**1. 문제 설명**

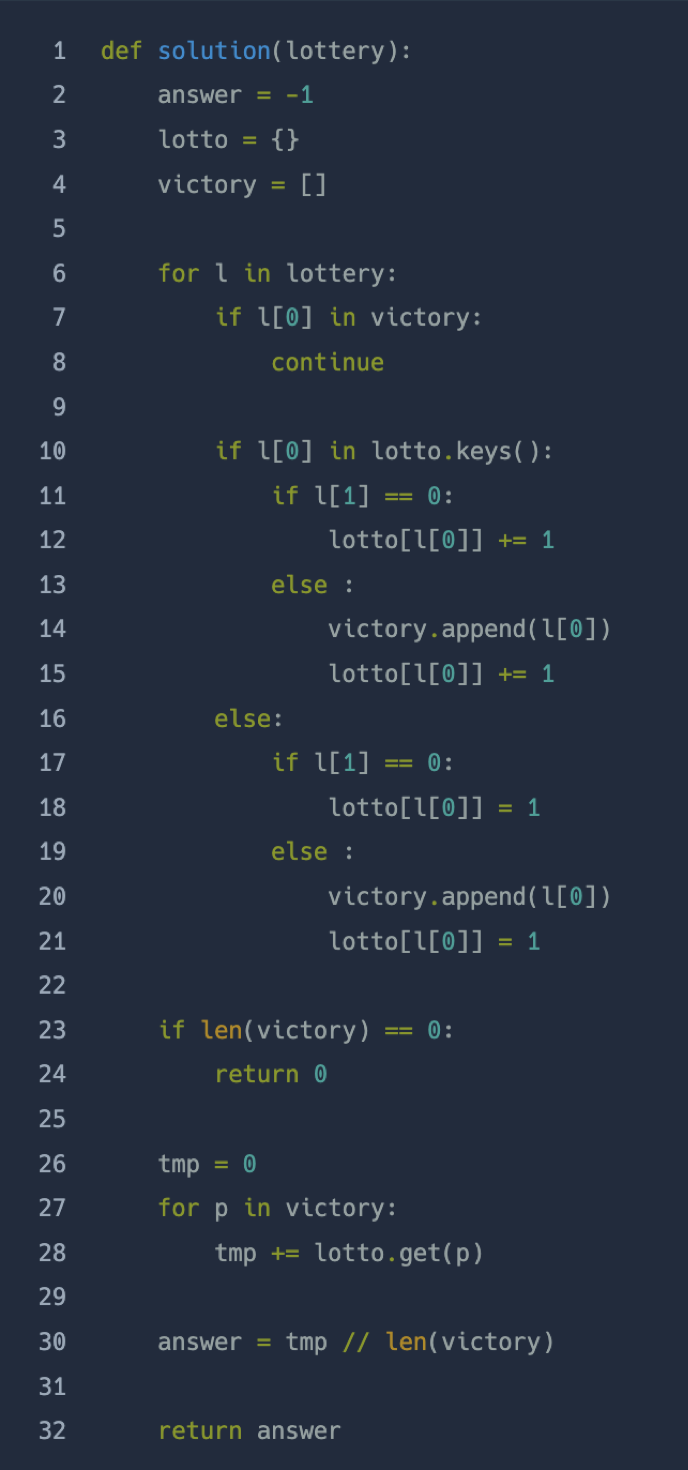
처음 당첨되기까지 평균 몇 번 복권을 구매했는지 구하세요.

* 꽝(0), 당첨(1) 두가지 경우
* USER\_ID는 1이상 1000이하 자연수
* 한번 이상 당첨된 사람 대상 조사
* 한번에 한 명씩 한 장씩 구매
* [유저 ID, 당첨 여부]
* 최초 당첨 이후에도 계속 복권 구매 가능
* 소수점 이하 버리고 정수만 return
* 한 명도 당첨되지 않은 경우에는 0 return

[Test Case]

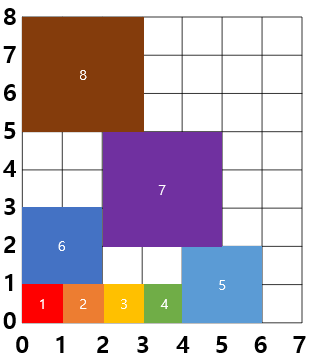
|  |  |
| --- | --- |
| lottery | result |
| [[1, 0], [35, 0], [1, 0], [100, 1], [35, 1], [100, 1], [35, 0], [1, 1], [1, 1]] | 2 |

[제출 코드]



**3. 문제설명**

n x m 크기 탁자 위에 다양한 크기의 정사각형을 다음 규칙에 따라 놓으려 합니다.



크기가 작은 정사각형부터 놓습니다.

1-1. 정사각형은 한 번에 하나씩 놓습니다.

탁자 위에 놓을 정사각형의 위치는 다음과 같이 정합니다.

2-1. 정사각형 밑변의 y축 값이 가장 작은 위치로 배치합니다.

