Problem A. 2016

Input file:	*standard input*
Output file:	*standard output*
Time limit:	2 seconds
Memory limit:	512 mebibytes

Happy New Year! The integer 2016 has exceptionally many divisors.

Let $d(n)$ be the number of divisors of n. For example, $d(12) = 6$ because it has 6 divisors: 1, 2, 3, 4, 6, and 12. A positive integer x is called *divisorful* if the number of positive integers y that satisfy both $y < x$ and $d(y) > d(x)$ is at most one. For example, 2016 is a divisorful number because among integers smaller than 2016, only 1680 has more divisors than 2016.

You are given an integer K. Compute the K-th (1-based) smallest divisorful number. If such number is strictly greater than 10^{18}, print -1 instead.

Input

The input contains one integer K ($1 \le K \le 10^9$).

Output

Print the answer in a single line.

Examples

standard input	standard output
10	14
1000000000	-1

Note

The smallest divisorful numbers are 1, 2, 3, 4, 5, 6, 8, 10, 12, 14, ...

Problem B. Airports

Input file:	*standard input*
Output file:	*standard output*
Time limit:	2 seconds
Memory limit:	512 mebibytes

Snuke is the owner of N airports. The coordinates of the i-th airport are (x_i, y_i). Snuke chooses a constant D, and for each pair of two airports p and q, adds a flight between these two airports if the Manhattan distance between p and q is at least D. Compute the maximum D that makes the airports connected (that is, any airport is reachable from any other airport by using one or more flights).

Note that the Manhattan distance between two points with coordinates (x_1, y_1) and (x_2, y_2) is defined as $|x_1 - x_2| + |y_1 - y_2|$.

Input

First line of the input contains one integer N ($2 \le N \le 10^5$). Then N lines follow, i-th of them contains two integers x_i and y_i — coordinates of the i-th airport ($0 \le x_i, y_i \le 10^9$). No two airports share the same position.

Output

Print the answer to the problem in a single line.

Example

standard input	standard output
6 1 7 8 5 6 3 10 3 5 2 6 10	9

Problem C. Jump

Input file:	*standard input*
Output file:	*standard output*
Time limit:	2 seconds
Memory limit:	512 mebibytes

Snuke is standing on an infinitely long road.

The position on this road is represented by a real number.

Snuke can perform N types of jumps. The jump of type i is symmetric with respect to the point a_i. That is, if he performs this jump at point x, he will jump to $2a_i - x$).

You are given Q queries. In the i-th query, you are asked to compute the minimum number of jumps Snuke must perform to go from s_i to t_i. If t_i is unreachable from s_i by performing a series of jumps, print -1 instead.

Input

First line of the input contains one integer N ($1 \le N \le 200$). Next N lines contain integers a_i, one per line ($0 \le a_1 < \ldots < a_N \le 10^4$). Next line contains one integer Q — the number of queries ($0 \le Q \le 10^5$). Each of the next Q lines contains one query and consists of two integers s_i and t_i ($0 \le s_i, t_i \le 10^4$).

Output

For each query, print the answer in a single line.

Example

standard input	standard output
4 1 2 4 7 10 2 3 5 6 6 0 3 7 10 3 7 6 5 5 2 10 4 10 10 10	-1 -1 2 2 -1 -1 0 3 1 0

Problem D. Merge

Input file:	*standard input*
Output file:	*standard output*
Time limit:	2 seconds
Memory limit:	512 mebibytes

Snuke wants to create an array R by merging two arrays P and Q. Formally, the array R is obtained in the following way:

- Initially, the array R is empty.
- While at least one of P and Q is non-empty, choose a non-empty array (P or Q), pop its leftmost element, and attach it to the right end of R.

You are given P and Q, they are permutations of $1, \ldots, N$. Compute the number of possible distinct arrays Snuke can create, and print the answer modulo $10^9 + 7$.

Input

First line of the input contains one integer N ($1 \leq N \leq 2000$). Second line contains N integers P_i ($1 \leq P_i \leq N$, $P_i \neq P_j$ if $i \neq j$). Third line contains N integers Q_i ($1 \leq Q_i \leq N$, $Q_i \neq Q_j$ if $i \neq j$).

