Problem A. Constructiveforces

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

You are given three integers $0 \le k \le m < n$. Construct binary string s of length n (s must consist only of 0s and 1s), such that in each substring of length m there are at most k ones and in each substring of length $m + 1$ there are at least k ones.

One can prove, that such a string always exists.

Input

The only line of input contains three integers n, m and k ($0 \le k \le m < n \le 100\,000$, $0 < m$).

Output

Output binary string of length n, such that in each substring of length m there are at most k ones and in each substring of length $m + 1$ there are at least k ones.

If there are many solutions, you can output any of them.

Examples

standard input	standard output
`4 2 1`	`0100`
`5 4 4`	`11111`

Problem B. Link Cut Cactus

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

You are given an undirected graph. Every vertex of the graph is labeled with some integer between 1 and 3.

Determine if there is a simple non-cyclic path (i. e., path, which doesn't visit any vertex twice), such that the sum of all labels in the path is divisible by 4.

Input

The first line contains one integer t ($1 \le t \le 10^6$), the number of test cases.

A description of each test case starts with two integers n and m ($1 \le n, m \le 10^6$), the number of vertices and edges in the graph.

The next line contains n integers $a_1, \ldots, a_n$ ($1 \le a_i \le 3$), the labels on corresponding vertices.

Each of the next m lines contains integers u and v ($1 \le u, v \le n$, $u \ne v$), describing an edge. There are no multiple edges.

It's guaranteed that both sum of n and m is at most 10^6.

Output

For each test case output "YES" (without quotes), if there is such path. Otherwise, output "NO" (without quotes).

Example

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

Problem C. Graph Mex Function

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

Consider a **directed** graph $G = (V, A)$ and define $N^+(v)$ the set of all vertices u such that $(v, u) \in A$.

The function $F\colon V \to \mathbb{N} \cup \{ 0 \}$ called graph mex function (GMF) if for any vertex v

$$F(v) = \mathrm{MEX}(\{ F(u) \mid u \in N^+(v) \}).$$

Note that $\mathrm{MEX}(S)$ is equal to the minimum non-negative integer value missing in the set S (i. e., **m**inimum **ex**cluded from S). For any directed acyclic graph (DAG) there is an unique GMF exists.

Find the GMF for a given DAG G.

Input

The first input line contains two integers n and m ($2 \le n, m \le 100\,000$), the number of vertices and arcs in the graph.

The ith of next m lines contains two integers u_i and v_i ($1 \le u_i, v_i \le n$, $u_i \ne v_i$), defining the arc $(u_i, v_i) \in A$.

Output

Output n integers $F(1)$, $F(2)$, …, $F(n)$, separate values by spaces.

Example

standard input	standard output
6 9 2 1 3 1 4 1 4 3 5 2 5 4 6 3 6 4 6 5	0 1 1 2 0 3

Problem D. Jumps

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	256 megabytes

There is a line with n cells numbered from left to right starting with 1. Uncle Bogdan is currently staying in the cell number 1, Papich is waiting in the cell number n. Uncle Bogdan is going to do jumps: for each cell number i, $1 \le i < n$, he can jump to any cell in the range $[l_i, r_i]$. It is guaranteed that $i < l_i \le r_i \le n$. As you can see, eventually, Uncle Bogdan will reach Papich. Papich wants to stream everlasting summer alone. That's why he will try to block exactly one cell. It means Uncle Bogdan will not be able to jump into a blocked cell (though he can jump over). Papich will not be able to block any of the cell number 1 and n.

You are asked to calculate the number of cells, such that blocking any of these cells Papich makes Uncle Bogdan unable to reach Papich.

Input

The first line contains one integer t ($1 \le t \le 5 \cdot 10^5$), the number of test cases.

Description of each test case starts with one integer n ($2 \le n \le 10^6$), the number of cells.

The ith of the next $n-1$ lines contains two integers l_i, r_i ($i < l_i \le r_i \le n$), left and right bounds of range to which Uncle Bogdan can jump from the cell number i.

It is guaranteed that the sum of n over all test cases doesn't exceed 10^6.

Output

For each test case output one integer, the desired number of "disabling" cells.

