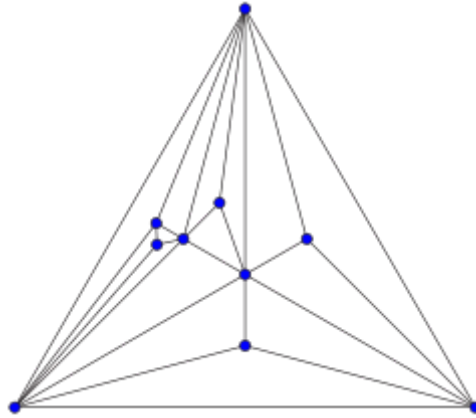


## Problem A. Apollonian Network

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

An Apollonian network is an undirected graph formed by recursively subdividing a triangle into three smaller triangles.



Yuhao Du has an Apollonian network with weighted edges. And he knows how to find a simple path with the largest possible sum of edge weights. Can you find it too?

### Input

The first line of the input contains one integer  $n$ : the number of vertices in Yuhao's Apollonian network ( $3 \leq n \leq 250$ ).

The next  $3(n - 2)$  lines contain a description of the edges of the graph. Each of these lines contains three integers  $a_i, b_i, c_i$ , describing an edge between vertices  $a_i$  and  $b_i$  with weight  $c_i$  ( $1 \leq a_i, b_i \leq n, a_i \neq b_i, 0 \leq c_i \leq 10^6$ ).

It is guaranteed that the given graph is an Apollonian network.

### Output

Output one integer: the largest sum of edge weights on a simple path in Yuhao's Apollonian network.

## Examples

standard input	standard output
3 1 2 1 2 3 1 3 1 2	3
10 1 2 4 2 3 4 3 1 3 6 1 3 6 2 3 6 3 4 4 6 4 4 3 4 4 2 3 5 1 3 5 6 3 5 2 4 10 1 4 10 3 3 10 6 3 7 1 4 7 10 4 7 6 3 8 1 3 8 3 4 8 10 4 9 3 4 9 8 3 9 10 3	35

## Note

In the first example, one of the optimal paths is  $2 \rightarrow 3 \rightarrow 1$ .

In the second example, one of the optimal paths is  $5 \rightarrow 2 \rightarrow 1 \rightarrow 7 \rightarrow 10 \rightarrow 8 \rightarrow 9 \rightarrow 3 \rightarrow 6 \rightarrow 4$ .

## Problem B. Bitwise Xor

Input file: *standard input*  
Output file: *standard output*  
Time limit: 2 seconds  
Memory limit: 1024 mebibytes

Zhong Ziqian got an integer array  $a_1, a_2, \dots, a_n$  and an integer  $x$  as birthday presents.

Every day after that, he tried to find a non-empty subsequence of this array  $1 \leq b_1 < b_2 < \dots < b_k \leq n$ , such that for all pairs  $(i, j)$  where  $1 \leq i < j \leq k$ , the inequality  $a_{b_i} \oplus a_{b_j} \geq x$  held. Here,  $\oplus$  is the bitwise exclusive-or operation.

Of course, every day he must find a different subsequence.

How many days can he do this without repeating himself? As this number may be very large, output it modulo 998 244 353.

### Input

The first line of the input contains two integers  $n$  and  $x$  ( $1 \leq n \leq 300\,000$ ,  $0 \leq x \leq 2^{60} - 1$ ). Here,  $n$  is the size of the array.

The next line contains  $n$  integers  $a_1, a_2, \dots, a_n$ : the array itself ( $0 \leq a_i \leq 2^{60} - 1$ ).

### Output

Output one integer: the number of subsequences of Ziqian's array such that bitwise xor of every pair of elements is at least  $x$ , modulo 998 244 353.

### Examples

standard input	standard output
3 0 0 1 2	7
3 2 0 1 2	5
3 3 0 1 2	4
7 4 11 5 5 8 3 1 3	35

### Note

In the first example, all  $2^3 - 1$  non-empty subsequences are suitable.

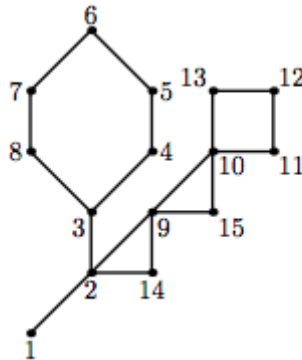
in the second example, two non-empty subsequences are not suitable, it is  $b = [1, 2]$  and  $b = [1, 2, 3]$ , that is because  $a_1 \oplus a_2 = 0 \oplus 1 = 1$  which is smaller than 2.

