

E. New Year Tree Decorations

time limit per test: 1 second

memory limit per test: 256 megabytes

input: standard input

output: standard output

Due to atheistic Soviet past, Christmas wasn't officially celebrated in Russia for most of the twentieth century. As a result, the Russian traditions for Christmas and New Year mixed into one event celebrated on the New Year but including the tree, a Santa-like 'Grandfather Frost', presents and huge family reunions and dinner parties all over the country. Bying a Tree at the New Year and installing it in the house is a tradition. Usually the whole family decorates the tree on the New Year Eve. We hope that Codeforces is a big and loving family, so in this problem we are going to decorate a tree as well.

So, our decoration consists of n pieces, each piece is a piece of colored paper, its border is a closed polyline of a special shape. The pieces go one by one as is shown on the picture. The i -th piece is a polyline that goes through points: $(0, 0)$, $(0, y_0)$, $(1, y_1)$, $(2, y_2)$, ..., (k, y_k) , $(k, 0)$. The width of each piece equals k .

The figure to the left shows the decoration, the figure to the right shows the individual pieces it consists of.

The piece number 1 (shown red on the figure) is the outer piece (we see it completely), piece number 2 (shown yellow) follows it (we don't see it completely as it is partially closed by the first piece) and so on. The programmers are quite curious guys, so the moment we hung a decoration on the New Year tree we started to wonder: what area of each piece can people see?

Input

The first line contains two integers, n and k ($1 \leq n, k \leq 300$). Each of the following n lines contains $k + 1$ integers — the description of the polyline. If the i -th line contains ontegers $y_{i,0}, y_{i,1}, \dots, y_{i,k}$, that means that the polyline of the i -th piece goes through points $(0, 0)$, $(0, y_{i,0})$, $(1, y_{i,1})$, $(2, y_{i,2})$, ..., $(k, y_{i,k})$, $(k, 0)$ ($1 \leq y_{i,j} \leq 1000$).

Output

Print n real numbers — for each polyline, the area of its visible part.

The answer will be considered correct if its relative or absolute error do not exceed 10^{-4} .

Examples

input
2 2 2 1 2 1 2 1
output
3.000000000000 0.500000000000

input
1 1 1 1
output
1.000000000000

input
4 1 2 7 7 2 5 5 6 4

output

4.5000000000000
1.2500000000000
0.0500000000000
0.0166666666667