Dimensionality Reduction

Input file:	`standard input`
Output file:	`standard output`
Time limit:	15 seconds
Memory limit:	256 megabytes

Vitya has N points in an M-dimensional space, where each coordinate of a point is a random variable chosen uniformly. However, Vitya finds it difficult to work with such high-dimensional points, so he wants to linearly transform them to become K-dimensional, where $K < M$, while keeping the pairwise distances as unchanged as possible.

A point X of dimension M is represented as $X = (X_1, X_2, ..., X_M)$, where $X_1, X_2, ..., X_m \in \mathbb{R}$.

A linear transformation of a point X of size M to a point Y of size K is defined by a matrix A of size $M \times K$. Then $Y = (Y_1, ..., Y_K)$, where $Y_i = A_{1i} \cdot X_1 + A_{2i} \cdot X_2 + ... + A_{Mi} \cdot X_M$.

The distance between two points X and Y of equal size M is calculated as $\sqrt{(X_1 - Y_1)^2 + (X_2 - Y_2)^2 + ... + (X_M - Y_M)^2}$ (this distance is called Euclidean distance).

In the first test, $t = 3$. The score for this test is 30 points. Each subtest is worth between 0 and 10 points. In this test, all coordinates of the point except the first and second are the same.

In the second test, $t = 7$. The score for this test is 70 points. Each subtest is worth between 0 and 10 points.

Input

The first line contains three integers N, M, K ($10 \leq N \leq 1000, 20 \leq M \leq 100, 10 \leq K \leq 100$).

Each of the following N lines contains M integers that represent the coordinates of the corresponding point. Each coordinate is an integer from -1000 to 1000.

Output

You need to output M lines, each containing K real numbers (each number from -10^6 to 10^6), which form a linear transformation that minimizes the sum of the absolute difference between pairwise distances of the original points and the compressed points.

Example

standard input	standard output
2 2 3 2 1 2 2 -2 -1 3 2 2 1 1 2 3 4	1.00000 0 0 1.00000 0 0 1.00000 0

Note

In the test example, in the first part, there are 2 points of dimension 3.

Point $(1; 2; 2)$ will be transformed into point $(1 * 1 + 2 * 0 + 2 * 0; 1 * 0 + 2 * 1 + 2 * 0) = (1; 2)$, $(-2; -1; 3)$ will be transformed into point $((-2) * 1 + (-1) * 0 + 3 * 0; (-2) * 0 + (-1) * 1 + (-3) * 0) = (-2; -1)$.

In the second part of the test, point $(1; 2)$ will be transformed into point $(1 * 1 + 2 * 0) = (1)$; point $(3; 4)$ will be transformed into point $(3 * 1 + 4 * 0) = (3)$.

The initial distance between the first and second points is $\sqrt{(1-3)^2 + (2-4)^2} = \sqrt{8}$.

The new distance between the first and second points is $\sqrt{(1-3)^2} = 2$.

Therefore, the error in this case is the difference between the distance between the first and second original points and the distance between the first and second points after the linear transformation $|\sqrt{8} - 2|$.

Note that there exists a linear transformation that preserves the distance better.