

Problem A. Assembly

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

Consider a programming language called assembly. In this language, there are 4 registers named A , B , C and D which are initially equal to 0 and can each store an integer between -2^{31} and $2^{31} - 1$ inclusive. A program written in this language consists of several lines, each containing an instruction. The execution of a program starts from the first line and each line is followed by the next one, except in the case of a **GOTO** instruction (described below). The available instructions are:

- **INPUT X** — Reads one number from the input and puts it into the register X .
- **OUTPUT X** — Outputs the value stored in the register X .
- **SET X TO Y** — Puts the value Y into the register X .
- **INCREASE X BY Y** — Increases the value of the register X by Y .
- **DECREASE X BY Y** — Decreases the value of the register X by Y .
- **GOTO Z** — If $Z = -1$, terminates the program, otherwise the execution jumps to the Z^{th} line.
- **IF $Y_1 \star Y_2$ GOTO Z_1 ELSE GOTO Z_2** — If the value of expression $Y_1 \star Y_2$ is true, it is equivalent to **GOTO Z_1** , otherwise it is equivalent to **GOTO Z_2** .

Here:

- X is a register name.
- Each of Y , Y_1 and Y_2 is a register name or an integer.
- Each of Z , Z_1 and Z_2 is either -1 or an integer between 1 and the number of lines in the program.
- $\star$ is one of the operators $==$, $!=$, $<$, $<=$, $>$ or $>=$.

Given two integers n and k , your goal is to write an assembly program that does the following:

1. Read a list of n integers $a_1, a_2, \dots, a_n$ from the input ($1 \leq a_i \leq 500$).
2. Find the k biggest prime numbers in the list. You may assume that there are at least k prime numbers in the list.
3. Output the product of these k numbers, modulo 2023.

Note that the assembly program should not read the values of n and k .

Input

The input contains two integers n and k ($1 \leq n \leq 100$, $1 \leq k \leq 4$).

Output

On the first line, print one integer m ($1 \leq m \leq 125$) — the number of lines in your program.

Then print m more lines. The i^{th} line should consist of the integer i followed by the i^{th} instruction of your assembly program.

You will receive the Wrong Answer verdict if the program you output:

- gives a wrong answer, or
- has more than 125 lines, or
- contains a line that does not follow the correct syntax of an available instruction, or
- executes more than 10^8 instructions before terminating, or
- stores a value outside the range $[-2^{31}, 2^{31} - 1]$ into a register.

Example

standard input	standard output
1 1	2 1 INPUT A 2 OUTPUT A

Problem B. Beautiful XOR Problem

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

You are given a sequence of n integers $a_1, a_2, \dots, a_n$ and a positive integer k .

Count the number of subsequences $b_1, b_2, \dots, b_l$ of a with length l such that $b_1 \oplus b_2 \oplus \dots \oplus b_l = 0$ (where $\oplus$ denotes bitwise XOR) and l is divisible by k .

Here, b is a subsequence of a if b can be obtained from a by the deletion of several (possibly, zero or all) elements.

Input

The first line of input contains two integers n and k ($1 \leq n \leq 10^6$, $1 \leq k \leq 20$).

The second line of input contains n integers $a_1, a_2, \dots, a_n$ ($0 \leq a_i \leq 10^6$).

Output

Print an integer — the number of subsequences satisfying the constraints. Since the answer can be very large, print it modulo 998244353.

Example

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

Problem C. Convolution

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

Given two sequences $a_0, a_1, \dots, a_{n-1}$ and $b_0, b_1, \dots, b_{m-1}$, the sequence $c_0, c_1, \dots, c_{n+m-2}$ is defined as follows:

$$c_k = \sum_{i+j=k} a_i b_j.$$

Your task is to find the value of

$$\sum_{k=0}^{n+m-2} c_k.$$

Input

The first line of input contains two integers n and m ($1 \leq n, m \leq 10^5$).

The second line of input contains n integers $a_0, a_1, \dots, a_{n-1}$ ($0 \leq a_i \leq 10^4$).

The third line of input contains m integers $b_0, b_1, \dots, b_{m-1}$ ($0 \leq b_i \leq 10^4$).

Output

Print one integer — the value described above.

