Problem B. Dominating Set

Input file:	*standard input*
Output file:	*standard output*
Time limit:	3 seconds
Memory limit:	64 mebibytes

Bobo has a bipartite graph $G = (V, E)$ with n vertices and m edges. He would like to choose a subset D of vertices such that for each vertex v not included in D, at least one of its neighbours is in D. Find the number of possible subsets bobo can choose.

Note:

1. A graph G is bipartite if and only if G contains no cycles of odd length.
2. A vertex u is a neighbour of v if and only if u and v are connected by an edge.

Input

The first line contains two integers n and m ($1 \le n \le 30$, $0 \le m \le 225$).

The i-th of the following m lines contains two integers a_i and b_i which denote an edge between the a_i-th and b_i-th vertices ($1 \le a_i, b_i \le n$).

It is guaranteed that there are no self loops and no multiple edges in the graph.

Output

Print one integer: the number of different subsets bobo can choose.

Examples

standard input	standard output
4 4 1 2 2 3 3 4 4 1	11
4 0	1

Problem F. Similar Subsequence

Input file:	*standard input*
Output file:	*standard output*
Time limit:	2 seconds
Memory limit:	64 mebibytes

Two sequences $\{a_1, a_2, \ldots, a_n\}$ and $\{b_1, b_2, \ldots, b_n\}$ are similar if and only if $(a_i - a_j) \cdot (b_i - b_j) > 0$ for all $1 \le i, j \le n$.

Bobo has two sequences $A = \{a_1, a_2, \ldots, a_n\}$ and $B = \{b_1, b_2, \ldots, b_m\}$, and he would like to check whether B contains a subsequence similar to A.

Input

The first line contains two integers n and m ($1 \le n, m \le 500$).

The second line contains n integers $a_1, a_2, \ldots, a_n$ ($1 \le a_i \le n$).

The third line contains m integers $b_1, b_2, \ldots, b_m$ ($1 \le b_i \le m$).

It is guaranteed that A is a permutation of $\{1, 2, \ldots, n\}$ and contains **no** subsequences similar to either $\{2, 1, 3\}$ or $\{2, 3, 1\}$.

Output

Print "`Yes`" if B contains a subsequence similar to A, or "`No`" otherwise.

Examples

standard input	standard output
3 4 1 2 3 1 3 2 4	Yes
3 4 1 2 3 4 4 4 4	No

Problem G. Random Arithmetics

Input file:	*standard input*
Output file:	*standard output*
Time limit:	2 seconds
Memory limit:	64 mebibytes

Bobo is playing with n integers $a_1, a_2, \ldots, a_n$. He repeatedly applies the following operation: choose a pair of integers x and y from them and replace them with either $x + y$ or $x \cdot y$. Each pair has the same probability of being chosen, and the arithmetic operation is also picked equiprobably. Thus, for example, there were $n \cdot (n - 1)$ outcomes with equal probability after the first operation.

After bobo repeated the operation described above $(n - 1)$ times, exactly one integer remained. Bobo would like to know the expected value of the remaining integer.

Input

The first line contains an integer n $(2 \le n \le 2000)$.

The second line contains n integers $a_1, a_2, \ldots, a_n$ $(0 \le a_i \le 10^9)$.

Output

If the expected value is $\frac{P}{Q}$, output $P \cdot Q^{-1} \bmod (10^9 + 7)$.

Note that Q^{-1} is the multiplicative inverse to Q: the number such that $Q \cdot Q^{-1} \equiv 1 \pmod{(10^9+7)}$.

Examples

standard input	standard output
2 1 1	500000005
3 1 2 3	250000008

Problem J. Welcome to ICPCCamp 2016!

Input file:	*standard input*
Output file:	*standard output*
Time limit:	2 seconds
Memory limit:	64 mebibytes

Welcome to ICPCCamp 2016! Bobo was now solving (maybe) the easiest problem in the contest.