Output

Print the answer in a single line.

Examples

standard input	standard output
4 3 1 2 4 3 1 2 4	14
10 5 7 3 1 6 4 2 10 9 8 2 8 9 1 5 6 10 4 3 7	127224

Problem E. Mirror Rice Cake

Input file:	*standard input*
Output file:	*standard output*
Time limit:	2 seconds
Memory limit:	512 mebibytes

Mirror Rice Cake (a stack of rice cakes) is a famous Japanese food that is used for celebrating a new year.

Snuke has N rice cakes to create a Mirror Rice Cake. The weight of the i-th rice cake is a_i. He wants to create a Mirror Rice Cake by choosing some of these rice cakes and stacking them in some order. Additionally, it must satisfy the following constraint: for each rice cake in the stack, the total weight of all rice cakes above it must be strictly smaller than its own weight.

Compute the maximum possible number of rice cakes he can use to create a Mirror Rice Cake.

Input

First line of the input contains one integer N ($1 \leq N \leq 1000$). Each of next N lines contains weight a_i of the i-th rice cake ($1 \leq a_i \leq 10^9$).

Output

Print the maximum possible number of rice cakes Snuke can use to create a Mirror Rice Cake.

Example

standard input	standard output
5 3 20 5 8 6	3

Note

For example, stack three rice cakes of sizes $3, 5, 20$ from top to bottom.

Problem F. Number Cards

Input file:	*standard input*
Output file:	*standard output*
Time limit:	2 seconds
Memory limit:	512 mebibytes

Snuke has N cards with numbers. The i-th card contains positive integer a_i, and the color of this card is c_i (in this problem, we represent colors by integers).

Snuke has the following hypothesis about the coloring scheme of these cards:

- Cards with $1 \le a_i \le M$ are colored by the same color.
- Cards with $M+1 \le a_i \le 2M$ are colored by the same color, and this color is different from the color used for $1 \le a_i \le M$.
- Cards with $2M+1 \le a_i \le 3M$ are colored by the same color, and this color is different from the colors used for $1 \le a_i \le 2M$.
- Cards with $3M+1 \le a_i \le 4M$ are colored by the same color, and this color is different from the colors used for $1 \le a_i \le 3M$.
- and so on.

How many positive integers M are consistent with all the cards he has? If the number of possibilities of M is infinite, print -1.

Input

First line of the input contains one integer N ($1 \le N \le 20$). Each of next N lines contains two integers a_i and c_i — number and color of one of Snuke's cards, respectively ($1 \le a_i \le 10^9$, $1 \le c_i \le 20$). It is guaranteed that the sequence a_i is strictly increasing.

Output

Print the answer in a single line.

Examples

standard input	standard output
4 27 2 2000 4 2015 4 2100 1	277
3 1 1 2 2 3 1	0

Problem H. Random Walk

Input file:	*standard input*
Output file:	*standard output*
Time limit:	3.5 seconds
Memory limit:	512 mebibytes

There is an infinitely large 2-dimensional square grid. The coordinates on this grid are represented by a pair of integers (i, j).

Snuke wants to do a random walk. He starts from $(0, 0)$ and makes N steps. When he is at (i, j), his position after the next step will be one of $(i-1, j)$, $(i, j-1)$, $(i, j+1)$, and $(i+1, j)$. Each of these possibilities will happen with probability $\frac{1}{4}$.

Let E be the expected number of visited cells during the random walk. Compute the value $E \times 4^N$ modulo M (this value is guaranteed to be an integer). Note that $(0, 0)$ is always considered visited.

Input

Input consists of two integers N and M ($1 \leq N \leq 5000$, $10^9 \leq M \leq 2 \times 10^9$).

Output

Print the answer in a single line.

Examples

standard input	standard output
2 1000000007	44
2015 2000000000	1892319232

Problem I. Robots

Input file:	*standard input*
Output file:	*standard output*
Time limit:	2 seconds
Memory limit:	512 mebibytes

Snuke has N robots. They are numbered 1 through N. Initially, the robot i is placed at (x_i, y_i), and the direction the robot is initially facing is d_i. Here, d_i is one of 'U', 'D', 'L', and 'R': they represent the y-plus direction, the y-minus direction, the x-minus direction, and the x-plus direction, respectively. Robots and Snuke are considered points on the plane.