Example

standard input	standard output
4 5 2 1 3 7 4 2 0 6 9	273

Problem D. DAG Probability

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

Three integers n , a and b are given.

A directed graph with n vertices numbered 1 to n and $\frac{n(n-1)}{2}$ edges is created in the following way: for every pair of integers u and v where $1 \leq u < v \leq n$, add an edge which is directed from u to v with probability $\frac{a}{b}$ or from v to u otherwise. All edges are chosen independently.

Find the probability that the created graph doesn't contain any cycle.

Input

The input contains three integers n , a and b ($1 \leq n \leq 10^6$, $1 \leq a < b < 998244353$).

Output

Output the probability modulo 998244353.

Formally, let $M = 998244353$. It can be shown that the answer can be expressed as an irreducible fraction $\frac{p}{q}$, where p and q are integers and $q \not\equiv 0 \pmod{M}$. Output the integer equal to $p \cdot q^{-1} \pmod{M}$. In other words, output an integer x such that $0 \leq x < M$ and $x \cdot q \equiv p \pmod{M}$.

Examples

standard input	standard output
15 1 3	527217576
410977 606205472 662422794	694202137

Problem E. Erase the Primes

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

A positive integer is called *good* if it is divisible by at least two different prime numbers.

For a sequence of positive integers, you may apply zero or more operations on it. Each operation consists of removing any two elements from the sequence and appending their product to the end. The *cost* of a sequence is defined as the minimum number of operations required to make all of its elements good, or -1 if it's impossible to do so.

Given an sequence of n integers $a_1, a_2, \dots, a_n$, you will be asked q queries about it. In each query, two indices l and r are given, and you should find the cost of the subsequence $a_l, a_{l+1}, \dots, a_{r-1}, a_r$.

Input

The first line of input contains two integers n and q ($1 \leq n, q \leq 2 \times 10^5$).

The second line contains n integers $a_1, a_2, \dots, a_n$ ($1 \leq a_i \leq 10^7$).

Then q lines follow. Each of these lines contains two integers l and r ($1 \leq l \leq r \leq n$) describing a query.

Output

For each query, print one number — the answer to the query.

Example

standard input	standard output
8 5	-1
2 8 3 5 1 9 42 3	2
5 5	2
1 3	4
6 8	3
1 8	
2 7	

Problem F. Forest Game

Input file: **standard input**
Output file: **standard output**
Time limit: 5 seconds
Memory limit: 256 megabytes

Consider a game that Alice and Bob play on an undirected graph.

First, Bob chooses k vertices uniformly at random and marks them as special. Alice does not know which vertices are special. Then Alice chooses a walk consisting of at most m edges starting from vertex x . If one of the vertices visited by the walk is special, Alice wins, otherwise Bob wins. In particular, if vertex x is special, Alice wins.

Given a graph with n vertices and no edges initially, your task is to process two types of queries:

- 1 $u\ v$ – add an edge between vertices u and v . It is guaranteed that v was not reachable from u before the query.
- 2 $x\ m\ k$ – find the probability that Alice wins the game with the parameters x , m and k on the current graph if she plays optimally.

Input

The first line of input contains two integers n and q ($1 \leq n \leq 2 \times 10^5, 1 \leq q \leq 5 \times 10^5$) – the number of vertices and queries.

The next q lines contain the descriptions of queries. The queries are given in an encoded format. Each query is a line starting with an integer t ($1 \leq t \leq 2$) – the type of the query.

- If $t = 1$, then two integers $u \oplus last$ and $v \oplus last$ follow ($1 \leq u, v \leq n, u \neq v$).
- If $t = 2$, then three integers $x \oplus last$, $m \oplus last$ and $k \oplus last$ follow ($1 \leq x \leq n, 1 \leq m \leq 10^9, 1 \leq k \leq n$).

Here $last$ denotes the answer to the last type 2 query or 0 if there were no type 2 queries previously, and $\oplus$ denotes the bitwise XOR operation.

Output

For each type 2 query, print the probability modulo 998244353.

