Problem A. Circles

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1.5 seconds
Memory limit:	256 mebibytes

There are N circles on a plane. The center of the i-th circle is (x_i, y_i), and the radius of this circle is $\sqrt{x_i^2 + y_i^2}$.

Count the number of lattice points (points with integer coordinates) that are inside at least one of the circles. Note that the boundaries are considered to be inside.

Input

Input format:

N
$x_1 \ y_1$
$x_2 \ y_2$
$\vdots$
$x_N \ y_N$

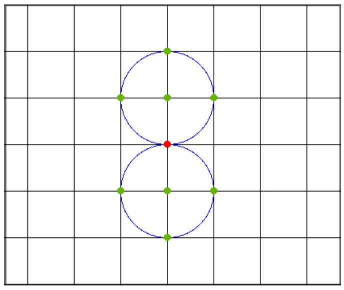
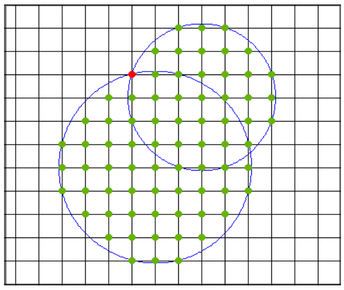
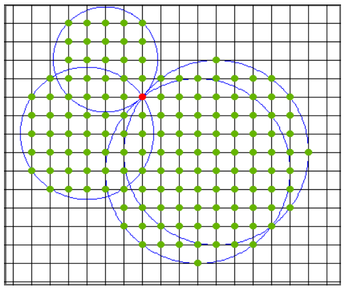
Constraints:

- $1 \le N \le 10^5$
- $-10^5 \le x_i, y_i \le 10^5$
- (x_i, y_i) are pairwise distinct.
- $(x_i, y_i) \ne (0, 0)$
- All values in the input are integers.

Output

Print the answer in a single line.

Examples

standard input	standard output	Notes
2 0 1 0 -1	9	
2 3 1 1 4	74	
4 3 4 4 3 -2 -2 -3 2	146	

Problem B. Point Pairs

Input file:	*standard input*
Output file:	*standard output*
Time limit:	2 seconds
Memory limit:	256 mebibytes

There are $2N+1$ points on a plane. The i-th point is at (X_i, Y_i). Two points i and j can be paired if $X_i = X_j$ or $Y_i = Y_j$.

For each point, determine the following:

- If you remove this point from the set of points, you get $2N$ points. Can these $2N$ points be separated into N disjoint pairs?

Input

Input format:

N
$X_1\ \ Y_1$
$X_2\ \ Y_2$
$\vdots$
$X_{2N+1}\ \ Y_{2N+1}$

Constraints:

- $1 \le N \le 100,000$
- $1 \le X_i, Y_i \le 2N+1$
- The points are pairwise distinct.
- All values in the input are integers.

Output

Output $2N+1$ lines. For the i-th line, print "OK" if all points except for the i-th can be separated into N disjoint pairs. Otherwise print "NG".

Examples

standard input	standard output
1 1 1 1 2 2 1	NG OK OK
2 1 1 1 2 2 2 2 3 3 3	OK NG OK NG OK
2 1 1 1 2 3 3 4 4 4 5	NG NG OK NG NG

Problem C. House Moving

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1.5 seconds
Memory limit:	256 mebibytes

There are N houses numbered 1 through N. The distance between the house i and the house j is $|i - j|$.

You want to assign M families to these houses. There are P_i people in the i-th family. No two families can be assigned to the same house.

Your objective is to maximize the *distance of residents*. For each (unordered) pair of two people among the M families, compute the distance between their houses. The *distance of residents* is defined as the sum of these values for all pairs.

Compute the maximum possible value of the *distance of residents*.

Input

Input format:

$N \quad M$
P_1
P_2
$\vdots$
P_M

Constraints:

- $2 \le N \le 10^6$
- $2 \le M \le \min(N, 1000)$
- $1 \le P_i \le 100$

Output

Print the answer in a single line.

Examples

standard input	standard output
4 3 1 1 2	11
10 10 3 1 4 1 5 9 2 6 5 3	2998
20 10 2 7 1 8 2 8 1 8 2 8	9852

Note

In the Sample 1, let A be the member of the first family, B be the member of the second family, and C, D be the members of the third family.

In the optimal assignment, the first family shuold go to the house 1, the second family should go to the house 2, and the third family shuold go to the house 4.

- The distance between A and B: 1
- The distance between A and C: 3
- The distance between A and D: 3
- The distance between B and C: 2
- The distance between B and D: 2
- The distance between C and D: 0

The *distance of residents* is 11.

Problem D. Nice Set of Points

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	256 mebibytes

Consider a set of points. You can move directly between two points if their x-coordinates are the same or their y-coordinates are the same. A set of points is called *nice* if for any two points in the set, the length of the shortest (direct or indirect) path is equal to the manhattan distance between them.

You are given N points. The i-th point is at (x_i, y_i).

You are allowed to add up to $10000 - N$ points. Convert the given set of points into a nice set.