Initially, no robots are moving. However, when a robot is touched by something (Snuke or another robot), it will immediately start moving in the direction it is facing with unit speed. These robots are made of strange material and they can pass through other robots. Once a robot starts moving, it keeps moving no matter what happens; even if it touches another robot, it won't change its direction and speed.

Snuke touched the robot 1 at time 0. Compute the coordinates of each robot at time T.

Input

First line of input contains two integers N and T ($1 \le N \le 10^5$, $0 \le T \le 10^{18}$). The i-th of next N lines contains two integers x_i and y_i and letter d_i — initial coordinates and direction of i-th robot ($0 \le x_i, y_i \le 10^9$, d_i is one of the following characters: 'U', 'D', 'L', 'R'). At time 0, no two robots are at the same position.

Output

Print N lines. In the i-th line, print the coordinates of the robot i at time T.

Example

standard input	standard output
5 10 1 0 U 3 1 U 1 2 R 1 1 L 0 1 R	1 10 3 6 9 2 -8 1 8 1

Problem J. Ropes

Input file:	*standard input*
Output file:	*standard output*
Time limit:	2 seconds
Memory limit:	512 mebibytes

N persons are sleeping. They are numbered 1 through N. Snuke wants to connect them using $N - 1$ ropes!

- The two ends of each rope must be attached to two distinct persons. These two persons will be directly connected by a rope.
- All persons must be connected by ropes directly or indirectly.
- Exactly a_i ropes must be attached to the person i.

Compute the number of ways to connect the persons while satisfying all conditions above, modulo $10^9 + 7$. Two ways are considered different if there is a pair of persons which are directly connected by a rope in one of the ways but not in the other one.

Input

First line of the input contains one integer N ($2 \le N \le 10^5$). The i-th of next N lines contains one integer a_i — number of ropes which must be attached to i-th person ($1 \le a_i \le 3$).

Output

Print the answer in a single line.

Example

standard input	standard output
9 1 3 2 1 3 1 2 1 2	1260

Problem L. String Modification

Input file:	*standard input*
Output file:	*standard output*
Time limit:	2 seconds
Memory limit:	512 mebibytes

Snuke received a string s as a new year present. Determine if he can convert it to his favorite string, t, by repeating the following operation zero or more times.

Operation: Choose a character from s, and insert another character right after the chosen character. The inserted character must be different from the chosen character.

For example, he can convert "`abca`" to "`adbca`" in a single operation by choosing the first '`a`' and inserting a '`d`' right after it. However, he can't convert "`abca`" to "`aabca`" in a similar way.

Input

First line of the input contains string s, second line contains string t. Both strings are composed of lowercase English letters, $1 \leq |s| \leq |t| \leq 5000$.

Output

Print "`Yes`" in case when Snuke can convert s to t, or "`No`" otherwise.

Examples

standard input	standard output
snuke snukent	Yes
snuke ssnuke	No

Problem N. Soccer Match

Input file:	*standard input*
Output file:	*standard output*
Time limit:	2 seconds
Memory limit:	512 mebibytes

In a football match, a team either earns a win, tie or loss.

Winner takes 3 points, for a tie teams are awarded with 1 point each, and loss is worth 0 points.

In the old book about the history of football, you found information that at first season in the league your favourite team played N games and earned K points.

You wonder about numbers of wins, draws and loses it can have, so write a program which calculates all possible outcomes.

Input

Input file contains two space separated integers: g $(0 < g \leq 100)$, and p $(0 \leq p \leq 300)$, representing the number of games played and the total points earned by the team, respectively.

It is guaranteed that there is at least one possible combination of wins, ties and losses that is consistent with the given information.

Output

Print list of possible outcomes, one on the line. Each outcome consists of three space-separated integers w, t and l — number of wins, number of ties, and number of loses. Outcomes must be printed by descending order of wins.

Example

standard input	standard output
6 10	3 1 2 2 4 0