

## 문제정의

주어진 블록을 정사각형 안에 배치하여 겹쳐지는 부분이 없는 경우가 있는지 확인하는 문제

## 문제해결

### 초기 선언

블록의 모양을 저장하는 3차원의 배열 `polys`를 선언한다. 조각의 최대 개수는 5이고 최대 `h`와 `w`는 4이므로  $5 \times 4 \times 4$ 의 크기를 가진다. 정사각형의 한 변의 최대 크기는 8이다. (블록의 최대 개수는 80이므로) 따라서 정사각형을 나타내는 `frame`의 크기는  $8 \times 8$ 로 선언한다. 각 블록의 `h`와 `w`를 저장하는 배열을 각각 `h_arr`와 `w_arr`로 선언한다. 크기는 각각 5로 한다.

### 예외처리

`n`개의 블록을 입력받아 `polys`, `h_arr`, `w_arr`에 각각의 블록의 모양 높이, 너비를 저장한다. 이때 입력받은 1의 개수를 같이 센다. 이 개수가 1~8의 제공수가 아니면 No solution possible을 출력한다.

### 정사각형 맞추기

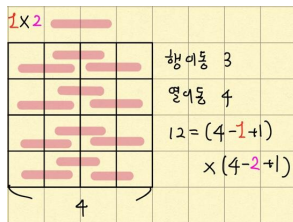
만약 블록의 총 개수가 제공수라면 정사각형 한 변의 길이 `m`을  $\sqrt{1}$ 의 총 개수로 지정한다. 그리고 각 조각에 대해 아래의 과정을 반복한다. `block`은 현재 `block`의 색인이며, 초기에 0으로 지정한다.

**블록놓기** : 블록을 놓을 수 있는 횟수는  $(m-w[\text{block}]+1) \times (m-h[\text{block}]+1)$ 이다 `frame`의 (0,0)부터 블록을 놓는다(<그림1>참고). 이동횟수가 `i`번째일때 `frame`의 맨 왼쪽(`x,y`)는  $x = i/(m-w[\text{block}]-1)$ ,  $y = i\%(m-w[\text{block}]+1)$ 이다.

**겹침 검사**: 만약 블록을 놓아야 하는 자리에 1이 존재한다면 다른 블록이 이미 있다는 뜻이므로 겹쳐진 블록을 지우고 다음 자리에 블록을 옮긴다. 이 블록이 마지막 블록이고 놓을 수 있는 마지막 위치에 블록이 놓아져도 겹침이 발생하면 No solution possible을 출력한다.

마지막 위치에 와도 블록이 겹치지만 마지막 블록이 아닌 경우, `block`을 1줄이고 현재 겹쳐진 블록을 지운다음 그 전의 블록을 다시 이동시키며 블록놓기와 겹침 검사를 시행한다. 블록이 겹치지 않는 경우이면, 놓아야 할 다음블록이 있는지 확인하고, 다음블록이 존재하지 않으면 정사각형이 만들어지는 경우이므로 배열을 출력한다. 다음 블록이 존재하는 경우 `block`에 1을 더하고 `block<n`인 경우 블록놓기로 넘어가서 다시 겹침검사로 넘어간다.

### 반복횟수



### <그림1>

$1 \times 2$ 의 크기를 가지는 블록이 존재할 때, 행으로는 3번이동하고 열은 4번을 이동한다. 총 이동 횟수를 정사각형의 크기와 가로 세로의 크기로 나타내면 (정사각형 크기 - 가로크기 + 1) \* (정사각형 크기 - 세로크기 + 1)이다

### 시간복잡도

정사각형의 한 변이 `m`이라 하면 블록의 평균 변은  $(m/\sqrt{n})$ 이 된다. 이때 블록이 이동하는 횟수는  $(m-m/\sqrt{n}+1)^2$ 이다. 정사각형의 크기에 대해 시간복잡도를 계산하면  $O(m^2)$ 이 최종 시간복잡도이다.

### 공간복잡도

각 입력의 최대 크기로 배열들의 크기를 잡았으므로 공간복잡도는  $O(1)$ 이다.