Formally, let $M = 998244353$. It can be shown that the answer can be expressed as an irreducible fraction $\frac{p}{q}$, where p and q are integers and $q \not\equiv 0 \pmod{M}$. Output the integer equal to $p \cdot q^{-1} \pmod{M}$. In other words, output an integer x such that $0 \leq x < M$ and $x \cdot q \equiv p \pmod{M}$.

Example

standard input	standard output
4 6	499122177
2 2 2 2	1
1 499122176 499122179	249561089
1 499122179 499122181	
2 499122181 499122179 499122178	
1 2 5	
2 3 2 0	

Problem G. Guessing by Divisibility

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

There is a hidden integer $1 \leq n \leq 10^4$. You may ask questions in the form “Is n divisible by x ?” for some integer $1 \leq x \leq 10^4$.

Your goal is to determine the value of n by asking at most 1500 questions.

Interaction Protocol

To ask a question, print a line in the form “? x ” ($1 \leq x \leq 10^4$). After that, you should read a line containing a string “Yes” or “No” denoting the answer to the question.

When you have determined the value of n , print the answer in the format “! n ”.

Remember to flush the output after printing a line.

The interactor is not adaptive, which means that the value n is fixed beforehand and doesn’t change during interaction.

Example

standard input	standard output
	? 4
Yes	
	? 64
No	
	? 28
Yes	
	? 8
Yes	
	! 56

Problem H. Highest Median Walk

Input file: standard input
Output file: standard output
Time limit: 5 seconds
Memory limit: 256 megabytes

There is a weighted directed graph with n vertices and m edges. The i^{th} edge is directed from vertex u_i to vertex v_i and has weight w_i . Find the maximum possible median of a walk with length at least k .

A walk of length s is a sequence of (not necessarily distinct) edges $e_1, e_2, \dots, e_s$ for which there exists a sequence of vertices $p_1, p_2, \dots, p_{s+1}$ such that the edge e_j is directed from p_j to p_{j+1} for $j = 1, 2, \dots, s$. The median of a sequence with s elements is defined as its $\lceil \frac{s}{2} \rceil$ -th smallest element. (For example, the median of $[1, 3, 4, 2]$ is 2 and the median of $[2, 5, 4, 3, 1]$ is 3.) The median of a walk is the median of the sequence formed by the weights of its edges.

Input

The first line of input contains three integers n, m, k ($1 \leq n \leq 10^5$, $1 \leq m \leq 2 \times 10^5$, $1 \leq k \leq 50$).

Then m lines follow. The i^{th} of these lines contains three integers u_i, v_i, w_i ($1 \leq u_i, v_i \leq n$, $1 \leq w_i \leq 10^9$).

The graph may contain multiple edges or self-loops.

Output

If no such walks exist, print -1 . Otherwise, print one number — the maximum median of a walk with length at least k .

Example

standard input	standard output
7 7 4 2 1 3 7 2 5 1 7 4 7 5 6 6 4 9 5 2 3 2 3 9	5

Problem I. Incomplete Information Queries

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

There is a directed graph with n vertices and $\frac{n(n-1)}{2}$ edges. For each pair of different vertices x and y , there is exactly one edge that is directed from x to y or from y to x . You are not given the entire graph, but for each vertex i you know c_i , the number of vertices reachable from it.

m pairs of integers $(u_1, v_1), (u_2, v_2), \dots, (u_m, v_m)$ are given. You can apply zero or more operations to the graph. In each operation, you can choose a pair (u_i, v_i) and make the edge between u_i and v_i bidirectional. For two vertices a and b , define $f(a, b)$ as the minimum number of operations required to make vertex b reachable from vertex a . If it is impossible to make b reachable from a , then $f(a, b) = -1$.

There are q independent queries about the graph. Each query is in one of the forms:

1. **S** a b – You should find the **smallest** value of $f(a, b)$ over all possible graphs satisfying the given constraints.
2. **L** a b – You should find the **largest** value of $f(a, b)$ over all possible graphs satisfying the given constraints.

Input

The first line of input contains three integers n, m and q ($1 \leq n, m, q \leq 5 \times 10^5$).

The next line contains n integers $c_1, c_2, \dots, c_n$ ($1 \leq c_i \leq n$).

