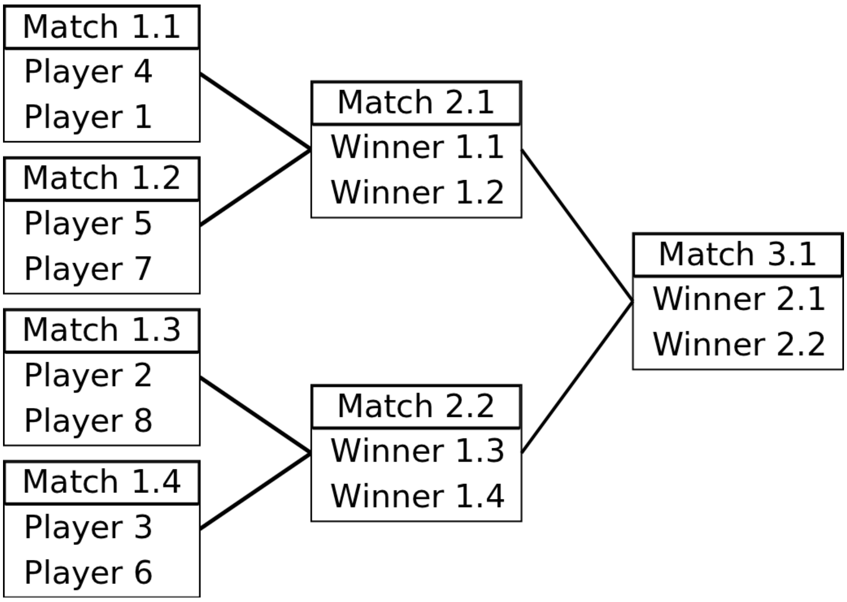
Joy of CompuTenis

Original statement in spanish at http://www.dc.uba.ar/events/icpc/download/problems/taip2011-problems.pdf

The Modern Club Association organizes every year a tournament of CompuTenis, which is a sport specially adapted to a public without any mensurable physical qualities. The rules of CompuTenis are very complex (suffices to say that they involve coding with your elbow glued to your ear), but fortunately it is not necessary to know them in detail to solve this problem. You just need to know that in a CompuTenis match two players oppose each other, and the match is won by the player that first wins **S** sets; in turn, each set is composed of several games, and to win a set a player must win at least **J** games, with a difference of at least **D** more games won than the opponent.

The tournament has **K** rounds, and there are **N** = $2^{\mathbf{K}}$ players in it, who all participate in the first round. In each round each of the remaining participating players is paired with another participating player, in order to play a single match. The winning player of each of these matches advances to the next round, whereas the losing player is automatically disqualified from the tournament. The winner of the only match of round **K** is thus the winner of the tournament.

It is desirable to make the tournament as long as possible, considering that the matches are broadcast on television and the Association is paid for every minute on air. Given any pair of different players, the Association knows the probability for one of them to win a game against the other. You are a member of the organizing committee, and your task is to prepare the fixture for the matches of each round in order to maximize the expected number of games played in the tournament. Doing this involves deciding which pairs of players will play a match in the first round, and then for each of the following rounds deciding which pair of matches from the previous round will provide the winning players for each match in that round. Note that this can only depend on the identification of the matches in the previous round. The following figure shows a possible fixture for **K** = 3 rounds and **N** = 2^3 = 8 players.



For the moment, the Association only wants to estimate their earnings, so it is sufficient to tell them the maximum expected number of games that can be played in the tournament.

Input

The input contains several test cases. Each test case is described using several lines. The first line contains four integer numbers, **K**, **S**, **J** and **D**. The value **K** denotes the number of rounds in the tournament (1 <= **K** <= 3). The value **S** denotes the number of sets that a player must win to win a match (1 <= **S** <= 10). The value **J** denotes the minimum number of games that a player must win in order to win a set, whereas **D** indicates that a player should win **D** more games than its opponent for him to win the set (1 <= **D** <= **J** <= 100). The players are identified by different integer numbers from 1 to **N** = $2^{\mathbf{K}}$. Each of the following **N** lines contains **N** values. In the **i**-th line, the **j**-th value is an integer number **P_ij** such that **p_ij** = **P_ij**/100 is the probability that player number **i** wins a game against player number **j** (0 <= **P_ij** <= 100 for 1 <= **i**, **j** <= **N**). You may assume that **P_ii** = 0 (no player opposes himself) and that **P_ij** + **P_ji** = 100 for **i** and **j** different.

The end of the input is denoted by a line containing four times the number -1.

Output

For each test case, you should print a line containing a rational number representing the maximum expected number of games that the given tournament can have. The result should be rounded to the nearest rational number with 2 decimal digits. In case of ties, round up. Note that you should always print 2 digits after the dot, even if this means ending with a zero.

Example

Input:

```
1 1 2 2
0 50
50 0
3 3 6 2
0 88 2 76 71 24 50 4
12 0 54 37 84 95 88 98
98 46 0 66 36 13 33 33
24 63 34 0 29 21 96 63
29 16 64 71 0 0 47 13
76 5 87 79 100 0 56 89
50 12 67 4 53 44 0 23
96 2 67 37 87 11 77 0
-1 -1 -1 -1
```

Output:

```
4.00
224.08
```