,

Problem A. Cake Distribution

Input file:	`standard input`
Output file:	`standard output`
Time limit:	15 seconds
Memory limit:	256 megabytes

You are about to have a birthday and you would like to prepare a birthday cake that weighs anywhere between 1 and 10^{18} grams inclusive. You know that there will be A, B or C guests at the party. You would like to cut this cake into pieces such that:

- The weight of each piece is a positive integer number of grams.
- No matter how many guests arrive, the pieces can be distributed between the guests in such a way that each guest gets the same amount of cake. Note that some guests might get more than one piece.

You don't want to spend too much time cutting the cake, so you would like to have at most 5,000 pieces. Let's do it!

Input

The only line of input contains the 3 numbers A, B, C, all positive integers not more than 1000.

Output

On the first line, output one number K, the number of pieces. On each of the next K lines, output a description of each piece, consisting of 4 numbers, w_i, a_i, b_i, c_i, where w_i is the weight of the piece in grams and a_i, b_i, c_i are indices of the person who will get this piece if A, B, or C guests arrive, respectively. The indices should satisfy $1 \le a_i \le A$, $1 \le b_i \le B$, $1 \le c_i \le C$. The sum of all w_is must be less than or equal to 10^{18}.

Example

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

Problem B. Favourite Number

Input file:	`standard input`
Output file:	`standard output`
Time limit:	15 seconds
Memory limit:	256 megabytes

Volodymyr's favourite number is A and it has an odd number of positive divisors. When you add K to this number, the resulting sum also has an odd number of positive divisors. Given K, find all possible values of A, Volodymyr's favourite number.

Input

The only line of input contains K, a positive integer not exceeding 10^9.

Output

On the first line, print the number of possible values of A. On the second line, print all possible values of A in ascending order, separated by a single space.

Examples

standard input	standard output
1	0
2	1 -1

Note

Note that since zero has infinitely many divisors, it cannot be classified as a number with an odd number of divisors.

Problem C. Pawn's Revenge

Input file:	`standard input`
Output file:	`standard output`
Time limit:	15 seconds
Memory limit:	256 megabytes

You are playing a special chess game against a professor and you've almost lost: the only piece you have left is a king. The professor just left the room to take a call, so you decided to cheat a little bit and put some extra pawns on the board so that each of opponent's pieces is under attack. As a reminder, a pawn attacks the two diagonally adjacent squares to the upper-left and upper-right of itself and a king attacks the eight adjacent squares (including the diagonally adjacent ones). You cannot put one piece on top of another piece. What is the smallest number of pawns you need to accomplish this task?

Input

The first row contains N, the dimension of the board, with $8 \leq N \leq 1000$. The game is played on an N by N chessboard. The next N lines have N symbols each describing the board. The symbol `-` means that the square is empty, `*` denotes a professor's piece and `K` denotes your king. Your pawns move upward (i.e. towards rows that appear earlier in the input).

Output

Output a line containing one number, the smallest number of pawns needed to attack all of the professor's chess pieces, or -1 if it is impossible to do so.

Example

standard input	standard output
8 -*-*---- -------- -------- -------- -----*K- -------- --*----- --------	2

,

Problem D. Deliveries

Input file:	`standard input`
Output file:	`standard output`
Time limit:	15 seconds
Memory limit:	256 megabytes

Sam is working as a truck driver and is frequently delivering goods between various warehouses. There are N warehouses in total. Some warehouses are connected, and there is exactly one way to get from each warehouse to each other warehouse, possibly passing several other warehouses along the way.

To make his trips, Sam will be using various battery-powered trucks. Each truck has a folding solar panel that can be used to charge the battery, but only when the truck is stopped. While traveling from one warehouse to another, Sam can stop at any point and wait a bit to recharge the battery. When the battery is out of charge, he can't move. Sam must also stop at every warehouse he passes through for security reasons. When he stops at a warehouse, he can also recharge the battery.

Sam doesn't like to stop too much, so he asked you to find the smallest number of stops for each route he takes. This includes both stops made for recharging and stops at warehouses, including the starting and final warehouses (for loading and unloading).

Input

The first line contains two numbers N and Q, the number of warehouses and the number of routes Sam takes, with $1 \leq N, Q \leq 100,000$. The next $N-1$ lines describe roads between warehouses. Each line contains 3 numbers: U, V, D. U and V are indices of warehouses, with $1 \leq U, V \leq N$. D is the length of the road between warehouses U and V in kilometres, with $1 \leq D \leq 20,000$. The next Q lines describe the routes Sam takes. Each line contains 3 numbers: S, F, T. S and F are indices of the starting and final warehouses on the route, with $1 \leq S, F \leq N$ and $S \neq F$. T is the capacity of the battery of the truck used for this route, expressed as the distance that the truck can travel on a full charge (in kilometres). T satisfies $1 \leq T \leq 20,000$.

Output

For each of the Q routes, output one line containing the smallest number of stops that need to be made along the route.

,

Examples

standard input	standard output
7 5 1 2 1 2 3 2 2 4 3 4 5 4 4 6 5 4 7 6 3 7 2 2 6 1 5 7 3 1 4 1 3 7 1	7 9 5 5 12
4 3 1 2 5 2 3 10 3 4 20 1 4 20 1 4 10 1 4 5	4 5 8

,

Problem E. XOR Pairing

Input file:	`standard input`
Output file:	`standard output`
Time limit:	15 seconds
Memory limit:	256 megabytes

There are N stones with non-negative integers $x_0, \ldots, x_{N-1}$ written on them, where N is even. We are to partition the stones into $\frac{N}{2}$ pairs, with each stone belonging to exactly one pair. For each pair, we calculate x_i XOR x_j by writing each of x_i and x_j in binary, taking the exclusive-or of each pair of corresponding bits, and interpreting the resulting bits as a binary number. We sum the values x_i XOR x_j over all the pairs of stones. The task is to find the minimum possible value of this overall sum over all possible pairings of the given stones. Furthermore, you are to find the number of stone pairings that achieve this minimal overall sum. Two pairings are considered to be different if one has a pair of stones that the other one does not have. Two stones are considered to be different if they have different indices, even if the value written on them is the same. The order of the pairs and the order of stones inside each pair is not significant. Since the number of such pairings can be huge, output it modulo $10^9 + 7$.

Input

The first line contains one even number N, the number of stones, satisfying $2 \leq N \leq 74$. The second line contains the N numbers written on the stones, $x_0, \ldots, x_{N-1}$. Each number satisfies $0 \leq x_i \leq 1000$.

Output

Output one line containing two numbers separated by a space: the smallest possible overall sum of the pairs x_i XOR x_j over all pairings, and the number of pairings that achieve this minimum sum modulo $10^9 + 7$.

Example

standard input	standard output
6 7 14 17 4 2 1	31 3