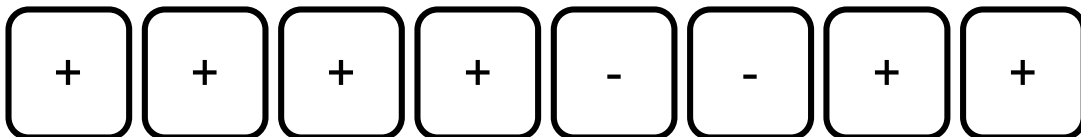
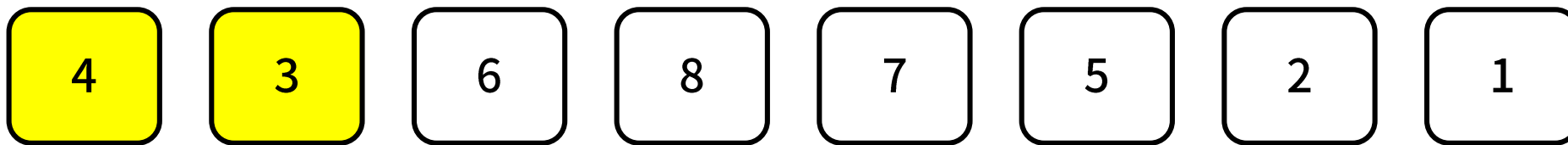
I 문제 풀이 핵심 아이디어

1. 스택에 원소를 삽입할 때는, 단순히 특정 수에 도달할 때까지 삽입하면 됩니다.
2. 스택에서 원소를 연달아 빼낼 때 내림차순을 유지할 수 있는지 확인합니다.



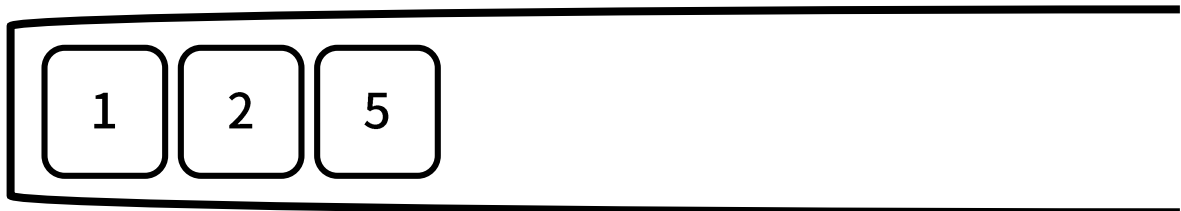
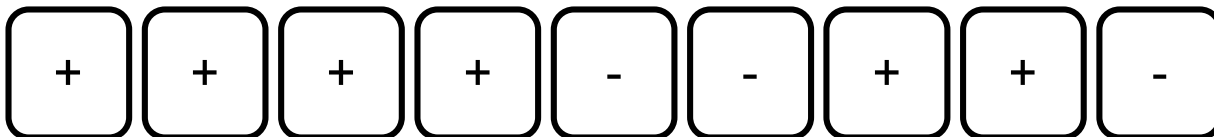
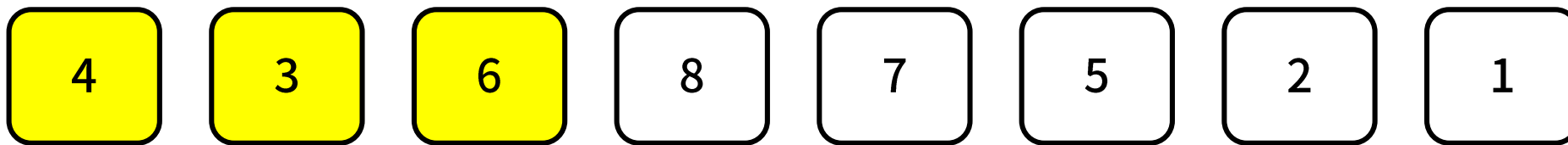
I 문제 풀이 핵심 아이디어

1. 스택에 원소를 삽입할 때는, 단순히 특정 수에 도달할 때까지 삽입하면 됩니다.
2. 스택에서 원소를 연달아 빼낼 때 내림차순을 유지할 수 있는지 확인합니다.