Input

Input Format:

N
$x_1\ \ y_1$
$x_2\ \ y_2$
$\vdots$
$x_N\ \ y_N$

Constraints:

- $2 \le N \le 1000$
- $1 \le x_i, y_i \le 1000$
- The points are pairwise distinct.
- Under these constraints, it is guaranteed that at least one solution exists.
- All values in the input are integers.

Output

Let $M (0 \le M \le 10000-N)$ be the number of added points, and $(s_1, t_1), \ldots, (s_M, t_M)$ be their coordinates. After adding these M points to the set, you get $N+M$ points. **These $N+M$ points must be pairwise distinct**, and this set must be nice. The coordinates must be integers.

Output the answer in the following format.

M
$s_1\ \ t_1$
$s_2\ \ t_2$
$\vdots$
$s_N\ \ t_M$

If there are multiple possible solutions, output any.

Examples

standard input	standard output
2 1 1 2 2	1 1 2
4 1 1 2 2 3 4 4 3	4 1 2 3 2 3 3 4 4
7 2 4 3 2 4 6 5 1 6 5 7 3 8 7	15 3 6 8 5 2 2 7 5 2 5 6 6 3 1 5 6 6 2 6 1 7 1 7 2 2 3 6 7 2 6

Note

In Sample 1, if you add $(1, 2)$, you can move between $(1, 1)$ and $(2, 2)$ via $(1, 2)$.

Problem E. Eel and Grid

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1.5 seconds
Memory limit:	256 mebibytes

There is an $H \times W$ grid. Let $(i,\ j)$ be the cell at the intersection of the i-th row $(0 \le i \le H-1)$ and the j-th column $(0 \le j \le W-1)$. Initially, there is an eel at the cell $(0,\ 0)$. The eel repeats the following process.

- If the current cell is painted, end the process.
- If the current cell is not painted, paint the cell and move to another cell. If the current cell is $(i,\ j)$, the new cell must be either $((i+1) \bmod H,\ j)$ or $(i,\ (j+1) \bmod W)$.

Count the number of ways to paint all cells and end the process at the cell $(0,\ 0)$, modulo 10^9+7. Two ways are considered distinct if the path traveled by the eel are distinct.

Input

Input Format:

$H \quad W$

Constraints:

- $2 \le H, W \le 10^6$

Output

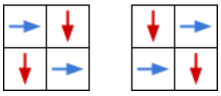
Print the answer modulo 10^9+7.

Examples

standard input	standard output
2 2	2
6 3	3
3 4	0
10 10	260
200 300	551887980

Note

The following picture shows the two ways in Sample 1:



Problem F. Right Angle Painting

Input file:	*standard input*
Output file:	*standard output*
Time limit:	4 seconds
Memory limit:	256 mebibytes

Takahashikun likes to paint floors. There is a floor divided into $N \times N$ grid, and some (possibly zero) cells may contain obstacles.

The information about the grid is given as N strings $S_1, \ldots, S_N$. The j-th character of S_i represents the cell (i, j): '`.`' and '`s`' represent an empty cell, and '`#`' represents a cell with obstacles.

There is excatly one cell with '`s`'. First, Takahashikun enters the cell with '`s`' and paints this cell. After that, he makes zero or more steps according to the following rule:

- In each step, he moves to one of (vertically or horizontally) adjacent cells and paint it.
- Except for the first step, the direction of movement must be changed by 90 degrees from the previous step. That is, after he moves horizontally he must move vertically, and vice versa.
- He must not enter already painted cells.
- He must not enter cells with obstacles.
- He must not go out of the grid.

Determine if he can paint all cells without obstacles.

Input

Input Format:

N
S_1
S_2
$\vdots$
S_N

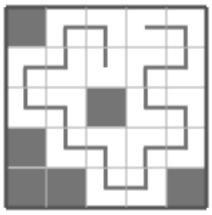
Constraints:

- $1 \leq N \leq 400$
- $|S_i| = N$
- Each character in S_i is one of '`.`', '`#`', or '`s`'.
- There is exactly one cell with '`s`'.
- There is at least one cell with '`.`'.

Output

Print "`POSSIBLE`" if he can paint all cells without obstacles. Otherwise print "`IMPOSSIBLE`".

Examples

standard input	standard output	Notes
5 #.... ..s.. ..#.. #.... ##..#	POSSIBLE	
5 s.### ..### ###.. #.... #..##	IMPOSSIBLE	

Problem G. Rectangle-free Grid

Input file:	*no input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	256 mebibytes

Construct an $N \times N$ grid with the following conditions:

- $2 \leq N \leq 150$
- Each cell is filled with either '`O`' or '`.`'.
- There are at least 1700 cells with '`O`'.
- For each tuple of four integers i, j, k, l such that $1 \leq i < j \leq N$ and $1 \leq k < l \leq N$, at least one of the four cells (i, k), (i, l), (j, k), (j, l) is filled with '`.`'.

Input

There is no input.

Output

The first line should contain an integer N. The following N lines should contain N characters each ('`O`' or '`.`'), and these N lines describe the grid.

Example

no input	standard output
	5 ....OOO ...O. ..O..

