Problem Statement (PerfectMemory)

You might have played the game called Memoria. In this game, there is a board consisting of **N** rows containing **M** cells each. Each of the cells has a symbol on its back. Each symbol occurs on exactly two cells on the board.

A move means turning a pair of cells *one by one* to see the symbols behind them. When the symbols differ, both of the cells are turned on their faces, thus hiding the symbols again. The player should remember the symbols. If the symbols on the backs of the turned cells coincide, the cells stay that way, i.e., don't turn back again. As soon as after some move all the cells on the board are turned (such that all the symbols are simultaneously visible), the game ends.

Manao has a perfect memory, so when he sees the symbol behind some cell, he remembers it forever. Manao is trying to finish the game in the least expected number of moves. Find the expected number of moves he will need to accomplish this.

Definition

Class:	PerfectMemory
Method:	getExpectation
Parameters:	int, int
Returns:	double
Method signature:	double getExpectation(int N, int M)

(be sure your method is public)

Notes

- The board Manao plays on is generated as follows. The same set of (**N** * **M**) / 2 symbols is used for each generation.
- The board contents are chosen uniformly among all valid **N** x **M** boards.
- The returned value must have an absolute or relative error less than 1e-9.

Constraints

- **N** will be between 1 and 50, inclusive.
- **M** will be between 1 and 50, inclusive.
- **N** * **M** will be even.

Examples

0)

```
1

2

Returns: 1.0
```

There are only two cells on the board, so the game always ends in one move.

1)

```
2

2

Returns: 2.6666666666666665
```

There are four cells. The game may flow in two possible scenarios:

1) In the first move, Manao turns two cells with equal symbols. The game ends in two moves then and the probability of such a first move is 1/3.

2) In the first move, Manao turns two cells with different symbols. Then he finishes the game in three moves and the probability of such a first move is 2/3.

The overall expected number of moves is 1/3 * 2 + 2/3 * 3 = 8/3.

2)

```
2

3

Returns: 4.333333333333334
```

3)

```
4

4

Returns: 12.392984792984793
```