

Hard Problem

Description

This is a hard problem, can you tell me the expected return value of the following hard program?

```
from random import randint

def hard(board, goal):
    cnt = 0
    while board != goal:
        c = randint(0, 1)
        x = randint(1, len(board))
        y = randint(1, len(board[0]))
        for i in range(x):
            for j in range(y):
                board[i][j] = c
            cnt += 1
    return cnt
```

The argument `board` and `goal` are 2-d binary matrices with same size. You can assume that `randint(1, r)` will return a uniform random integer in $[l, r]$, and `range(x)` is a list $[0, 1, \dots, x - 1]$.

Input

The first line contains a integer T indicating the total number of test cases. Each test case starts with a line containing two integers n, m , denoting the size of `board` and `goal`. The following n lines, each line contains m digits, denoting the binary matrix `board`. The following n lines denoting the binary matrix `goal` in same format.

- $1 \leq T \leq 100$
- $1 \leq n, m \leq 5$

You can assume that the expected return value will not exceed 10^9 .

Output

For each test case, output the expected return value of the hard program. The error within 10^{-6} would be accepted.

Sample Input

```
3
1 2
00
11
2 1
1
0
1
0
2 2
10
10
00
00
```

Sample Output

```
6.000000000
0.000000000
12.857142857
```