Problem

On a `N*N` grid, there was `N*N` cats, each cat sits on a cell of the grid. For example, when N = 2, we have the following grid (each character C denotes a cat):

```
+---+---+
| c | c |
+---+---+
| c | c |
+---+---+
```

Cats like to jumps around, so each minute, each cat jumps to one of the neighbouring cell. 2 cells are considered neighbours if they share 1 edge. Note that each cell has 4 edges, but the cells at the corners only have 2 neighbours, cells at the edges have 3 neighbours and the other cells have 4 neighbours.

After T minutes, what is the expected number of unoccupied cells?

> After T minutes, the board can be in a number of possible states. We're looking for the Expected Value of the unoccupied cells. Sum up the number the unoccupied cells in each possible state and divide the result by the number of possible state. See Explanation below for more details.

Constraints:

- 1 <= N <= 30
- 1 <= T <= 50

Input:

- 1st line: N and T

Output:

- 1st line: result, rounded to *exactly* 6 decimal places. Note that if you output less than 6 decimal places or more than 6 decimal places, your output will be judged as wrong.

Example:

Input

```
2 0
```

Output

```
0.000000
```

Input

```
2 1
```

Output

```
1.000000
```

Explanation

- In first test case, after 0 second, all the cats are still in their initial positions. So the expected value is 0.0.
- In the second test case, after 1 second, each cat can jump to one of the 2 neighbouring cells (note that they must jump). Below are some of the possible state of the grid after 1 minute:
    - Cat in `(1, 1)` jump to `(1, 2)`. Cat in `(1, 2)` --> `(1, 1)`. Cat in `(2, 1)` --> `(2, 2)` and cat in `(2, 2)` --> `(2, 1)`:

```
+---+---+
| c | c |
+---+---+
| c | c |
+---+---+
```

- The cat in cell `(1, 2)` and `(2, 1)` jumped to cell `(1, 1)`. Cat in `(1, 1)` and `(2, 2)` jumped to `(1, 2)`:

```
+---+---+
|cc |cc |
+---+---+
|   |   |
+---+---+
```

- There are 16 states in total, by counting the unoccupied cells and divide the result by 16, we get the answer 1.0.