Example

standard input	standard output
2 7 4 5 3 7 4 7 5 5 6 6 7 7 2 2 2	2 0

Problem E. Positive Thinking

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

You decide to change and start to think positively. You begin with an array of integers (if you think that it's the strangest legend that you've ever read, you should read some other problems in this contest). In one operation you can choose any number and either decrease or increase it by 1. Calculate minimum number of operations to make product of all numbers in this array positive.

Input

The first line contains one integer t $(1 \le t \le 2 \cdot 10^5)$ — number of testcases.

Description of each testcase starts with one integer n $(1 \le n \le 2 \cdot 10^5)$ — length of an array.

Next line contains n integers $a_1, \ldots, a_n$ $(-10^9 \le a_i \le 10^9)$ — array a.

It's guaranteed that sum of n over all testcases doesn't exceed $2 \cdot 10^5$.

Output

For each testcase print one integer — minimum number of operations to make product of all numbers in this array positive.

Example

standard input	standard output
3 3 -3 -2 -2 9 9 9 8 2 4 4 3 5 3 3 0 0 0	3 0 3

Problem F. 3D-chans are not needed

Input file:	`standard input`
Output file:	`standard output`
Time limit:	8 seconds
Memory limit:	256 megabytes

Consider a 4-dimensional lattice cube $[1, n]^4$ and all lattice points within it, i. e., for which the coordinates range between (and including) 1 and n. (Namely, points (x_1, x_2, x_3, x_4), such that $1 \le x_i \le n$ and $x_i \in \mathbb{Z}$ for all $i = 1, 2, 3, 4$.) Also, you are given m constraints. The i-th of them is the following: the Manhattan distance to the lattice point $(y_1^i, y_2^i, y_3^i, y_4^i)$ in the cube is not greater then r_i.

Calculate the number of lattice points in the cube, which satisfy all given constraints. Since the problem doesn't look bad without this, calculate the answer modulo 998244353.

The Manhattan distance between two points $\mathbf{a} = (a_1, a_2, a_3, a_4)$ and $\mathbf{b} = (b_1, b_2, b_3, b_4)$ is defined as

$$d(\mathbf{a}, \mathbf{b}) = |a_1 - b_1| + |a_2 - b_2| + |a_3 - b_3| + |a_4 - b_4|.$$

Input

The first line contains two integers m, n ($1 \le m, n \le 2 \cdot 10^5$), the number of constraints, and the upper bound for coordinates.

The i-th of the next m lines contains 5 integers y_1^i, y_2^i, y_3^i, y_4^i and r_i ($1 \le y_j^i \le n$, $0 \le r_i \le 4n$), the description of the i-th constraint.

Since the jury is evil, you are not guaranteed anything. There may be equal points, there may be constraints that are satisfied by points outside of the cube, the answer may be zero, the answer may be nonzero. Good luck!

Output

Output the number of points modulo 998244353.

Examples

standard input	standard output
1 2 1 1 2 1 1	5
2 9 1 4 8 8 8 1 3 3 7 7	366
4 200000 89720 90321 165797 98663 492130 160473 118956 106147 16203 280661 84981 6679 185061 19011 527856 125272 154317 149362 149529 204616	231512004

Problem G. Modular Hashing

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

Given an integer n.

Let's define function $f_n(x)$ as vector function with values from $\mathbb{Z}^{n-1}$:

$$f_n(x) = \begin{bmatrix} x \bmod 2 \\ x \bmod 3 \\ x \bmod 4 \\ \dots \\ x \bmod n \end{bmatrix}.$$

If one calculate values $f_n(1)$, $f_n(2)$, $\dots f_n(m)$ how many distinct vectors will be?

Input

The only line contains two integers n and m ($2 \le n \le 100$, $1 \le m \le 10^{18}$).

Output

Output required number of distinct vectors.

Examples

standard input	standard output
2 10	2
5 3	3
100 100	100

Problem H. Something Random Modulo Something Random

Input file:	`standard input`
Output file:	`standard output`
Time limit:	3 seconds
Memory limit:	256 megabytes

You will generate $2n$ random integers. Each integer is generated uniformly at random in range $[0, 2l]$. Calculate probability that sum of first n numbers, that you generated, is bigger than sum of n last numbers, that you generated. Since we live in 2021, calculate probability by given prime modulo.

