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 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:

- $2 \leq N \leq 1000$
- $1 \leq x_i, y_i \leq 1000$
- 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 $M (0 \leq M \leq 10000-N)$ 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 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 $i(1 \le i \le N-1)$, 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:

- $2 \le N \le 10^5$
- $1 \le C_i \le i$
- $0 \le A_i \le 10^9$
- 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 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:

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

Constraints:

- $2 \leq N \leq 100$
- $1 \leq M \leq 30$
- $2 \leq K \leq 10, K \leq N$
- $1 \leq D_i \leq M$
- 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 $4 \to 3 \to 2$.

Problem K. Campus

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

Your university hired several workers to tear down the obsolette wall between new and old campus.

You are given a description of the wall as a sequence of points in a 2-D coordinate system; all coordinates will be integers, and we will make sure that all wall pieces are either horizontal or vertical, but never at an angle. You also know, how much wall (length) per hour one person can tear down, and how many workers are hired.

From this, you are to compute the necessary time to tear down the entire wall.

Input

The first line of the input contains three integers n, s and f. The integer $1 \le n \le 1000$ is the number of straight-line segments in the wall. $0 < s < 100$ is the number of hours it takes one person to tear down one meter of wall. $1 \le p \le 1000$ is the number of workers hired to tear down the wall.

This is followed by $n + 1$ lines, each containing two integers x_i, y_i with $-10^4 \le x_i, y_i \le 10^4$. This is the i-th point describing the wall. The i-th segment of the wall runs from (x_i, y_i) to (x_{i+1}, y_{i+1}). As promised above, all wall pieces are horizontal or vertical, meaning that for each i, either $x_{i+1} = x_i$ or $y_{i+1} = y_i$. Furthermore, we will ensure that the wall never crosses itself.

Output

Output the number of hours it will take the p workers to tear down the entire wall, rounded up to the nearest integer. (So if it would take 3.1 hours, you should output 4, not 3.)

Example

standard input	standard output
3 2 1 -2 4 1 4 1 2 3 2	14

Problem L. Race

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

When you organize an racing team, you have quite a few optimization problems to solve. One of them is how to utilize your available racing cars and drivers to get as much cars as possible participating in the race.

You will be given a list of currently available cars and drivers, and for each driver the list of cars this pilot could possibly drive.

Input

The first line of the input contains two integers $0 \le m, n \le 200$. m is the number of available cars, while n is the number of drivers. This is followed by n lines, one for each driver. The first number on line i is the number $0 \le m_i \le m$ of cars that driver i is capable of driving. This is followed m_i integers, each between 1 and m, describing the m_i distinct cars that i can drive.

Output

Print the maximum total number of cars you can send to the race.

Example

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

Problem M. Tip

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

At his favorite restaurant "Ukkonen and Tree" famous coach Mike follows next rules:

- He pays only whole dollar amount;
- He leaves at least 0.2 tip when he eats out (i.e if meal is $40, Mike leaves $8 for tip, so totally he pays at least $48, if meal is $14, then tip would be $2.80, but because Mike pays only whole dollar amount, totally he pays at least $17).
- If his total bill (meal plus tip) is not a palindrome, he will increase the total (by adding to the tip) to make the total a palindrome. He will, of course, add the minimum needed to make the total a palindrome.

Given Mike's meal cost, your program should determine the total bill (following his rules).

Input

First line of the input contains one integer T ($1 \le T \le 10^4$) — number of the test cases.

Each test case consists of one integer m ($5 \le m \le 10^4$) — cost of Mike's meal in dollars.

Output

For each test case print one integer — total bill paid by Mike following his rules.

Example

standard input	standard output
2 14 83	22 101

Problem N. Integer Triangle

Input file:	
Output file:	
Time limit:	2 seconds
Memory limit:	256 mebibytes

Given two integers a and b, find out number of different integers c such as exists non-degenerated triangle with sides a, b and c.

Input

First line of the input contains one integer T $(1 \le T \le 100)$ — number of test cases.

Each test case is placed on separate line and consists of two integers a and b $(1 \le a, b \le 100)$.

Output

For each test case print one integer — number of different integer sides of triangle with sides a and b.

Example

1 4 3	5

Problem O. Number of Even

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

For given integers A and B count number of even integers between A and B, inclusive.

Input

First line of the input contains two integers A and B ($-10^9 \le A, B \le 10^9$).

Output

Print one integer — answer to the problem.

Example

standard input	standard output
1 1	0
2 -4	4