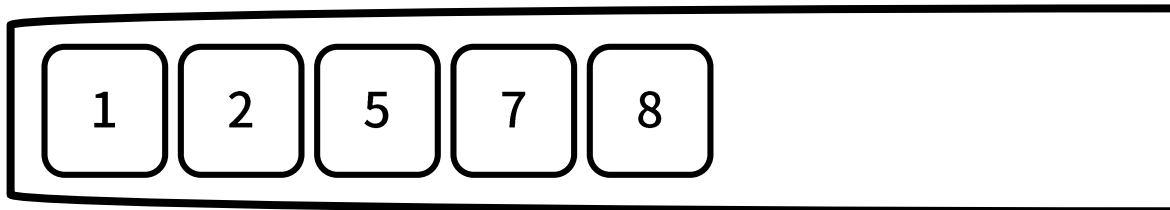
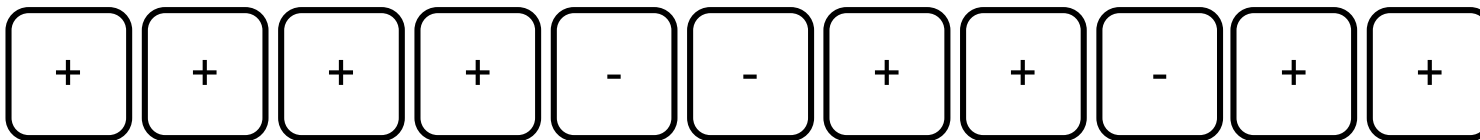
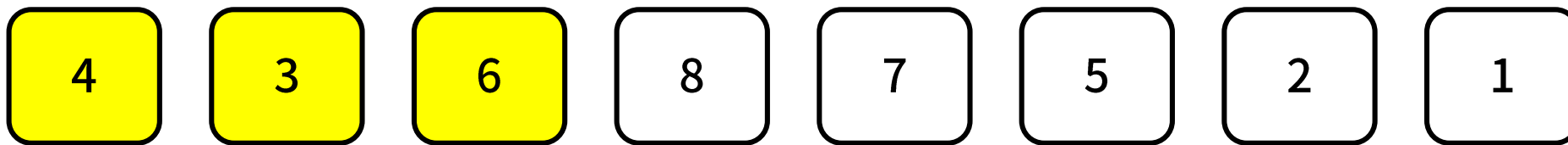
I 문제 풀이 핵심 아이디어

1. 스택에 원소를 삽입할 때는, 단순히 특정 수에 도달할 때까지 삽입하면 됩니다.
2. 스택에서 원소를 연달아 빼낼 때 내림차순을 유지할 수 있는지 확인합니다.



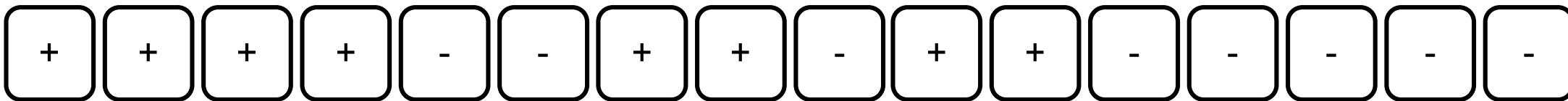
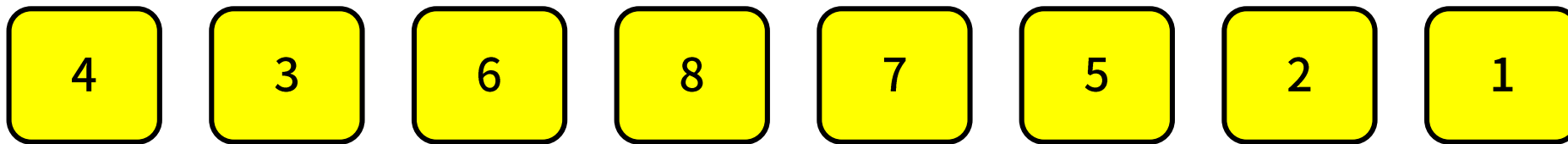
I 문제 풀이 핵심 아이디어

1. 스택에 원소를 삽입할 때는, 단순히 특정 수에 도달할 때까지 삽입하면 됩니다.
2. 스택에서 원소를 연달아 빼낼 때 내림차순을 유지할 수 있는지 확인합니다.