Then m lines follow. The i^{th} of these lines contains two integers u_i and v_i ($1 \leq u_i, v_i \leq n, u_i \neq v_i$).

Then q more lines follow. Each line contains a character t followed by two integers a and b describing a query ($t \in \{\text{S}, \text{L}\}, 1 \leq a, b \leq n$).

You may assume that there exists at least one graph satisfying the constraints.

Output

For each query print an integer — the answer to the query.

Example

standard input	standard output
5 3 3	-1
4 3 5 1 2	0
5 1	2
1 2	
2 4	
S 1 3	
L 1 2	
S 4 1	

Problem J. Joining Arrays

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

Define the *permutation scaling* of an array $[B_1, B_2, \dots, B_m]$ as the unique permutation P of $[1, 2, 3, \dots, m]$ such that for each pair of indices $1 \leq i < j \leq m$, $P_i < P_j$ if and only if $B_i \leq B_j$. For example, the *permutation scaling* of the array $[2, 1, 1, 5, 8, 3]$ is $[3, 1, 2, 5, 6, 4]$.

There are n sets of integers $S_1, S_2, \dots, S_n$. An array $[A_1, A_2, \dots, A_n]$ is formed by choosing one element from each set, that is, $A_i \in S_i$ for $i = 1, 2, \dots, n$.

How many ways are there to choose the elements such that the *permutation scaling* of A is lexicographically maximal?

Input

The first line of input contains an integer n ($1 \leq n \leq 5 \times 10^5$).

Then the descriptions of $S_1, S_2, \dots, S_n$ follow. The description of each set S_i contains two lines. The first line contains an integer k_i , the size of S_i ($1 \leq k_i \leq 5 \times 10^5$). The second line contains k_i distinct integers $s_1, s_2, \dots, s_{k_i}$, the elements of S_i ($1 \leq s_j \leq 10^9$).

It is guaranteed that $\sum_{i=1}^n k_i \leq 5 \times 10^5$.

Output

Print one integer — the number of ways. Since it can be very large, output it modulo 998244353.

Examples

standard input	standard output
4 5 7 11 3 10 1 3 8 6 4 1 15 3 1 5 99	13
4 2 2 1 3 5 1 6 1 3 3 6 5 7	3

Note

In the first example, the lexicographically maximal *permutation scaling* is $[3, 2, 4, 1]$ and there are 13 different ways to choose the array A : $[7, 4, 15, 1]$, $[7, 6, 15, 1]$, $[7, 6, 15, 5]$, $[10, 4, 15, 1]$, $[10, 6, 15, 1]$, $[10, 6, 15, 5]$, $[10, 8, 15, 1]$, $[10, 8, 15, 5]$, $[11, 6, 15, 1]$, $[11, 6, 15, 5]$, $[11, 8, 15, 1]$ and $[11, 8, 15, 5]$.

In the second example, the lexicographically maximal *permutation scaling* is $[2, 1, 3, 4]$ and there are 3 different ways to choose the array A : $[2, 1, 3, 5]$, $[2, 1, 3, 6]$ and $[2, 1, 3, 7]$.

Problem K. Kid's First Geometry Problem

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

There is a convex polygon A on the Cartesian plane. There are also k more convex polygons $B_1, B_2, \dots, B_k$. Initially, for each $i = 1, 2, \dots, k$, the interiors of A and B_i intersect each other.

Your goal is to make A not intersect any of $B_1, B_2, \dots, B_k$ (only boundaries touching is allowed). To achieve that, you can choose two real numbers dx and dy and translate the polygon A by the vector (dx, dy) . The cost of doing this is $|dx| + |dy|$.

Find the minimum cost to achieve the goal.

Input

The first line contains an integer n ($3 \leq n \leq 75000$), the number of vertices in A . Then n lines follow. Each line contains two integers x, y ($-10^8 \leq x, y \leq 10^8$) denoting the coordinates of a vertex of A .

The next line contains an integer k ($1 \leq k \leq 25000$). Then the descriptions of $B_1, B_2, \dots, B_k$ follow. The description of B_i starts with a line containing an integer m_i ($3 \leq m_i \leq 75000$), the number of vertices in B_i . Then m_i lines follow. Each line contains two integers x, y ($-10^8 \leq x, y \leq 10^8$) denoting the coordinates of a vertex of B_i .

