

A. Bender Problem

time limit per test: 2 seconds

memory limit per test: 256 megabytes

input: standard input

output: standard output

Robot Bender decided to make Fray a birthday present. He drove n nails and numbered them from 1 to n in some order. Bender decided to make a picture using metal rods. The picture is a closed polyline, which vertices should be nails (in the given order). The segments of the polyline should be parallel to the coordinate axes. Polyline is allowed to have self-intersections. Bender can take a rod and fold it exactly once in any place to form an angle of 90 degrees. Then he can attach the place of the fold to some unoccupied nail and attach two ends of this rod to adjacent nails. A nail is considered unoccupied if there is no rod attached to it (neither by its end nor by the fold place). No rod could be used twice. It is not required to use all the rods.

Help Bender to solve this difficult task.

Input

The first line contains two positive integers n and m ($4 \leq n \leq 500$, $2 \leq m \leq 500$, n is even) — the amount of nails and the amount of rods. i -th of the following n lines contains a pair of integers, denoting the coordinates of the i -th nail. Nails should be connected in the same order as they are given in the input. The last line contains m integers — the lengths of the rods. All coordinates do not exceed 10^4 by absolute value. Lengths of the rods are between 1 and 200 000. No rod can be used twice. It is guaranteed that all segments of the given polyline are parallel to coordinate axes. No three consecutive nails lie on the same line.

Output

If it is impossible to solve Bender's problem, output NO. Otherwise, output YES in the first line, and in the second line output n numbers — i -th of them should be the number of rod, which fold place is attached to the i -th nail, or -1 , if there is no such rod.

If there are multiple solutions, print any of them.

Examples

input
4 2 0 0 0 2 2 2 2 0 4 4
output
YES 1 -1 2 -1

input
6 3 0 0 1 0 1 1 2 1 2 2 0 2 3 2 3
output
YES 1 -1 2 -1 3 -1

input

6 3
0 0
1 0
1 1
2 1
2 2
0 2
2 2 3

output

NO