I 문제 풀이 핵심 아이디어

1. 스택에 원소를 삽입할 때는, 단순히 특정 수에 도달할 때까지 삽입하면 됩니다.
2. 스택에서 원소를 연달아 빼낼 때 내림차순을 유지할 수 있는지 확인합니다.



| 소스코드

```
n = int(input())

count = 1
stack = []
result = []

for i in range(1, n + 1): # 데이터 개수만큼 반복
    data = int(input())
    while count <= data: # 입력 받은 데이터에 도달할 때까지 삽입
        stack.append(count)
        count += 1
        result.append('+')
    if stack[-1] == data: # 스택의 최상위 원소가 데이터와 같을 때 출력
        stack.pop()
        result.append('-')
    else: # 불가능한 경우
        print('NO')
        exit(0)

print('\n'.join(result)) # 가능한 경우
```

I 혼자 힘으로 풀어 보기

문제 제목: 프린터 큐

문제 난이도: 하(Easy)

문제 유형: 큐, 구현, 그리디

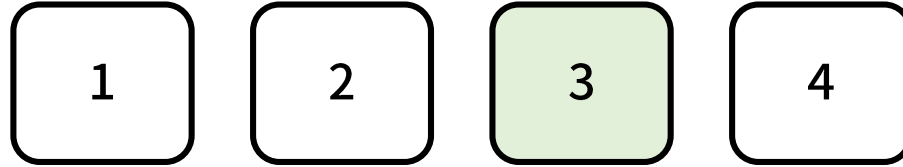
추천 풀이 시간: 25분

I 문제 풀이 핵심 아이디어

1. 데이터의 개수($N \leq 100$)가 많지 않으므로, 단순히 문제에서 요구하는 대로 구현합니다.
2. 현재 리스트에서 가장 큰 수가 앞에 올 때까지 회전시킨 뒤에 추출합니다.
3. 가장 큰 수가 문서 M에 해당하면서 가장 앞에 있을 때 프로그램을 종료합니다.

