

카드 짝 맞추기

Python3

문제 설명

게임 개발자인 베로니는 개발 연습을 위해 다음과 같은 간단한 카드 짝맞추기 보드 게임을 개발해 보려고 합니다.

게임이 시작되면 화면에는 카드 16장이 뒷면을 위로하여 4 x 4 크기의 격자 형태로 표시되어 있습니다. 각 카드의 앞면에는 카카오프렌즈 캐릭터 그림이 그려져 있으며, 8가지의 캐릭터 그림이 그려진 카드가 각기 2장씩 화면에 무작위로 배치되어 있습니다.

유저가 카드를 2장 선택하여 앞면으로 뒤집었을 때 같은 그림이 그려진 카드면 해당 카드는 게임 화면에서 사라지며, 같은 그림이 아니라면 원래 상태로 뒷면이 보이도록 뒤집힙니다. 이와 같은 방법으로 모든 카드를 화면에서 사라지게 하면 게임이 종료됩니다.







게임에서 카드를 선택하는 방법은 다음과 같습니다.

- 카드는 커서를 이용해서 선택할 수 있습니다.
 - 커서는 4 x 4 화면에서 유저가 선택한 현재 위치를 표시하는 "굵고 빨간 테두리 상자"를 의미합니다.
- 커서는 [Ctrl] 키와 방향키에 의해 이동되며 키 조작법은 다음과 같습니다.
 - 방향키 ←, ↑, ↓, → 중 하나를 누르면, 커서가 누른 키 방향으로 1칸 이동합니다.
 - [Ctrl] 키를 누른 상태에서 방향키 ←, ↑, ↓, → 중 하나를 누르면, 누른 키 방향에 있는 가장 가까운 카드로 한번에 이동합니다.
 - 만약, 해당 방향에 카드가 하나도 없다면 그 방향의 가장 마지막 칸으로 이동합니다.
 - 만약, 누른 키 방향으로 이동 가능한 카드 또는 빈 공간이 없어 이동할 수 없다면 커서는 움직이지 않습니다.
- 커서가 위치한 카드를 뒤집기 위해서는 [Enter] 키를 입력합니다.
 - [Enter] 키를 입력해서 카드를 뒤집었을 때
 - 앞면이 보이는 카드가 1장 뿐이라면 그림을 맞출 수 없으므로 두번째 카드를 뒤집을 때 까지 앞면을 유지합니다.
 - 앞면이 보이는 카드가 2장이 된 경우, 두개의 카드에 그려진 그림이 같으면 해당 카드들이 화면에서 사라지며, 그림이 다르다면 두 카드 모두 뒷면이 보이도록 다시 뒤집힙니다.

"베로니"는 게임 진행 중 카드의 짝을 맞춰 몇 장 제거된 상태에서 카드 앞면의 그림을 알고 있다면, 남은 카드를 모두 제거하는데 필요한 키 조작 횟수의 최솟값을 구해 보려고 합니다. 키 조작 횟수는 방향키와 [Enter] 키를 누르는 동작을 각각 조작 횟수 1로 계산하며, [Ctrl] 키와 방향키를 함께 누르는 동작 또한 조작 횟수 1로 계산합니다.