The vertices of each polygon are given in counterclockwise order, and no three vertices of the same polygon are collinear.

It is guaranteed that $\sum_{i=1}^k m_i \leq 75000$.

Output

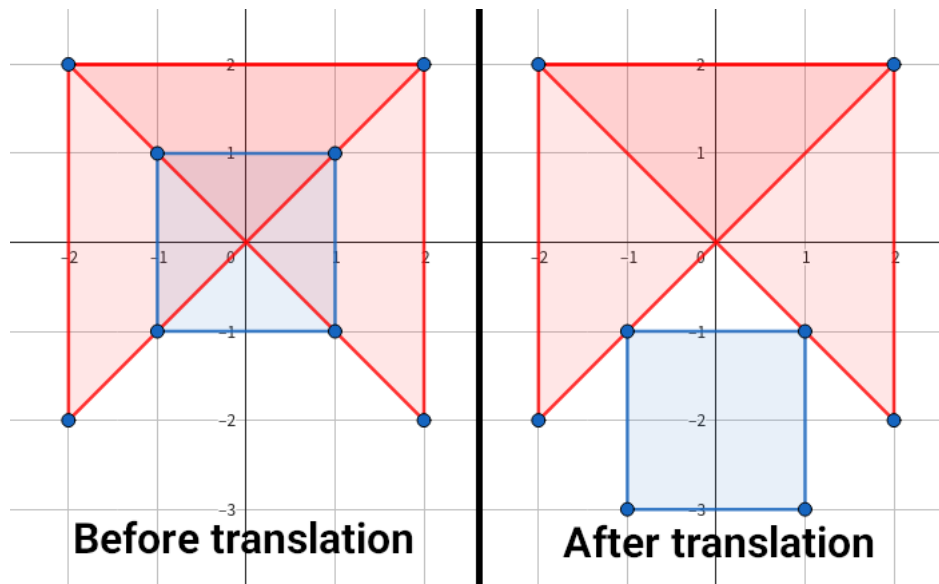
Print the minimum cost rounded up to the nearest integer. In other words, print the minimum integer that is not less than the answer.

Examples

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

Note

The first example is shown in the diagram below. You can choose $(dx, dy) = (0, -2)$. The minimum cost is 2.



In the second example, you can choose $(dx, dy) = (2, 0.8)$. The minimum cost is 2.8, so you should output 3.

Problem L. Lying Faces

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

There are n people numbered 1 to n , and each of them has a nose length which is a positive integer unknown to you. Each person is either a liar or a truth teller. Truth tellers always tell the truth and liars always lie. It is known that there is at least one liar and there are more truth tellers than liars.

Your goal is to find out all the liars. To do that, you are allowed ask at most 10001 queries. In each query, you choose two distinct indices $1 \leq a, b \leq n$, and ask the a^{th} person “What is the length of the b^{th} person’s nose?”. After answering the question, if the a^{th} person is a liar, their nose length increases by an unknown positive integer, which may be different every time.

Interaction Protocol

Initially, the jury program sends one integer n ($3 \leq n \leq 10000$) — the number of people.

After that, your program can submit queries to the jury program by printing the following line:

- `? a b` — ask the a^{th} person about the b^{th} person’s nose length ($1 \leq a, b \leq n, a \neq b$).

After each query, the jury prints an integer x on one line. Either:

- x is a positive integer, which is the reply from the a^{th} person. In this case you may assume that $1 \leq x \leq 10^9$.
- x is -1 , which means your query is incorrect or you have asked too many queries. After receiving -1 , your program should exit immediately to receive the Wrong Answer verdict.

To submit the answer, your program should send the output in the following format (and then exit immediately):

- `! k p1 p2 ... pk` — the number of liars and their indices in increasing order.

Remember to flush the output after printing a line.

The jury program can be adaptive, which means that the set of liars and the nose lengths may change during interaction, but it’s guaranteed that there always exists a valid initial configuration that produces an outcome that is consistent with all the queries.

Examples

standard input	standard output
5	? 3 4
42	! 2 2 4
3	? 1 2
420	? 3 2
420	! 1 2