In the third example,  $b = [1], b = [2], b = [3], b = [2, 3]$  are suitable.

## Problem C. Counting Cactus

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

NEERC featured a number of problems about cactuses: **connected** undirected graphs in which every edge belongs to at most one simple cycle. Intuitively, a cactus is a generalization of a tree where some cycles are allowed. An example of a cactus from NEERC 2007 problem is given on the picture below.



Dreamoon has an undirected graph. Now he is wondering, how many subgraphs (subsets of edges) of his graph are cactuses? Can you help him find this value modulo 998 244 353?

### Input

The first line contains two integers  $n$  and  $m$ : the number of vertices and edges in the Dreamoon's graph ( $1 \leq n \leq 13$ ,  $0 \leq m \leq \frac{n(n-1)}{2}$ ).

The next  $m$  lines describe edges in the graph. The  $i$ -th of these lines contains two integers  $a_i$  and  $b_i$  ( $1 \leq a_i, b_i \leq n$ ,  $a_i \neq b_i$ ), denoting an edge between vertices  $a_i$  and  $b_i$ . It is guaranteed that there are no multiple edges.

### Output

Output one integer: the number of cactus subgraphs of Dreamoon's graph, modulo 998 244 353.

## Examples

standard input	standard output
3 3 1 2 2 3 3 1	4
5 0	0
8 9 1 5 1 8 2 4 2 8 3 4 3 6 4 7 5 7 6 8	35

## Note

Sorry, Dreamoon.

## Problem D. Determinant

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

Um\_nik has a simple connected undirected graph with the following property:

For any subset  $A$  of  $k + 1$  vertices of the graph, there exist two vertices  $a, b \in A$  and some edge  $e$ , such that all paths from  $a$  to  $b$  contain edge  $e$ .

Please help him find the determinant of the adjacency matrix of his graph modulo 998 244 353.

### Input

The first line contains three integers  $n, m, k$ : the number of vertices and edges in the graph and the given parameter ( $1 \leq n \leq 25\,000$ ,  $n - 1 \leq m \leq 500\,000$ ,  $1 \leq k \leq 25$ ).

The next  $m$  lines describe edges of the graph. Each of them contains two integers  $u$  and  $v$ : the two vertices connected by an edge ( $1 \leq u, v \leq n$ ,  $u \neq v$ ).

It is guaranteed that this graph is connected and also for any subset  $A$  of  $k + 1$  vertices of the graph, there exist two vertices  $a, b \in A$  and an edge  $e$  such that all paths from  $a$  to  $b$  contain edge  $e$ . It is guaranteed that this graph doesn't contain multiple edges.

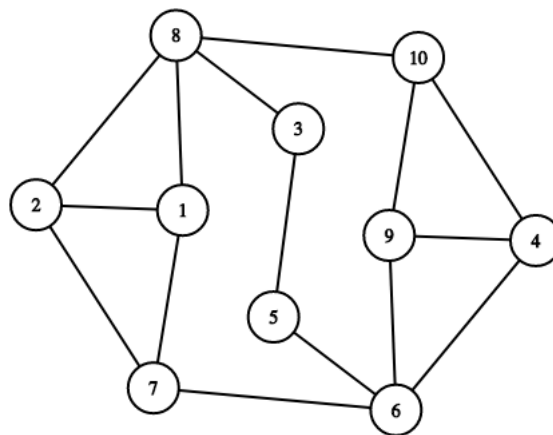
### Output

Print a single integer: the determinant of Um\_nik graph's adjacency matrix modulo 998 244 353.

## Examples

standard input	standard output
<pre>4 3 1 1 2 2 3 3 4</pre>	<pre>1</pre>
<pre>6 6 3 2 3 5 6 2 5 1 2 3 4 6 2</pre>	<pre>998244352</pre>
<pre>10 15 10 1 8 1 7 6 7 2 8 6 9 1 2 4 9 4 10 4 6 5 6 3 8 9 10 8 10 3 5 2 7</pre>	<pre>35</pre>

## Note



## Problem E. Easy Win

Input file: *standard input*  
Output file: *standard output*  
Time limit: 1.5 seconds  
Memory limit: 256 mebibytes

V-o\_o-V and LHiC are playing a game.

At first, gritukan shows them an undirected graph with  $n$  vertices where each edge has a pile of stones on it.

After that, LHiC chooses some non-empty subset of edges of this graph that forms edge-disjoint edge-simple cycles (in other words, each connected component should have an Euler circuit). If he can't (in other words, if the graph is acyclic), he loses immediately.