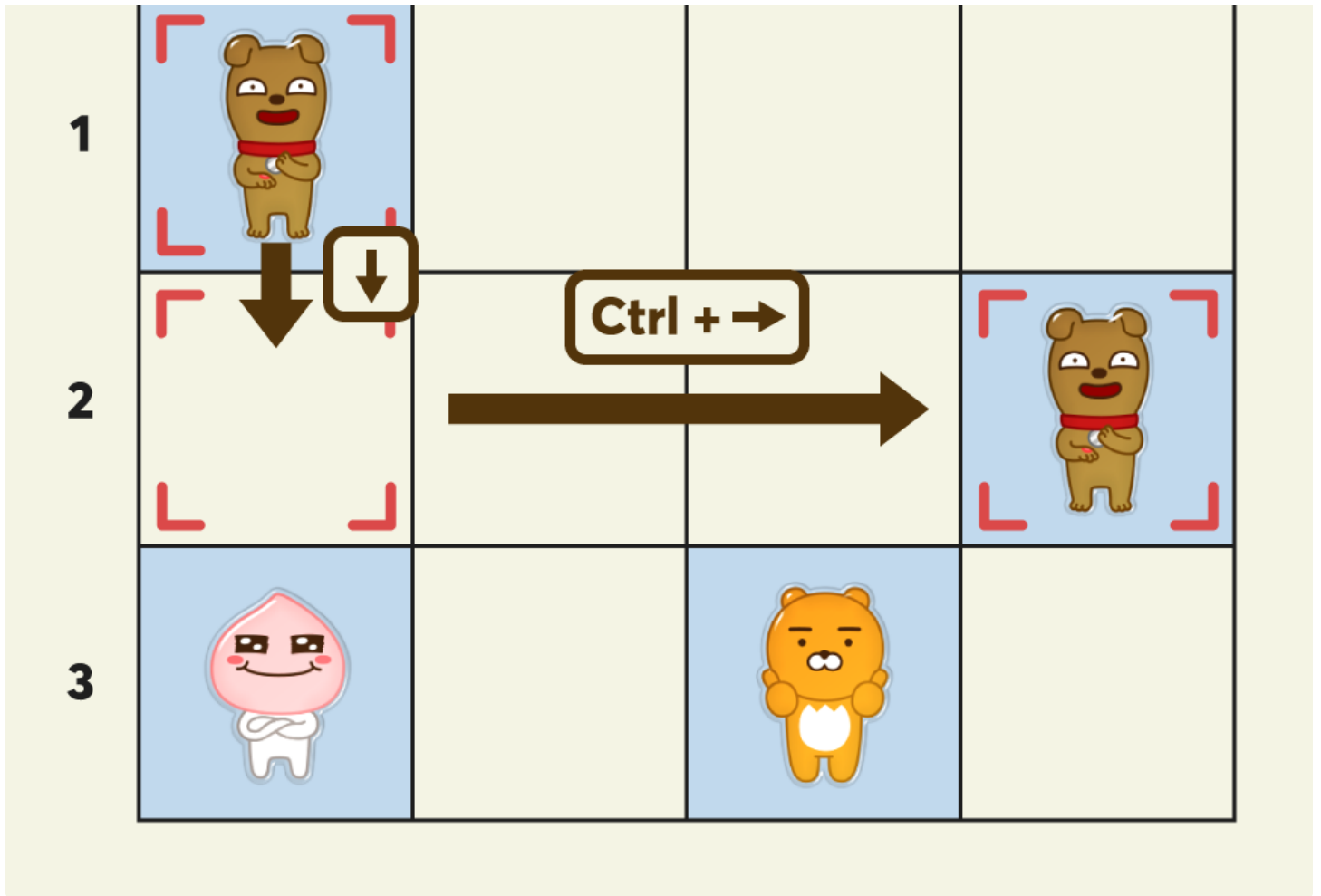
다음은 카드가 몇 장 제거된 상태의 게임 화면에서 커서를 이동하는 예시입니다.

아래 그림에서 빈 칸은 이미 카드가 제거되어 없어진 칸을 의미하며, 그림이 그려진 칸은 카드 앞 면에 그려진 그림을 나타냅니다.

