

크레인 인형뽑기 게임

1:△ 2:◇ 3:□ 4:✱ 5:♡

board 7(9x9)

| | | | | |
|-----|-----|-----|-----|-----|
| | | | | |
| | | 3:✱ | | 2:△ |
| | 4:✱ | ♡ | | 4:✱ |
| 4:✱ | ◇ | ✱ | 3:✱ | ◇ |
| 4:✱ | ♡ | △ | □ | △ |
| 1 | 2 | 3 | 4 | 5 |

basket

| |
|---------|
| 4:△ |
| 3:△ |
| 4:□ |
| 3:□ |
| 4:✱ |
| count=2 |

| |
|---------|
| |
| |
| 5:□ |
| 3:□ |
| 4:✱ |
| count=4 |

→ 스택 사용해서
현재 집은 인형과
바구니의 재입력 인형이
같은지 pop한 count이 2이하

moves 크기/만큼 움직여서 basket에 쌓음

moves[0]=1

1에서 0이 아닌 것중에 제일 위에 쌓인 인형 basket에 넣기

moves[1]=5

보드판 만들기

board[j][moves[i]-1] == 0이면 j++

" != 0이면 0으로 바꾸고 basket에 추가.

| | | | | | |
|---|---|---|---|---|---|
| 0 | | | | | |
| 1 | | | | | |
| 2 | | | | | |
| 3 | | | | | |
| 4 | | | | | |
| | 0 | 1 | 2 | 3 | 4 |

move 1-1 2-1 3-1 4-1 5-1

1. 필로 변수 선언 (basket, count)

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

```
int count = 0;
```

2. 로직 실행

```
for(int i=0; i < moves.length; i++){  
    for(int j=0; j < board.length; j++){  
        int doll = board[j][moves[i]-1];
```

3. 만약에 0이면 밑으로 이동해야 함.

```
if(doll == 0){  
    continue;
```

4. 0이 아니면 로직 진행

```
} else {
```

5. 우선 인형을 뿔았으므로 0으로 바꿔줌.

```
board[j][moves[i]-1] = 0;
```

6. basket 안에 인형이 있고 제일 위위 인형과 지금 뽑은 인형 같은 인형인지 비교해서 같으면 pop하고 count +2

```
if(!basket.isEmpty() && doll == basket.peek()){
```

```
    basket.pop();
```

```
    count += 2;
```

```
} else {
```

```
    basket.push(doll);
```

```
}
```

```
}
```

```
}
```

```
return count;
```