I 문제 풀이 핵심 아이디어

index = 2
count = 0

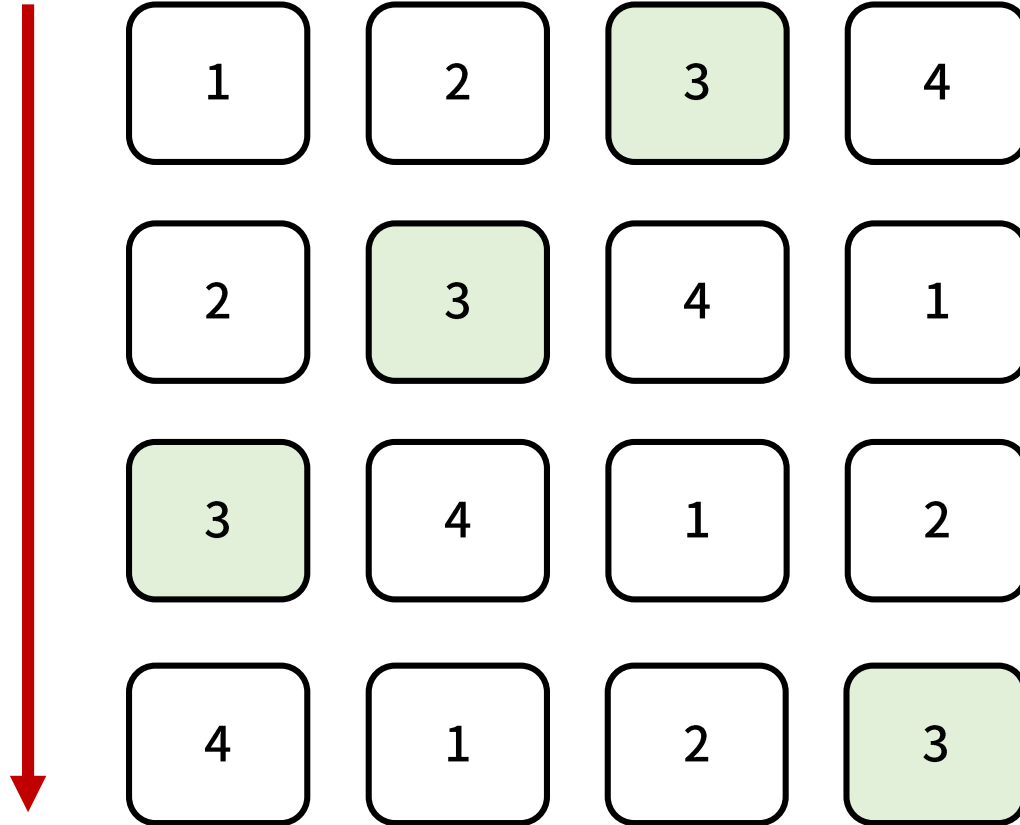


I 문제 풀이 핵심 아이디어

index = 2

count = 0

가장 큰 수: 4

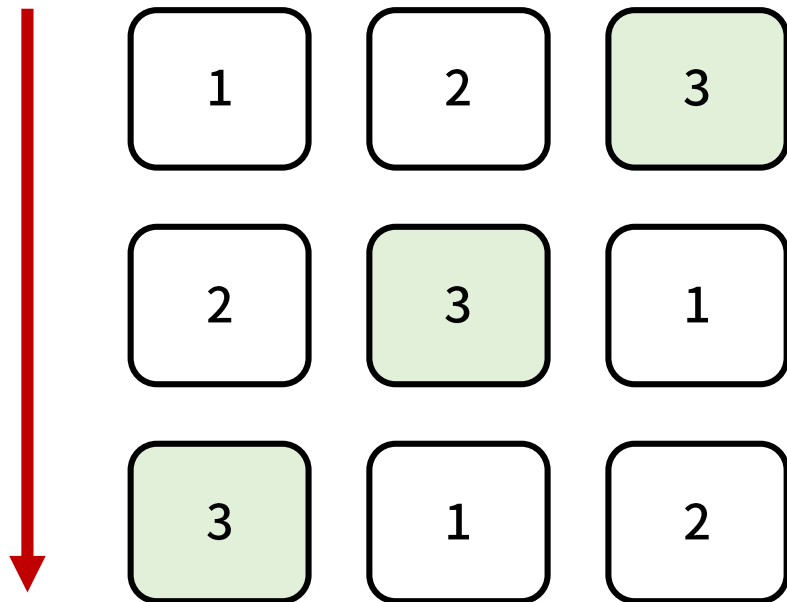


I 문제 풀이 핵심 아이디어

index = 2

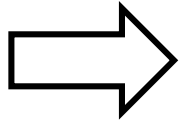
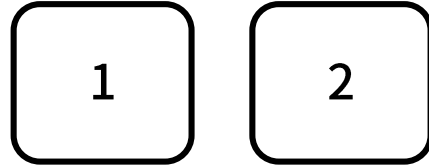
count = 1

가장 큰 수: 3



I 문제 풀이 핵심 아이디어

index = 2
count = 2



목표로 하는 문서 M이 2번째로 추출되었으므로
정답은 2입니다.

| 소스코드

```
test_case = int(input())

for _ in range(test_case):
    n, m = list(map(int, input().split(' ')))
    queue = list(map(int, input().split(' ')))
    queue = [(i, idx) for idx, i in enumerate(queue)]

    count = 0
    while True:
        if queue[0][0] == max(queue, key=lambda x: x[0])[0]:
            count += 1
            if queue[0][1] == m:
                print(count)
                break
            else:
                queue.pop(0)
        else:
            queue.append(queue.pop(0))
```

I 혼자 힘으로 풀어 보기

문제 제목: 키로거

문제 난이도: 중(Medium)

문제 유형: 스택, 구현, 그리디

추천 풀이 시간: 40분

I 문제 풀이 핵심 아이디어

1. 문자열 크기가 최대 1,000,000이므로 시뮬레이션 방식으로는 문제를 해결할 수 없습니다.
2. 스택을 활용하여 선형시간 문제를 해결할 수 있는 알고리즘을 설계합니다.