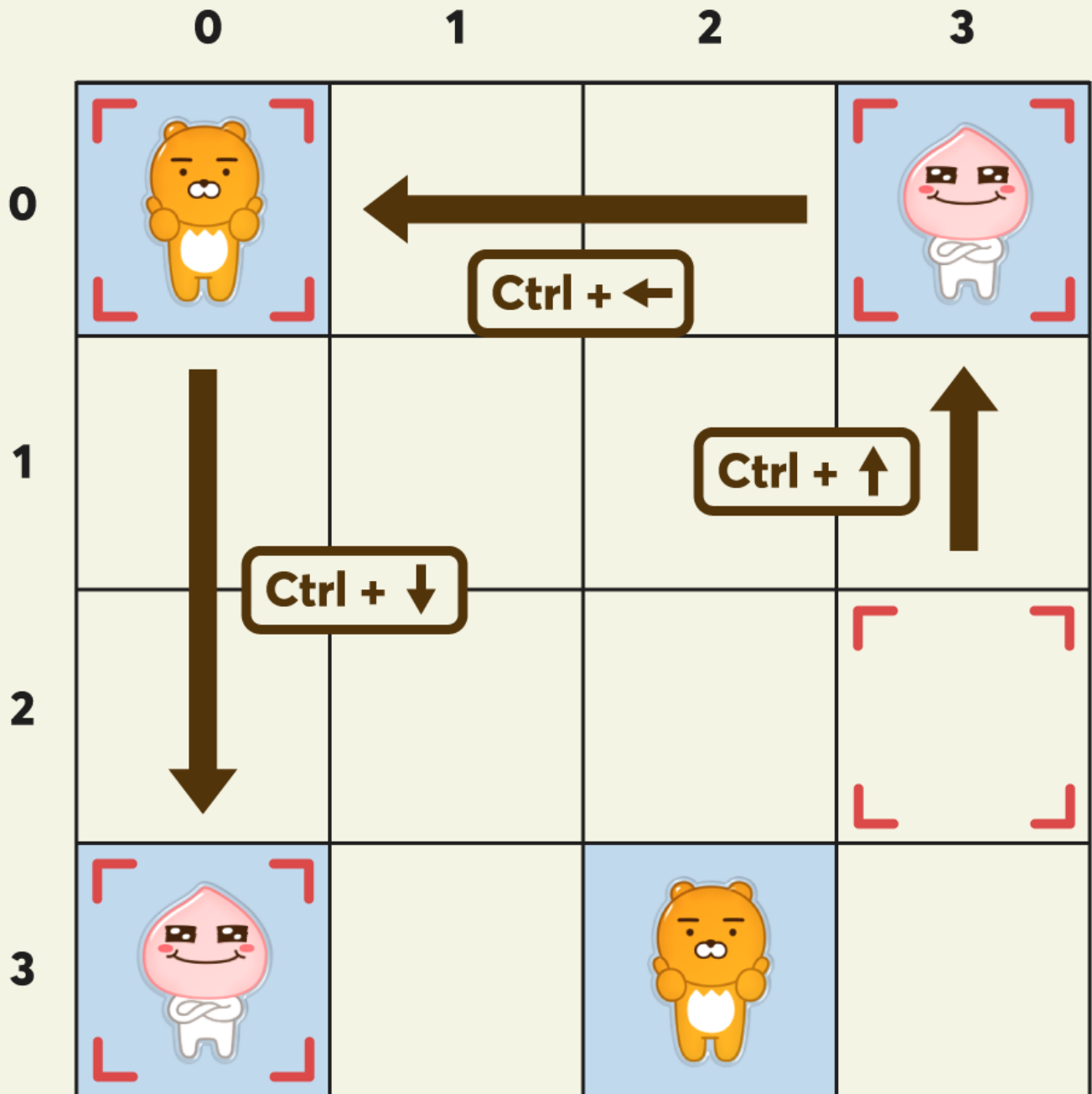
	0	1	2	3
0				
1				
2				
3				

예시에서 커서는 두번째 행, 첫번째 열 위치에서 시작하였습니다.