Otherwise, LHiC and V-o\_o-V play a game of Nim with the piles on the chosen edges. V-o\_o-V moves first. In a single move, a player may remove an arbitrary positive number of stones from a single pile. The player who cannot make a move loses.

Let's call a graph **good** if LHiC can't choose a non-empty subset of edge-disjoint cycles on which he will win Nim.

Gritukan asks  $q$  queries, can you help him?

There is a set of possible edges which can be picked by gritukan to form a **good** graph. Initially, this set is empty. In query  $i$ , first, an edge  $i$  connecting vertices  $u_i$  and  $v_i$ , with a pile of  $a_i$  stones on it and weight  $w_i$ , is added to the set of possible edges. After that, you should find the largest sum of weights of a **good** graph that gritukan can form using a subset of edges  $1, 2, \dots, i$ .

### Input

The first line contains two integers  $n$  and  $q$ : the number of vertices in the graph and the number of queries ( $2 \leq n \leq 64$ ,  $1 \leq q \leq 200\,000$ ).

Each of the next  $q$  lines contains four integers  $u_i, v_i, a_i, w_i$ , describing the edge added during  $i$ -th query ( $1 \leq u_i, v_i \leq n$ ,  $u_i \neq v_i$ ,  $0 \leq a_i < 2^{60}$ ,  $1 \leq w_i \leq 10^9$ ).

### Output

Print  $q$  lines. For the  $i$ -th query, you should print the largest sum of weights of a **good** graph that you can form using a subset of edges  $1, 2, \dots, i$ .

## Examples

standard input	standard output
3 3 1 2 0 1 2 3 0 1 3 1 0 2	1 2 3
6 6 1 2 1 1 2 3 1 2 3 4 1 3 4 1 1 4 5 6 1 2 6 5 1 1	1 3 6 9 11 11
5 5 1 2 0 1 2 3 1 1 3 4 2 3 4 5 4 9 5 1 7 29	1 2 5 14 42
5 1 3 5 35 35	35

## Problem F. Fast Spanning Tree

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

Wang Xiuhan has an initially empty undirected graph on  $n$  vertices.

Each vertex has a weight, which is a non-negative integer.

Also, he has  $m$  tuples  $(a_i, b_i, s_i)$ , where  $1 \leq a_i, b_i \leq n$ ,  $a_i \neq b_i$ , and  $s_i$  is a non-negative integer.

After that, he starts the following process:

- If there is no such  $i$  that  $a_i$  and  $b_i$  lie in different connected components of the graph and  $(\text{total weight of vertices in the component of } a_i) + (\text{total weight of vertices in the component of } b_i) \geq s_i$ , end the process.
- Otherwise, choose the smallest such  $i$ , add an edge between  $a_i$  and  $b_i$  to the graph, write this  $i$  in the notepad, and repeat the process (but now on the larger graph).

After the process was completed, a misfortune happened... Someone stole his notepad! Can you help him restore all numbers efficiently?

### Input

The first line of input contains two integers  $n$  and  $m$ : the number of vertices in Xiuhan's graph and the number of tuples he has ( $1 \leq n, m \leq 300\,000$ ).

The second line contains  $n$  space-separated integers,  $w_1, w_2, \dots, w_n$ : weights of the vertices ( $0 \leq w_i \leq 10^6$ ).

The next  $m$  lines contain a description of Xiuhan's tuples. Each of these lines contains three integers  $a_i, b_i, s_i$  ( $1 \leq a_i, b_i \leq n$ ,  $a_i \neq b_i$ ,  $0 \leq s_i \leq 10^6$ ).

### Output

On the first line, print one integer: the number of integers Xiuhan wrote in the notepad.

On the next line, you should write all these integers in the order he wrote them.

### Examples

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

## Problem G. Grammarly

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

CauchySheep has a string  $s$ .

He looked at all its different non-empty substrings and added a directed edge from  $a$  to  $b$  if  $|b| + 1 = |a|$  and  $b$  is a substring of  $a$ .

You need to calculate the number of simple paths starting from  $s$  in this graph, modulo 998 244 353.

### Input

The first line of the input contains a string  $s$  consisting of lowercase Latin letters: the string CauchySheep has ( $1 \leq |s| \leq 300\,000$ ).