2-2. 2-1에서 정한 y축 상에 놓을 수 있는 공간이 여러 개인 경우, 정사각형 좌변의 x축 값이 가장 작은 위치로 배치합니다.

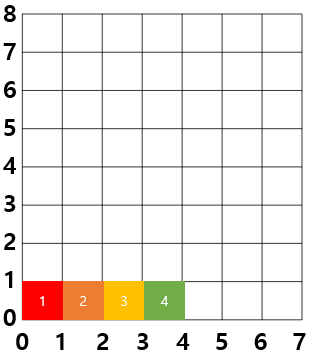
2-3. x, y값은 정수여야 합니다.

정사각형이 다른 정사각형과 겹치거나, 탁자를 벗어나지 않아야 합니다.

정사각형의 각 변이 x, y축과 평행해야 합니다.

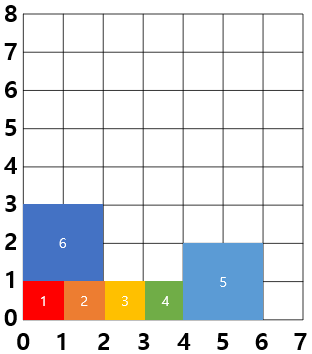
다음은 7 x 8 크기 탁자 위에 1 x 1 정사각형 4개, 2 x 2 정사각형 2개, 3 x 3 정사각형 2개를 규칙에 맞게 놓는 예시입니다.

1. 가장 작은 1 x 1 크기 정사각형을 조건에 맞게 차례대로 놓으면 다음 그림과 같습니다.

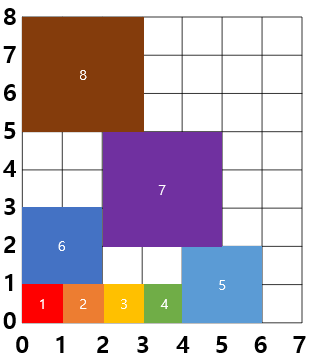


1. 다음으로 2 x 2 크기 정사각형을 조건에 맞게 차례대로 놓으면 다음 그림과 같습니다.

그림과 같이 5번 정사각형을 놓은 후 6번 정사각형을 놓으면 됩니다.



1. 마지막으로 3 x 3 크기 정사각형을 조건에 맞게 차례대로 놓으면 다음 그림과 같습니다.



탁자의 X축(가로) 길이 n, Y축(세로) 길이 m, 탁자 위에 놓을 정사각형의 변의 길이와 개수가 담긴 2차원 정수 배열 rectangle이 매개변수로 주어집니다. 정사각형을 조건에 맞게 탁자 위에 놓은 후, 탁자 위에 놓인 정사각형의 좌측 하단 꼭짓점 좌표와 한 변의 길이를 [x축 좌표, y축 좌표, 한 변의 길이] 형태로 배열에 담아 return 하도록 solution 함수를 완성해주세요. 정답은 탁자 위에 놓은 순서대로 담겨있으면 됩니다.

* 제한사항
  + 1 ≤ n ≤ 20
  + 1 ≤ m ≤ 20
  + 1 ≤ rectangle의 행(세로) 길이 ≤ 20
  + rectangle의 열(가로) 길이 = 2
  + rectangle의 각 행은 [정사각형의 한 변의 길이, 정사각형 개수] 형태입니다.
  + 1 ≤ 정사각형의 한 변의 길이 ≤ 10
  + 1 ≤ 정사각형 개수 ≤ 10
  + 한 변의 길이가 같은 정사각형에 대한 정보가 서로 다른 행에 주어지는 경우는 없습니다.
  + 전체 정사각형의 개수 ≤ 40
  + 탁자 위에 모든 정사각형을 놓을 수 없는 경우, 남는 정사각형은 무시합니다.
  + 즉, 정답 배열에는 탁자 위에 놓인 정사각형의 정보만 담으면 됩니다.
  + 탁자 위에 정사각형을 하나 이상 놓을 수 있는 경우만 입력으로 주어집니다.

입출력 예

|  |  |  |  |
| --- | --- | --- | --- |
| **n** | **m** | **rectangle** | **result** |
| 7 | 8 | [[2,2], [1,4], [3,2]] | [[0,0,1],[1,0,1],[2,0,1].[3,0,1],[4,0,2],[0,1,2],[2,2,3],[0,5,3]] |

3. 로봇청소기 – 그래프 문제

왼쪽 최상단에서 오른쪽 최하단으로 이동하는 문제

오른쪽이나 아래쪽으로는 +1

대각선으로는 +2 만큼 증가

0,0 에서 시작해서 n-1,n-1로 이동할 떄

각 원소들의 점수를 value로 갖는 리스트 반환

Input: n,n 리스트

Output: 리스트

다음과 같이 이동

1->9 까지 이동

|  |  |  |
| --- | --- | --- |
| 1 | 2 | 6 |
| 3 | 5 | 7 |
| 4 | 8 | 9 |