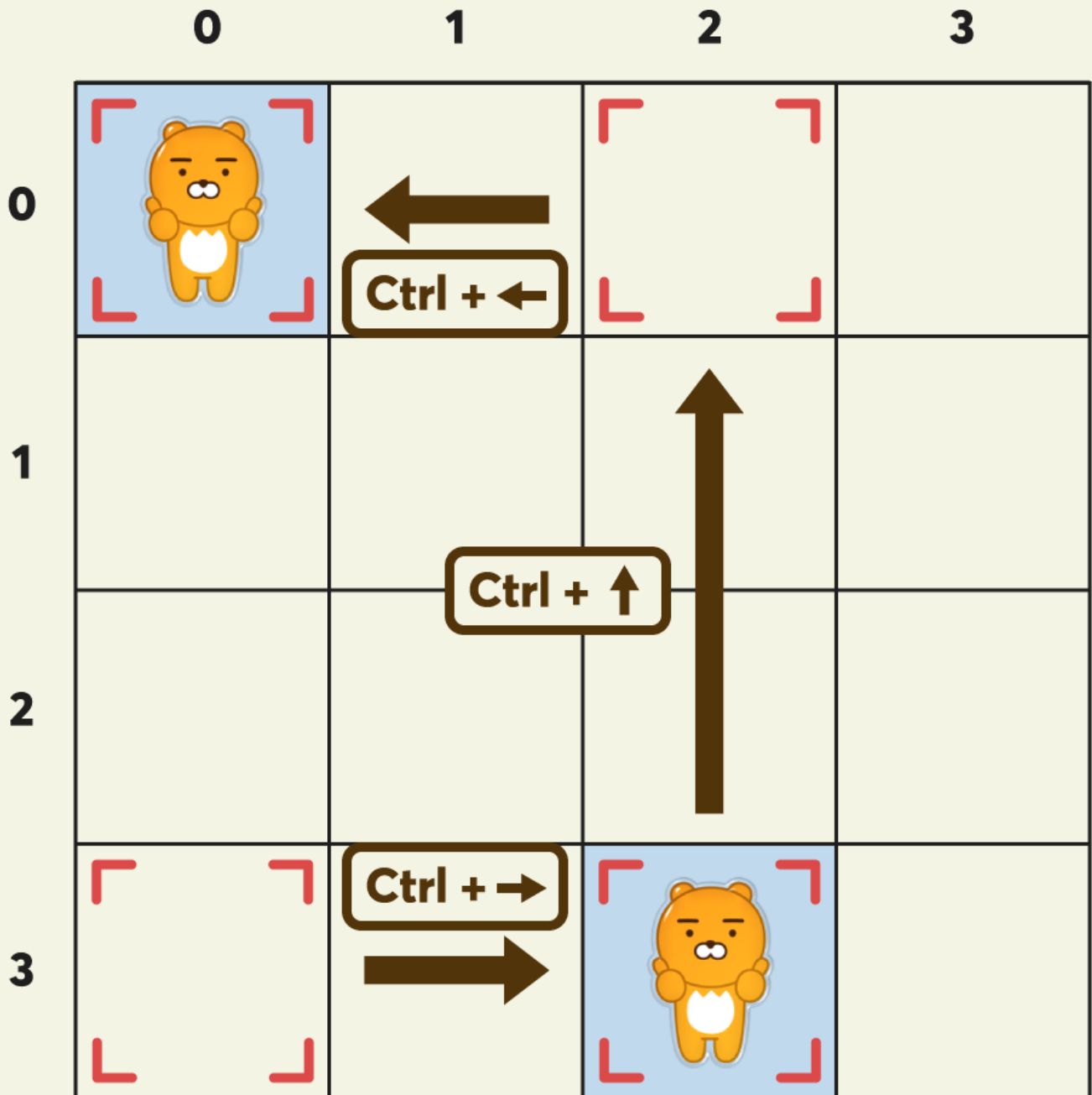
	0	1	2	3
0				



[Enter] 입력, ↓ 이동, [Ctrl]+→ 이동, [Enter] 입력 = 키 조작 4회



[Ctrl]+↑ 이동, [Enter] 입력, [Ctrl]+← 이동, [Ctrl]+↓ 이동, [Enter] 입력 = 키 조작 5회



[Ctrl]+→ 이동, [Enter] 입력, [Ctrl]+↑ 이동, [Ctrl]+← 이동, [Enter] 입력 = 키 조작 5회

위와 같은 방법으로 커서를 이동하여 카드를 선택하고 그림을 맞추어 카드를 모두 제거하기 위해서는 총 14번(방향 이동 8번, [Enter] 키 입력 6번)의 키 조작 횟수가 필요합니다.

[문제]

현재 카드가 놓인 상태를 나타내는 2차원 배열 board와 커서의 처음 위치 r, c 가 매개변수로 주어질 때, 모든 카드를 제거하기 위한 키 조작 횟수의 최솟값을 return 하도록 solution 함수를 완성해 주세요.

[제한사항]

- board는 4×4 크기의 2차원 배열입니다.
- board 배열의 각 원소는 0 이상 6 이하인 자연수입니다.
 - 0은 카드가 제거된 빈 칸을 나타냅니다.
 - 1 부터 6까지의 자연수는 2개씩 들어있으며 같은 숫자는 같은 그림의 카드를 의미합니다.
 - 뒤집을 카드가 없는 경우(board의 모든 원소가 0인 경우)는 입력으로 주어지지 않습니다.
- r은 커서의 최초 세로(행) 위치를 의미합니다.
- c는 커서의 최초 가로(열) 위치를 의미합니다.
- r과 c는 0 이상 3 이하인 정수입니다.
- 게임 화면의 좌측 상단이 (0, 0), 우측 하단이 (3, 3) 입니다.