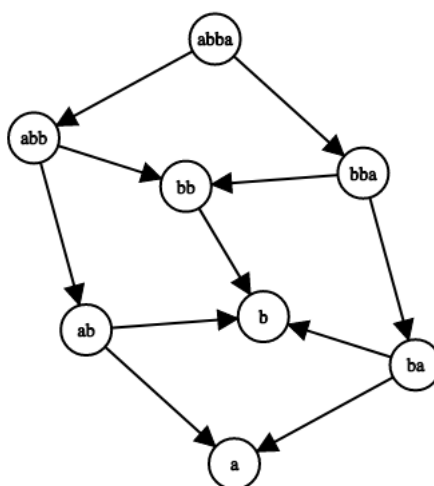
### Output

Output one integer: the number of simple paths starting from  $s$  in CauchySheep's graph, modulo 998 244 353.

### Examples

standard input	standard output
abba	13
benbeipo	255
iqiiiiiiqq	300
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa	35

### Note



## Problem H. Honorable Mention

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

Ilya Zban has an array  $a_1, a_2, \dots, a_n$ . A segment  $[l \dots r]$  of the array is the array  $a_l, a_{l+1}, \dots, a_r$ .

Ilya has  $q$  ordered triples of the form  $(l, r, k)$ , where  $1 \leq l \leq r \leq n$  and  $1 \leq k \leq r - l + 1$ . For each such triple, he asked you to answer the following query: “what is the largest sum of sums of elements of  $k$  non-empty non-intersecting subsegments of the segment  $[l \dots r]$ ?”.

### Input

The first line of input contains two integers  $n$  and  $q$ : the number of elements in the array and the number of queries ( $1 \leq n, q \leq 35\,000$ ).

The second line contains  $n$  space-separated integers  $a_1, a_2, \dots, a_n$ : the given array ( $-35\,000 \leq a_i \leq 35\,000$ ).

The next  $q$  lines contain queries. Each of them contains three integers  $l, r, k$ : the given segment and the number of non-intersecting subsegments on it that you should find ( $1 \leq l \leq r \leq n, 1 \leq k \leq r - l + 1$ ).

### Output

Output  $q$  integers on separate lines: the answers to the queries.

### Examples

standard input	standard output
5 5 -1 2 -3 4 -5 1 5 1 1 5 2 1 5 3 1 5 4 1 5 5	4 6 5 2 -3
5 1 7 7 7 7 7 1 5 1	35

## Problem I. Interactive Vertex

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

*This is an interactive problem.*

Endagorion has a tree on  $n$  vertices, and he also showed it to you. He chooses one vertex  $u$  as a special vertex, but now he won't tell you anything about it!

Instead, you can ask him questions. For each question, you should choose a vertex  $x$ , an integer  $k$ , and  $k$  vertices  $v_1, v_2, \dots, v_k$ , and he will tell you whether it is true that  $\min(\text{dist}(u, v_i)) \geq \text{dist}(u, x)$ . Here,  $\text{dist}(p, q)$  is the number of edges in the simple path between vertices  $p$  and  $q$  in the tree.

You should guess the special vertex using at most  $4\lceil \log_2 n \rceil$  queries.

Endagorion is very honest, so he wouldn't change the vertex between your queries (in other words, the interactor is not adaptive).

**As the constraints are large, and flush is an expensive operation, make sure that you are not flushing too often. You may do it only once after each query.**

### Interaction Protocol

The interaction starts with a line containing one integer  $n$ : the number of vertices in Endagorion's tree ( $2 \leq n \leq 200\,000$ ).

Each of the next  $n - 1$  lines contains two integers  $u$  and  $v$  ( $1 \leq u, v \leq n$ ), denoting an edge between  $u$  and  $v$ . It is guaranteed that the given edges form a tree.

After that, you can make queries.

To make a query, print one line containing “?  $k$ ” ( $1 \leq k \leq n$ ), an integer  $x$  ( $1 \leq x \leq n$ ) and then  $k$  distinct integers  $v_1, v_2, \dots, v_k$  ( $1 \leq v_i \leq n$ ). Separate consecutive integers by single spaces. Then **flush** the output.

After each query, read one line with a single integer  $ans \in \{0, 1\}$ . If  $\min(\text{dist}(u, v_i)) \geq \text{dist}(u, x)$ , then  $ans$  will be equal to 1. Otherwise,  $ans$  will be equal to 0.

When you find the special vertex  $u$  ( $1 \leq u \leq n$ ), print one line containing “!  $u$ ”, and then flush the output and terminate.

Your solution will get **Wrong Answer** or **Time Limit Exceeded** if you make more than  $4\lceil \log_2 n \rceil$  queries.

Your solution will get **Idleness Limit Exceeded** if you don't print anything or forget to flush the output.