Bobo was given 6666 integers $a_1, a_2, \ldots, a_{6666}$ between 1 and 2016. He was asked to find a subset of 2016 integers such that their sum is a multiple of 2016.

"Stupid..." Bobo murmured. However, he quickly found that such problem would not be solvable in any of the following years until ICPCCamp 2025. Can you figure out why?

Input

Each of the 6666 lines contains an integer a_i ($1 \le a_i \le 2016$).

Output

Print 2016 distinct indices $b_1, b_2, \ldots, b_{2016}$ such that Bobo could use $a_{b_1}, a_{b_2}, \ldots, a_{b_{2016}}$ as the answer.

Any 2016 indices such that the sum of the respective numbers is a multiple of 2016 will be accepted.

Example

standard input	standard output
1 1 1 ...(6660 lines omitted) 1 1 1	1 2 3 ...(2010 lines omitted) 2014 2015 2016

Problem K. Ant's Way

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	64 mebibytes

Given a simple polygon on the plane (polygon is called simple if its area is greater than zero and two edges have one common point if they share a vertice and no common points otherwise). The ant sits in the point (x_0, y_0) and starts moving straightforward along the vector (v_x, v_y).

Compute length of the ant's way, which lies strictly inside the polygon.

Input

First line of the input contains one integer N — number of vertices in the polygon ($1 \le N \le 100$). Each of next N lines contains coordinates x_i and y_i of one vertice of the polygon, coordinates are listed either in clockwise or counterclockwise order. Last line contains coordinates (x_0, y_0) of the points, where ant sits at the start of process and coordinates of vector of its movement (v_x, v_y).

All coordinates are real numbers given with no more than with three digits after the decimal point and does not exceed 1000 by their absolute value.

Output

Print one number — length of ant's path, lying strictly inside the polygon, with absolute or relative error 10^{-5} or better.

Examples

standard input	standard output
4 0 0 0 1 1 1 1 0 -1 -1 1 1	1.4142135624
4 0 0 0 1 1 1 1 0 -1 0 1 0	0.00000
5 0 0 1 10 2 1 3 10 4 0 -1 1 1 0	3.8000000000

Problem L. Cut The Rectangle

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	64 mebibytes

Given two triangles by their side lengths.

Check if it is possible to produce those triangles by cutting some rectangle with a single line segment and freely flipping and/or rotating the resulting pieces.

Input

The input consists of two lines. The first line consists of three space-separated integers, indicating the side lengths of the first triangle.

Similarly, the second line contains three space-separated integers, denoting the side lengths of the second triangle.

It is guaranteed that the side lengths produce valid triangles. You may assume that the maximum side length of a triangle is 100, and that the minimum is 1.

Output

If there exists a rectangle which could have been cut to form triangles of the given side lengths, output 1. Otherwise, output 0.

Examples

standard input	standard output
6 8 10 8 6 10	1
7 4 5 4 5 7	0

Problem M. Diversity

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	64 mebibytes

Lets call number of distinct letters in the given string the *diversity* of this string. For example, the string "`acm`" has diversity 3, same as the string "`icpc`".

Bobo likes strings which have diversity either 1 or 2. Someone has given him a string and Bobo wants to turn it into a string that he like. Bobo got a magic eraser which will delete one letter from any place of string.

Compute the minimum number of letters he must erase in order to turn the string into a string with diversity at most 2.

Input

The input will consist of a line with a single string consisting of at least 1 and at most 100 lowercase letters.

Output

Print a single integer, indicating the minimum number letters Bobo need to erase in order to give the string a diversity of 1 or 2.

Examples

standard input	standard output
bobo	0
china	3
acmicpc	3

Problem N. Cogwheels

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	64 mebibytes

Let we have n cogwheels installed on the plane. You are given the coordinate of axis and radius of each cogwheel. Let first year is source, and n-th is the target.

