Codeforces Beta Round #36

A. Extra-terrestrial Intelligence

time limit per test: 2 seconds
memory limit per test: 64 megabytes
input: input.txt
output: output.txt

Recently Vasya got interested in finding extra-terrestrial intelligence. He made a simple extra-terrestrial signals' receiver and was keeping a record of the signals for n days in a row. Each of those n days Vasya wrote a 1 in his notebook if he had received a signal that day and a 0 if he hadn't. Vasya thinks that he has found extra-terrestrial intelligence if there is a system in the way the signals has been received, i.e. if all the intervals between successive signals are equal. Otherwise, Vasya thinks that the signals were sent by some stupid aliens no one cares about. Help Vasya to deduce from the information given by the receiver if he has found extra-terrestrial intelligence or not.

Input

The first line contains integer n ($3 \le n \le 100$) — amount of days during which Vasya checked if there were any signals. The second line contains n characters 1 or 0 — the record Vasya kept each of those n days. It's guaranteed that the given record sequence contains at least three 1s.

Output

If Vasya has found extra-terrestrial intelligence, output YES, otherwise output NO.

Examples

input
8 00111000
output
YES

input
7 1001011
output
NO

input
7 1010100
output
YES

B. Fractal

time limit per test: 2 seconds
memory limit per test: 64 megabytes
input: input.txt
output: output.txt

Ever since Kalevitch, a famous Berland abstractionist, heard of fractals, he made them the main topic of his canvases. Every morning the artist takes a piece of graph paper and starts with making a model of his future canvas. He takes a square as big as $n \times n$ squares and paints some of them black. Then he takes a clean square piece of paper and paints the fractal using the following algorithm:

Step 1. The paper is divided into n^2 identical squares and some of them are painted black according to the model.

Step 2. Every square that remains white is divided into n^2 smaller squares and some of them are painted black according to the model.

Every following step repeats step 2.

Unfortunately, this tiresome work demands too much time from the painting genius. Kalevitch has been dreaming of making the process automatic to move to making 3D or even 4D fractals.

Input

The first line contains integers n and k ($2 \le n \le 3$, $1 \le k \le 5$), where k is the amount of steps of the algorithm. Each of the following n lines contains n symbols that determine the model. Symbol «.» stands for a white square, whereas «*» stands for a black one. It is guaranteed that the model has at least one white square.

Output

Output a matrix $n^k \times n^k$ which is what a picture should look like after k steps of the algorithm.

Examples

input
2 3 .* ..
output
.******* ..****** .*.***** ....**** .***.*** ..**..** .*.*.*.* ....

input
3 2 .*. *** .*.
output
.*.***.*. ********* .*.***.*. ********* ********* .*.***.*. ********* .*.***.*.

C. Bowls

time limit per test: 2 seconds
memory limit per test: 64 megabytes
input: input.txt
output: output.txt

Once Petya was in such a good mood that he decided to help his mum with the washing-up. There were n dirty bowls in the sink. From the geometrical point of view each bowl looks like a blunted cone. We can disregard the width of the walls and bottom. Petya puts the clean bowls one on another naturally, i. e. so that their vertical axes coincide (see the picture). You will be given the order in which Petya washes the bowls. Determine the height of the construction, i.e. the distance from the bottom of the lowest bowl to the top of the highest one.

Input

The first input line contains integer n ($1 \le n \le 3000$). Each of the following n lines contains 3 integers h, r and R ($1 \le h \le 10000, 1 \le r < R \le 10000$). They are the height of a bowl, the radius of its bottom and the radius of its top. The plates are given in the order Petya puts them on the table.

Output

Output the height of the plate pile accurate to at least 10^{-6}.

Examples

input
2 40 10 50 60 20 30
output
70.00000000

input
3 50 30 80 35 25 70 40 10 90
output
55.00000000

D. New Game with a Chess Piece

time limit per test: 2 seconds
memory limit per test: 64 megabytes
input: input.txt
output: output.txt

Petya and Vasya are inventing a new game that requires a rectangular board and one chess piece. At the beginning of the game the piece stands in the upper-left corner of the board. Two players move the piece in turns. Each turn the chess piece can be moved either one square to the right or one square down or jump k squares diagonally down and to the right. The player who can't move the piece loses.

The guys haven't yet thought what to call the game or the best size of the board for it. Your task is to write a program that can determine the outcome of the game depending on the board size.

Input

The first input line contains two integers t and k ($1 \le t \le 20$, $1 \le k \le 10^9$). Each of the following t lines contains two numbers n, m — the board's length and width ($1 \le n, m \le 10^9$).

Output

Output t lines that can determine the outcomes of the game on every board. Write «+» if the first player is a winner, and «-» otherwise.

Examples

input
10 2 1 1 1 2 2 1 2 2 1 3 2 3 3 1 3 2 3 3 4 3

output
- + + - - + - + + +

E. Two Paths

time limit per test: 2 seconds
memory limit per test: 64 megabytes
input: input.txt
output: output.txt

Once archaeologists found m mysterious papers, each of which had a pair of integers written on them. Ancient people were known to like writing down the indexes of the roads they walked along, as «a b» or «b a», where a, b are the indexes of two different cities joint by the road . It is also known that the mysterious papers are pages of two travel journals (those days a new journal was written for every new journey).

During one journey the traveler could walk along one and the same road several times in one or several directions but in that case he wrote a new entry for each time in his journal. Besides, the archaeologists think that the direction the traveler took on a road had no effect upon the entry: the entry that looks like «a b» could refer to the road from a to b as well as to the road from b to a.

The archaeologists want to put the pages in the right order and reconstruct the two travel paths but unfortunately, they are bad at programming. That's where you come in. Go help them!

Input

The first input line contains integer m ($1 \le m \le 10000$). Each of the following m lines describes one paper. Each description consists of two integers a, b ($1 \le a, b \le 10000$, $a \ne b$).

Output

In the first line output the number L_1. That is the length of the first path, i.e. the amount of papers in its description. In the following line output L_1 space-separated numbers — the indexes of the papers that describe the first path. In the third and fourth lines output similarly the length of the second path L_2 and the path itself. Both paths must contain at least one road, i.e. condition $L_1 > 0$ and $L_2 > 0$ must be met. The papers are numbered from 1 to m according to the order of their appearance in the input file. The numbers should be output in the order in which the traveler passed the corresponding roads. If the answer is not unique, output any.

If it's impossible to find such two paths, output «-1».

Don't forget that each paper should be used exactly once, i.e $L_1 + L_2 = m$.

Examples

input
2 4 5 4 3
output
1 2 1 1

input
1 1 2
output
-1