Note

Output for this example satisfies all conditions but the third (number of '`O`' in the grid).

Problem H. Cups and Beans

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	256 mebibytes

There are N cups numbered 0 through $N-1$. For each $i (1 \leq i \leq N-1)$, the cup i contains A_i beans, and this cup is labeled with an integer C_i.

Two people will play the following game:

- In each turn, the player chooses a bean from one of the cups except for the cup 0.
- If he chooses a bean from the cup i, he must move it to one of the cups $i - C_i, \ldots, i-1$.
- The players take turns alternately. If a player can't choose a bean, he loses.

Who will win if both players play optimally?

Input

Input Format:

N
$C_1 \;\; A_1$
$C_2 \;\; A_2$
$\vdots$
$C_{N-1} \;\; A_{N-1}$

Constraints:

- $2 \leq N \leq 10^5$
- $1 \leq C_i \leq i$
- $0 \leq A_i \leq 10^9$
- At least one of A_i is nonzero.
- All values in the input are integers.

Output

Print the name of the winner: "`First`" or "`Second`".

Examples

standard input	standard output
3 1 0 1 1	Second
7 1 1 2 0 1 0 2 0 4 1 3 0	First
7 1 1 2 0 1 9 2 10 4 3 3 5	Second

Note

Notes to the Sample 1:

- In the first turn, the first player must move a bean from 2 to 1.
- In the second turn, the second player must move a bean from 1 to 0.
- In the third turn, the first player can't choose a bean and loses.

Problem I. Edge Coloring

Input file:	*standard input*
Output file:	*standard output*
Time limit:	2 seconds
Memory limit:	256 mebibytes

You are given a simple connected undirected graph with N vertices and M edges. The i-th edge connects the vertices a_i and b_i.

Initially, the edges are not colored. Takahashikun wants to color the i-th edge with the color c_i.

He can color the edges in the following way:

- First he chooses a vertex, and he repeats zero or more steps.
- In each step, he chooses a vertex adjacent to the current vertex and moves to the chosen vertex along an edge. This edge is colored red or blue (the color is defined according to the rule below).
- In odd-indexed (1-based) steps he uses red. In even-indexed steps he uses blue.
- If he colors an already-colored edges, the color of the edge is updated to the new color.

Determine if he can color all edges with correct colors.

Input

Input Format:

N M
a_1 b_1 c_1
a_2 b_2 c_2
$\vdots$
a_M b_M c_M

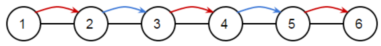
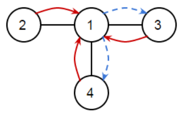
Constraints:

- $2 \le N \le 2000$
- $1 \le M \le 2000$
- $1 \le a_i < b_i \le N$
- The pairs (a_i, b_i) are pairwise distinct.
- c_i is either an '`r`' (red) or a '`b`' (blue).
- The graph is connected.

Output

Print "`Yes`", if Takahashikun can color all edges with correct colors or "`No`" otherwise.

Examples

<table>
<tr><th>standard input</th><th>standard output</th><th>Notes</th></tr>
<tr><td>6 5
1 2 r
2 3 b
3 4 r
4 5 b
5 6 r</td><td>Yes</td><td>
Start from vertex 1.</td></tr>
<tr><td>4 3
1 2 r
1 3 r
1 4 r</td><td>Yes</td><td>
Start from vertex 2.
Dashed lines represent
overpainted colors.</td></tr>
<tr><td>3 3
1 2 b
1 3 b
2 3 b</td><td>No</td><td></td></tr>
</table>

Problem J. Travel in Sugar Country

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1.5 seconds
Memory limit:	256 mebibytes

There are N towns numbered 1 through N. There is a bidirectional road between towns i and $i+1$, and its length is D_i. Thus, for each pairs $(a,\ b)$ $(a < b)$, the distance between towns a and b is $D(a,b) = D_a + D_{a+1} + \ldots + D_{b-1}$.

At each town there is a sugar shop. An ant wants to visit K distinct shops.

The ant wants to choose a set of K distinct shops and the order to visit them. For example, if it decides to visit the shops $S_1, \ldots, S_K$ in this order, the total distance it travels will be $D(S_1, S_2) + D(S_2, S_3) + \ldots + D(S_{K-1}, S_K)$.

In how many ways the total distance it travels become a multiple of M? Print the answer modulo 10^9+7.

Input

Input Format:

N M K
D_1
D_2
$\vdots$
D_{N-1}

Constraints:

- $2 \le N \le 100$
- $1 \le M \le 30$
- $2 \le K \le 10, K \le N$
- $1 \le D_i \le M$
- All values in the input are integers.

Output

Print the answer modulo $10^9 + 7$.

Examples

standard input	standard output
4 4 3 2 1 3	6
15 5 10 5 5 5 5 5 5 5 5 5 5 5 5 5 5	897286330

Note

In Sample 1, there are six ways: $1 \to 3 \to 2$, $2 \to 3 \to 1$, $2 \to 1 \to 4$, $4 \to 1 \to 2$, $2 \to 3 \to 4$, and $4 \to 3 \to 2$.