Check what happens to the target cogwheel if you will try to turn the source:

- the source cogwheel is locked and can't move, because otherwise it cause some cogwheel to turn in both directions simultaneously, or
- the source cogwheel can rotate, but it is not connected to the target cogwheel, or
- the source cogwheel turns the target cogwheel, then print a signed ratio for them.

In case when source cogwheel is locked and not connected to the target cogwheel at the same time, information about lock have higher priority.

Input

The first line of input contains one integer — number of cogwheels n ($1 \le n \le 1,000$), the total number of cogwheels. Then n lines follow, one per cogwheel, each line contains the x, y ($-10^4 \le x, y \le 10^4$) and r ($1 \le r \le 10^4$), where (x, y) is the position of the center of the cogwheel, and r — radius of the cogwheel. Source cogwheel listed first, target cogwheel listed n-th.

Assume that a cogwheel will mesh with any other cogwheel if (and only if) they are tangent to each other. It is guaranteed that no two cogwheels overlap.

Output

Output a single line, with the following content, based on the result:

- -1 if the source cogwheel is blocked.
- 0 if the source cogwheel can move but is not connected to the target.
- Two space-separated integers a and b, if the source cogwheel moves the target cogwheel, where $a : b$ is the ratio of source cogwheel revolutions to target cogwheel revolutions, $gcd(a, b) = 1$, $a > 0$, $b > 0$ if the target turns in the same direction as the source, and negative otherwise.

Example

standard input	standard output
2 0 0 10 0 30 20	2 -1
2 0 0 1 0 3 1	0
3 0 0 11 0 33 22 44 0 33	-1

Problem O. Pairs

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	64 mebibytes

In the game ¡¡Pairs¿¿ special cards are used. Each card have two integers on it: one denotes a suit and other denotes a value. Cards are not unique (i.e. may exist two cards with same suit and same value).

Player gets set of cards from dealer. Then for each suit, represented in this set, player must discard exactly two cards of this suit (if he can't because he got only one card in some suit, player lose immediately and receive no bonus). After that, bonus of the deal is calculated as sum of all values for cards, which were not discarded.

Given information about cards you have, find out cards you must discard to obtain maximum bonus or determine that this hand is losing.

Input

First line of the input contains one integer N ($1 \le N \le 299999$) — number of cards on your hand. i'th of next N lines contains two integers — description of the i-th leftmost card: suit s_i ($1 \le s_i \le 10^9$) and value v_i ($0 \le v_i \le 10^9$). Cards in the hand are numbered sequentially starting from 1.

Output

If you will lose immediately with this hand, print -1. In other case print in first line number K of suits you have, then for each suit print in the new line two integers — indices in the hand (counting from left to right) of cards, which must be discarded for this suit. Suits can be listed in arbitrary order; if several solution exists, you may print any of them.

Examples

standard input	standard output
10 9 2 9 3 7 1 7 2 9 4 9 3 1 2 1 0 1 0 7 3	3 3 4 1 2 8 9

Problem P. Trip Between Corners

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	64 mebibytes

Bobo stays on an $n \times m$ grid where each square on the grid contains a digit on it. From a given square that has digit k on it, Bobo can move only jumping exactly k squares in one of the four cardinal directions. Bobo cannot go beyond the edges of the grid; grid does not wrap.

What is the minimum number of moves required for Bobo to get from the top-left corner to the bottom-right corner?

Input

The first line of input contains two integers n and m ($1 \le n, m \le 500$) — dimensions of the grid. It is guaranteed that at $mn > 1$.

Each of the next n lines will contain string of exactly m digits — the $n \times m$ grid. Each digit is between 0 and 9, inclusive.

The top-left corner of the grid will correspond to the first character in the first line of the input, and the bottom-right one — to he last character in the n-th line of the input.

Output

If Bobo can move from top-let corner to the bottom-right, print minimum number of moves required for that, otherwise print -1.

Examples

standard input	standard output
3 3 111 111 111	4
2 2 33 33	-1