I 문제 풀이 핵심 아이디어

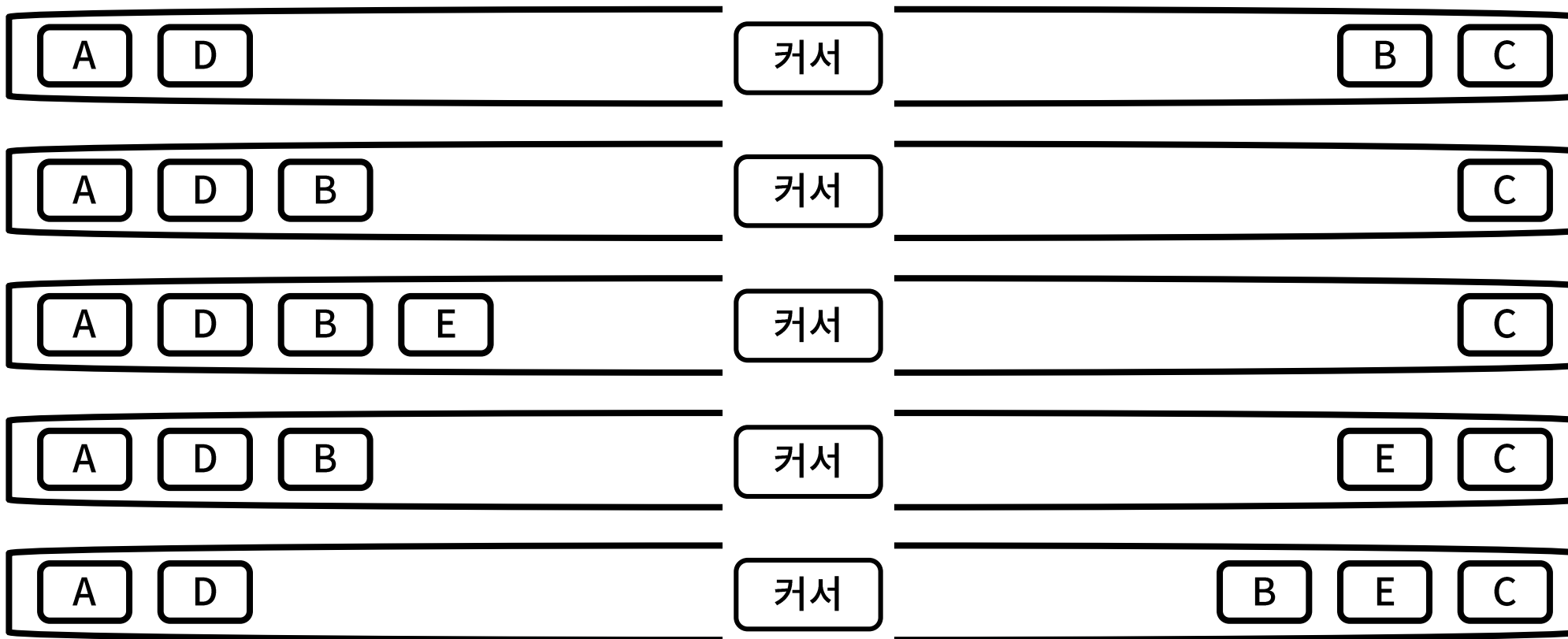
- 1) 스택 두 개를 만들고, 스택 두 개의 중간 지점을 커서(Cursor)로 간주합니다.
- 2) 문자 입력: 왼쪽 스택에 원소를 삽입합니다.
- 3) - 입력: 왼쪽 스택에서 원소를 삭제합니다.
- 4) < 입력: 왼쪽 스택에서 오른쪽 스택으로 원소를 이동합니다.
- 5) > 입력: 오른쪽 스택에서 왼쪽 스택으로 원소를 이동합니다.