[입출력 예]

board	r	c	result
[[1,0,0,3],[2,0,0,0],[0,0,0,2],[3,0,1,0]]	1	0	14
[[3,0,0,2],[0,0,1,0],[0,1,0,0],[2,0,0,3]]	0	1	16

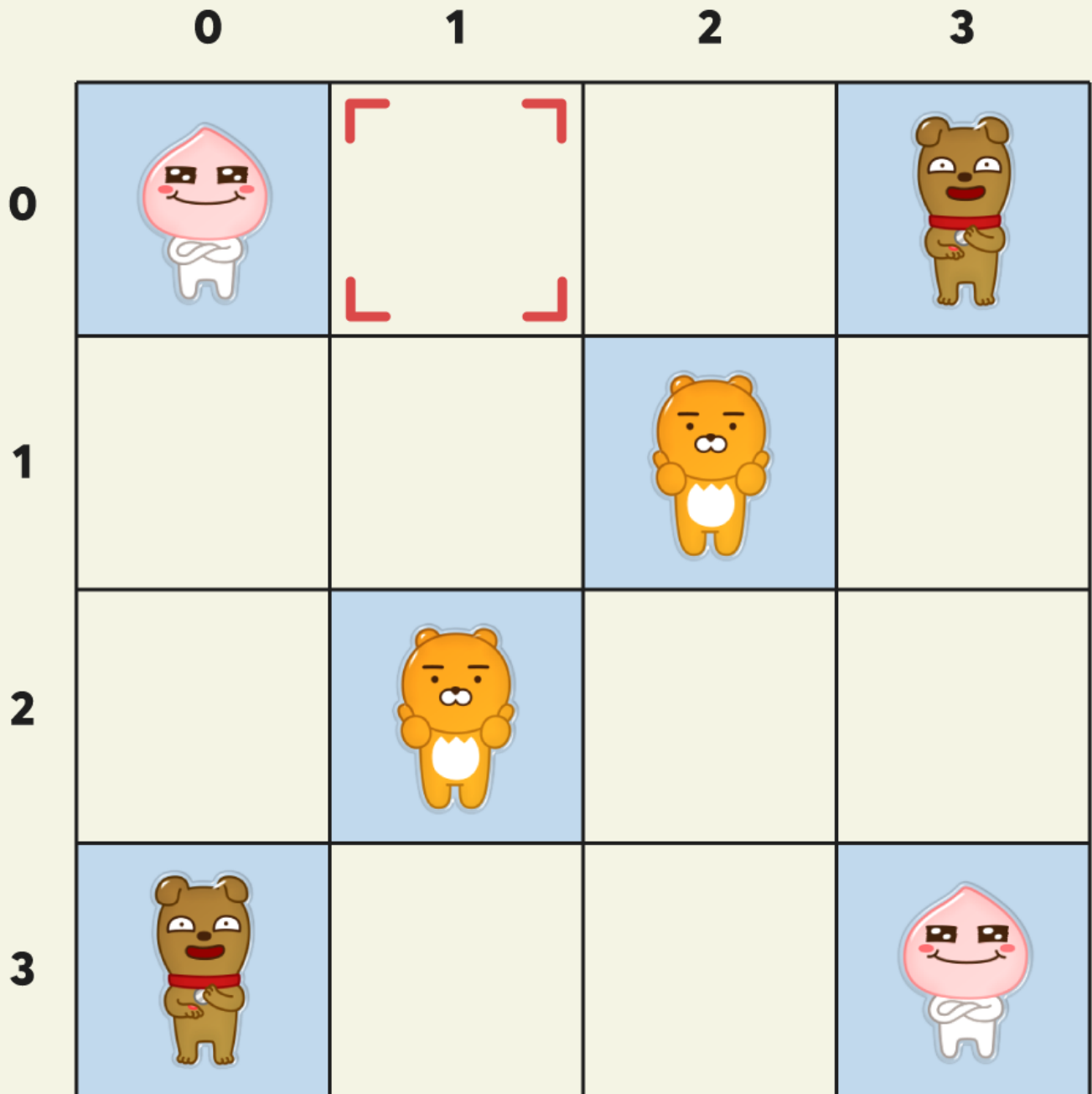
입출력 예에 대한 설명

입출력 예 #1

문제의 예시와 같습니다.

입출력 예 #2

입력으로 주어진 게임 화면은 아래 그림과 같습니다.



위 게임 화면에서 모든 카드를 제거하기 위한 키 조작 횟수의 최솟값은 16번 입니다.

solution.py

```
1 def solution(board, r, c):
2     answer = 0
3     return answer
```

실행 결과

실행 결과가 여기에 표시됩니다.

- 초기화
- 코드 실행
- 제출 후 채점하기