Input

The only line will contain three integers n, l and MOD ($1 \le n \le 2000, 1 \le l \le 2000, 10^8 \le MOD \le 10^9$) — parameters for generation and given MOD. It's guaranteed that MOD will be prime.

Output

As the answer can be represented as a rational number $\frac{p}{q}$ for coprime p and q, you need to find the value of $p \cdot q^{-1} \ mod \ MOD$. You can prove that $q \ mod \ MOD \neq 0$.

Examples

standard input	standard output
1 1 998244353	332748118
3 2 148822829	36403255

Note

For the first sample, out of 9 ways equiprobable ways to generate numbers, only 3 satisfy statement(namely, $[2, 0]$, $[2, 1]$, $[1, 0]$). Therefore, answer is $1 \cdot 3^{-1} \ mod \ 998244353 = 332748118$.

Problem I. Destiny

Input file:	`standard input`
Output file:	`standard output`
Time limit:	3 seconds
Memory limit:	512 megabytes

Once, Fedos found in the left pocket an array a consisting of n integers, and in the right pocket q queries. Queries are of the following 3 types:

1. Add d to all a_i for $l \le i \le r$.
2. Replace all a_i in $l \le i \le r$ by prefix minimum on this segment. In other words, do:
   - $a[l] = a[l]$
   - $a[l+1] = min(a[l], a[l+1])$
   - ...
   - $a[r] = min(a[l], \ldots, a[r])$

   All updates should be done simultaneously.
3. Calculate minimum on the segment $[l, r]$.

If there are queries, they should be answered. Your task is to calculate answers to all queries of the third type.

Input

First line contains one number n $(1 \le n \le 5 \cdot 10^5)$ — length of an array.

Second line contains n numbers a_i $(1 \le a_i \le 10^9)$.

Third line contains one number q $(1 \le q \le 5 \cdot 10^5)$ — number of queries.

Next q lines consist of description of queries. Each line starts from number t ($t = 1$ or $t = 2$ or $t = 3$) — type of a query. If $t = 1$, then numbers l, r, d $(1 \le l \le r \le n$, $1 \le d \le 1000)$ follow — segment and which value should be added. If $t = 2$ or $t = 3$, then numbers $l, r (1 \le l \le r \le n)$ follow — segment for the query of second or third type.

Output

For each query of third type output answer for it.

Example

standard input	standard output
5 0 0 0 4 2 12 1 2 4 1 1 2 4 1 3 3 4 3 2 4 2 2 3 2 1 4 1 1 3 1 2 1 2 2 4 5 1 2 3 2 1 4 4 1 3 2 5	2 2 0

Problem J. Sad Problem

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

Petya had a sequence of n integers $a_1, a_2, \ldots, a_n$. Once he calculated for each i from 1 to n a number of such j $(1 \le j \le i)$ that $a_j < a_i$ and number of such k $(i \le k \le n)$ that $a_k < a_i$ and defined them as l_i and r_i respectively.

Someone shuffled all triples (a_i, l_i, r_i), and that made Petya sad. You are Petya's friend and you want to help him to restore the original sequence.

Input

The first line contains one integer t $(1 \le t \le 10^3)$, the number of test cases.

Description of each test case starts with one integer n $(2 \le n \le 100)$, the size of the sequence.

The ith of the next n lines contains three integers a_i, l_i and r_i $(1 \le a_i \le 1000,\ 0 \le l_i, r_i \le 1000)$.

Output

For each test case if it's possible to restore the original sequence print "Yes" in the first line and the original sequence in the second line, or print "No" in a single line.

Example

standard input	standard output
6 3 2 0 1 3 1 1 1 0 0 5 2 1 0 3 1 1 1 0 0 5 2 2 4 3 0 3 1 0 0 2 1 1 1 0 0 5 2 1 1 3 2 1 1 0 0 3 3 0 1 0 0 2 1 0 0 1 0 1 2 1 0 0 1 0 1	Yes 2 3 1 Yes 1 3 5 2 4 Yes 1 2 1 Yes 1 2 3 1 3 No No

Problem K. Split Circles

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

Given two circles $(x - x_1)^2 + (y - y_1)^2 = r_1^2$ and $(x - x_2)^2 + (y - y_2)^2 = r_2^2$.

Find a line that splits the first circle into two parts with ratio of area $a_1 : b_1$ and the second one into two parts with ratio of area $a_2 : b_2$.

