Problem I. Colored Points

Input file:	`stdin`
Output file:	`stdout`
Time limit:	3 seconds
Memory limit:	256 megabytes

There are points of two colors on the plane. Each point is colored red or blue. You are to find such polygon that:

- it is convex and all it's vertices are in red points,
- the polygon should not contain in the inner part or on the perimeter any blue point.

Find the required polygon with the greatest area.

Input

The input starts with integer n ($1 \leq n \leq 100$) — the number of red points and n pairs of coordinates after it. The input continues with integer m ($1 \leq m \leq 100$) — the number of blue points and m pairs of coordinates after it. No two points are coincident. The coordinates are between -1000 and 1000.

Output

Print k — the number of vertices of the required polygon. Also print sequence $f_1, f_2, \ldots, f_k$, where f_i is the index of the red point which is the i-th vertex. The red points are numerated from 1 to n. Print f_i in clockwise or counter-clockwise order.

If there no required polygon with the positive area print the only number 0.

Example

stdin	stdout

3
0 0 3 0 0 4
1
1 1
	0

4
0 0 3 0 3 3 0 3
1
1 1
	3
	4 2 3

Problem J. Cigarettes

Input file:	`stdin`
Output file:	`stdout`
Time limit:	1 second
Memory limit:	256 megabytes

Peter has n cigarettes. He smokes them one by one keeping all the butts. Out of $k > 1$ butts he can roll a new cigarette. How many cigarettes can Peter have?

Input

Input is a sequence of lines. Each line contains two integer numbers giving the values of n and k ($k \leq n$).

Output

For each line of input, output one integer number on a separate line giving the maximum number of cigarettes that Peter can have. It fits to 32-bit signed integer.

Example

stdin	stdout

4 3
10 3
100 5
	5
	14
	124

Problem K. Matrix

Input file:	`stdin`
Output file:	`stdout`
Time limit:	1 second
Memory limit:	256 megabytes

There was a binary square symmetric matrix, there were zeros on the diagonal. Someone calculated numbers of ones in each row and wrote them. If two or more rows had the same number of ones the number was written only once. So the written sequence doesn't contain equal numbers.

Your task is to recover a matrix. It should be binary square symmetric matrix with zeros on the diagonal. Print the smallest (by size) matrix that fits in the given requirments.

Input

The first line contains m ($1 \le m \le 300$) — the length of the given sequence. The next line contains the given sequence. The number in it are between 0 and 300.

Output

If there is no solution print the only number `-1`. If solution exists the first line should contain t — the minimal possible matrix size. The following t lines should contain the required matrix.

Example

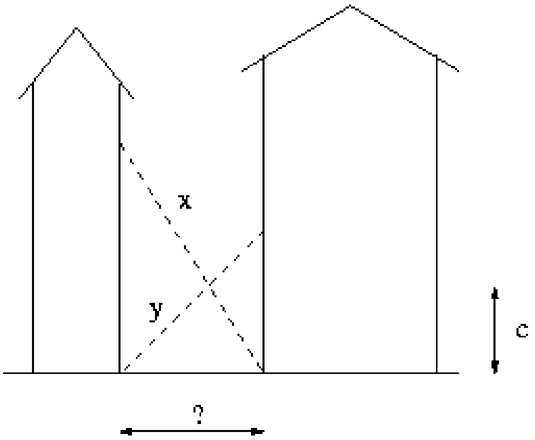
stdin	stdout

2
1 2
	3
	0 1 1
	1 0 0
	1 0 0

Problem L. Crossed ladders

Input file:	`stdin`
Output file:	`stdout`
Time limit:	1 second
Memory limit:	256 megabytes

A narrow street is lined with tall buildings. An x foot long ladder is rested at the base of the building on the right side of the street and leans on the building on the left side. A y foot long ladder is rested at the base of the building on the left side of the street and leans on the building on the right side. The point where the two ladders cross is exactly c feet from the ground. How wide is the street?



Input

Each line of input contains three positive floating point numbers giving the values of x, y, and c. Use 8-byte type to store and process given floating numbers (`double` in C++ or Java).

Output

For each line of input, output one line with a floating point number giving the width of the street in feet, with exactly three decimal digits in the fraction.

Example

stdin	stdout

30 40 10
12.619429 8.163332 3
10 10 3
10 10 1
	26.033
	7.000
	8.000
	9.798