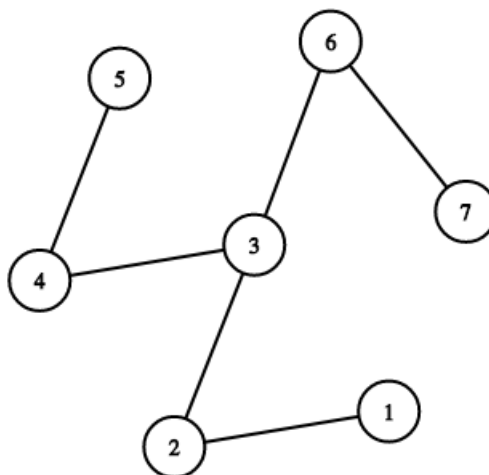
To **flush**, you need to do the following right after printing a query and a line break:

- `fflush(stdout)` or `cout.flush()` in C++;
- `System.out.flush()` in Java;
- `flush(output)` in Pascal;
- `stdout.flush()` in Python;
- see documentation for other languages.

## Examples

standard input	standard output
5 1 2 1 3 1 4 1 5 1	? 4 1 2 3 4 5 ! 1
5 1 2 1 3 1 4 1 5 0 0 0 0	? 4 1 2 3 4 5 ? 3 1 2 3 4 ? 2 1 2 3 ? 1 1 2 ! 2
7 1 2 2 3 3 4 4 5 3 6 6 7 1	? 3 3 5 7 1 ! 3

## Note



## Problem J. Jiry Matchings

Input file: *standard input*  
Output file: *standard output*  
Time limit: 6 seconds  
Memory limit: 512 mebibytes

Ruyi Ji has a tree where the vertices are numbered by integers from 1 to  $n$  and each edge has a weight. For each  $k \leq (n - 1)$ , he asked you to find the largest total weight of a matching with  $k$  edges if it exists.

### Input

The first line of input contains one integer  $n$ : the number of vertices in the tree ( $2 \leq n \leq 200\,000$ ).

Each of the next  $n - 1$  lines contains three integers  $u_i, v_i, w_i$ , describing an edge from  $u_i$  to  $v_i$  with weight  $w_i$  in the tree ( $1 \leq u_i, v_i \leq n, u_i \neq v_i, -10^9 \leq w_i \leq 10^9$ ).

It is guaranteed that the given graph is a tree.

### Output

Output  $n - 1$  integers: the largest weights of the matchings with  $1, 2, \dots, n - 1$  edges. If there is no such matching for the current  $k$ , print “?” instead.

### Examples

standard input	standard output
5 1 2 3 2 3 5 2 4 4 3 5 2	5 6 ? ?
10 2 8 -5 5 10 5 3 4 -5 1 6 5 3 9 5 1 7 -3 4 8 -5 10 8 -5 1 8 -3	5 10 15 10 ? ? ? ? ?
2 1 2 35	35

### Note

In the first sample, with  $k = 1$  you should take edge  $(2, 3)$  with weight 5. And with  $k = 2$  you should take two edges,  $(2, 4)$  and  $(3, 5)$ , with total weight 6. There are no matchings with a greater number of edges.

## Problem K. K-pop Strings

Input file: *standard input*  
Output file: *standard output*  
Time limit: 7 seconds  
Memory limit: 512 mebibytes

A substring  $s[l..r]$  is a tandem repeat if  $r - l + 1$  is even and  $s[l.. \frac{l+r-1}{2}] = s[\frac{l+r+1}{2}..r]$ .

Recently Gennady came up with a method to calculate the number of tandem repeats in a string using suffix structures, and now he came up with a new type of strings based on tandem repeats. Gennady thinks that string  $s$  of length  $n$  is a K-pop string if there are no tandem repeats of length  $\geq n - k$ .

Help him find the number of K-pop strings consisting only of the characters '1', '2', ..., '9', 'a', 'b', ..., 'z', modulo 998 244 353.

### Input

The first line of input contains two integers  $n$  and  $k$ : the required length of string and the parameter ( $1 \leq n \leq 100$ ,  $0 \leq k \leq 16$ ).

### Output

Output one integer: the number of K-pop strings of length  $n$  for the given  $k$ , consisting only of nonzero digits and lowercase alphabetic characters, modulo 998 244 353.

### Examples

standard input	standard output
1 16	35
4 0	1499400
15 5	911125634

### Note

The answer for the first example is 35 because all strings of length 1 are possible: "1", "2", ..., "9", "a", "b", ..., "z".

The answer for the second example is  $35^4 - 35^2$ .