Input

Each test consists of multiple sets of input data. The first input line contains an integer t ($1 \le t \le 1000$), the number of input data sets. It is followed by a description of input data sets.

The first line of each test case contains five integers x_1, y_1, r_1, a_1 and b_1 ($-10 \le |x_1|, |y_1| \le 10$, $1 \le r_1 \le 10$, $1 \le a_1 \le b_1 \le 10$).

The second line of the test case contains five integers x_2, y_2, r_2, a_2 and b_2 ($-10 \le |x_2|, |y_2| \le 10$, $1 \le r_2 \le 10$, $1 \le a_2 \le b_2 \le 10$).

Output

For each test case output a linear equation in the form $ax + by + c = 0$ by giving three numbers a, b and c ($a^2 + b^2 \ge \frac{1}{2}$). If no such line exists output three zeros.

Your answer will be considered correct if either an absolute or relative error of the requested ratios does not exceed 10^{-6} relatively to the expected ones. This means that if the expected ratio is x and your answer is y, then your answer is considered correct if and only if $\frac{|x-y|}{\max\{1,|x|,|y|\}} \le 10^{-6}$.

Examples

standard input	standard output
3 0 0 3 1 1 0 2 1 1 3 3 4 5 2 3 7 5 1 2 9 -4 -4 4 4 4 4 4 4 4 4	-0.9793883314 -0.2019863766 0.0000000000 0.3040744006 -0.9526482871 2.1096889776 -0.7071067812 0.7071067812 0.0000000000
1 0 0 5 1 1 0 0 3 2 3	0.0000000000 0.0000000000 0.0000000000

Problem L. SqrtSqrtSqrt

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

Define $F(n) = \sum_{x=1}^{n} \lfloor \sqrt{x} \rfloor$. Your task is to calculate $F(n)$ for a given n.

Input

In the first line you are given integer t $(1 \le t \le 10^5)$ — the number of test cases.

Desctiption of each test case consists of one line with one integer n $(1 \le n \le 10^{12})$.

Output

For each test case print the answer.

Example

standard input	standard output
10 1 2 3 4 5 6 7 8 9 8822814	1 2 3 5 7 9 11 13 16 17466658605

Problem M. There could be your problem name

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

There could be your problem legend.

You are given natural number n. You should calculate minimum amount of integers, each of them should consist of the same digits, which sum up to n.

For example, you can represent $23 = 22 + 1$, $199 = 99 + 99 + 1$.

Input

In the only line you are given natural number n ($1 \le n < 10^{200}$).

Output

Output minimum amount of consisting of the same digits integers which sum up to n.

Examples

standard input	standard output
1	1
228	2
22832213371529156615881703148811142	30

Problem N. Let's sum

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

Let's denote function $f(n, a)$, which receives number n and set of digits a and returns smallest integer, which is not less than n and which consists only of the digits from set a. If there is no such number, returned value is 0. For example, $f(42, \{2, 4, 6, 8\}) = 42$, $f(33, \{2, 4, 6, 8\}) = 42$, $f(44, \{4\}) = 44$, $f(42, \{0\}) = 0$.

Function $s(n, k)$ receives numbers n and k and returns sum $f(n, a)$ over all sets a, which are subsets of set consisting of the first k digits, so $s(n, k) = \sum_{a \subset \{0,1,\ldots k-1\}} f(n, a)$. For example, $s(n, 2) = f(n, \{\}) + f(n, \{0\}) + f(n, \{1\}) + f(n, \{0, 1\})$.

For given integers l, r and k calculate $\sum_{l \le i \le r} s(i, k)$ modulo 998244353.

Input

The only line of input contains three integers l, r and k ($1 \le l \le r \le 10^{18}$, $2 \le k \le 10$).

Output

Print value of the given sum.

Examples

standard input	standard output
`8 9 2`	`42`
`54 675 3`	`4200000`

Note

In the first sample test $s(8, 2) = f(8, \{\}) + f(8, \{0\}) + f(8, \{1\}) + f(8, \{0, 1\}) = 0 + 0 + 11 + 10 = 21$, $s(9, 2) = 21$ and sum is equal to $s(8, 2) + s(9, 2) = 21 + 21 = 42$.