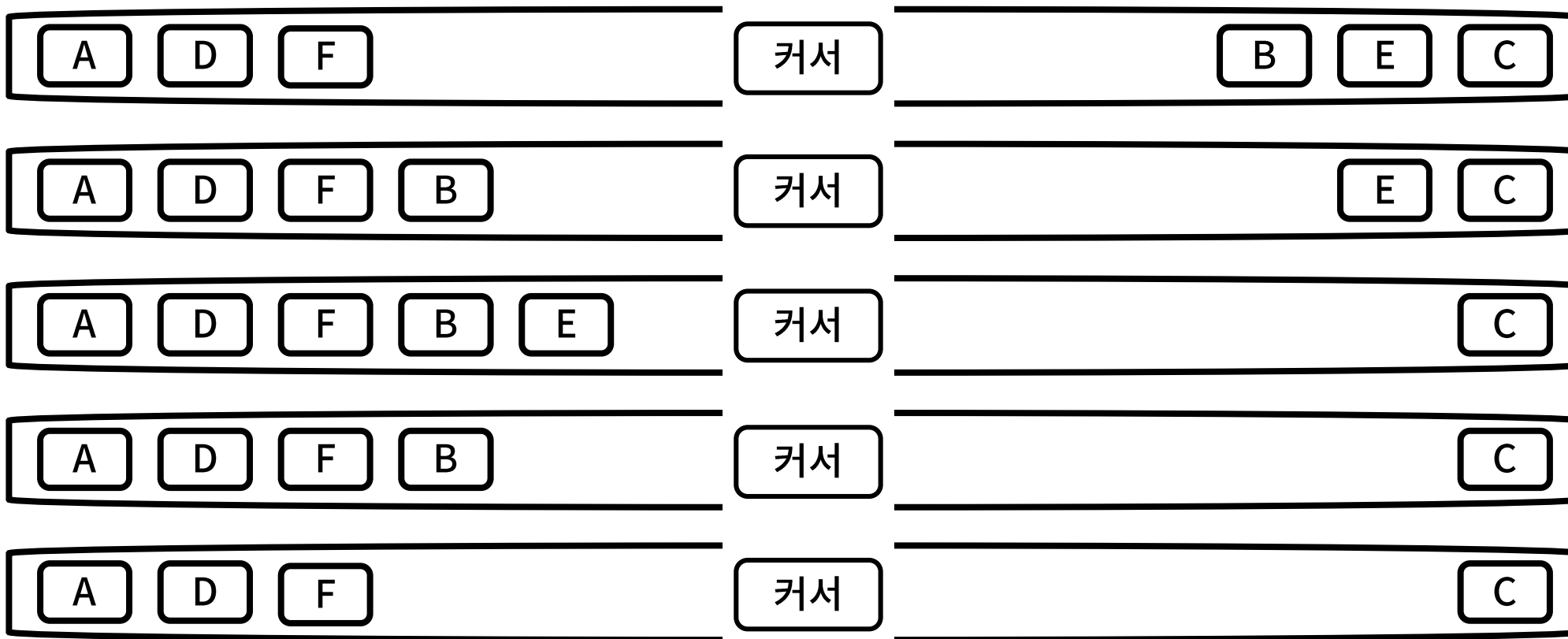
I 문제 풀이 핵심 아이디어

문자열: ABC<<D>E<<F>>---

I 문제 풀이 핵심 아이디어

문자열: ABC<<D>E<<F>>---

I 문제 풀이 핵심 아이디어

문자열: ABC<<D>E<<F>>---

| 소스코드

```
test_case = int(input())

for _ in range(test_case):
    left_stack = []
    right_stack = []
    data = input()
    for i in data:
        if i == '-':
            if left_stack:
                left_stack.pop()
        elif i == '<':
            if left_stack:
                right_stack.append(left_stack.pop())
        elif i == '>':
            if right_stack:
                left_stack.append(right_stack.pop())
        else:
            left_stack.append(i)
    left_stack.extend(reversed(right_stack))
    print(''.join(left_stack))
```