第二届 YUC 柚子杯半决赛：人类智慧大赛天堂站

Association for the Worship of rubisama, Apr 29th 2024

【比赛简介】

Ciallo，第二届柚子杯（Yuzusoft Cup）第七站将在 2024 年 4 月 29 日，位于天堂举行。比赛出题人是るび様（rubi sama），最新的作品《死に逝く騎士、異世界に響く断末魔》在 4 月 26 日已经发售，欢迎大家游玩！

比赛采用 ICPC 赛制，时长 360 小时（15 天）。

比赛采用编译选项如下：

对 C++语言	`-std=c++14 -O2 -static`

比赛试题一共有 12 题，共 23 页。请检查印刷是否完整无误。

出题组织：るび様を崇める会。

比赛网址：yuzu-soft.com/yuzucup、https://rubisama.com/（分流）。

比赛在线测评：hfoj.net/contest。

Hosted By

Prepared By

Problem A. Clacking Balls

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

There are m baskets placed along a circle, numbered from 1 to m in clockwise order (basket m is next to basket 1). Furthermore, there are n balls, where ball i is initially placed in basket a_i , and no basket contains more than one ball.

Alice is allowed to perform the following operation, which always takes exactly one second whether you move/throw a ball or not:

- Alice chooses an integer i between 1 and n uniformly at random.
- If ball i was thrown away before, do nothing.
- Otherwise, ball i is moved from the basket currently containing it to the next basket (in clockwise order). If the target basket currently contains another ball j , throw ball j away.

She repeats this operation until there is exactly one ball left. Calculate the expected time needed (in seconds) for Alice to end the process.

Input

Each test contains multiple test cases. The first line contains an integer t $(1 \le t \le 10^4)$ — the number of test cases. The description of the test cases follows.

The first line of each test case contains two integers n and m $(1 \le n \le 3 \cdot 10^5, n \le m \le 10^9)$ — the number of balls and the number of baskets.

The second line of each test case contains n integers $a_1, a_2, ..., a_n$ $(1 \le a_i \le m$, a_i 's are pairwise distinct) — the initial position of each ball.

It is guaranteed that the sum of n over all test cases does not exceed $3 \cdot 10^5$.

Output

For each test case, print one integer: the expected amount of time (in seconds) Alice needs to end the process, modulo $10^9 + 7$.

Sample

standard input	standard output
5 3 10 5 1 4 2 15 15 1 6 6 1 2 3 4 5 6 6 9 6 5 4 3 2 1 1 100 69	600000042 14 35 333333409 0

Note

In the first test case, Alice could have proceeded as follows (we define $a_i = -1$ if ball i has been thrown out):

- Initially, $a = [5,1,4]$.
- Alice chooses $i = 2$ with probability $\frac{1}{3}$, and ball 2 is moved to basket 2 . After this, $a = [5,2,4]$.
- Alice chooses $i = 2$ with probability $\frac{1}{3}$, and ball 2 is moved to basket 3 . After this, $a = [5,3,4]$.
- Alice chooses $i = 2$ with probability $\frac{1}{3}$, and ball 2 is moved to basket 4 . As basket 4 previously contains ball 3 , this ball is thrown out. After this, $a = [5,4,-1]$.
- Alice chooses $i = 3$ with probability $\frac{1}{3}$. Ball 3 has already been thrown out, so nothing happens. After this, $a = [5,4,-1]$.
- Alice chooses $i = 2$ with probability $\frac{1}{3}$, and ball 2 is moved to basket 5 , which throws out ball 1 . After this, $a = [-1,5,-1]$, and the process ends.

The answer for this test case is $\frac{189}{5}$ .The answer for the second test case is 14 (note that these two balls are next to each other).

The answer for the third test case is 35 .

The answer for the fourth test case is $\frac{220}{3}$.

In the fifth test case, as there is only one ball initially, the answer is 0 .

Problem B. The Harmonization of XOR

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

You are given an array of exactly n numbers $[1,2,3,...,n]$ along with integers k and x .

Partition the array in exactly k non-empty disjoint subsequences such that the bitwise XOR of all numbers in each subsequence is x , and each number is in exactly one subsequence. Notice that there are no constraints on the length of each subsequence.

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

For example, for $n = 15$, $k = 6$, $x = 7$, the following scheme is valid:

- $[6,10,11]$, $6 \oplus 10 \oplus 11 = 7$,
- $[5,12,14]$, $5 \oplus 12 \oplus 14 = 7$,
- $[3,9,13]$, $3 \oplus 9 \oplus 13 = 7$,
- $[1,2,4]$, $1 \oplus 2 \oplus 4 = 7$,
- $[8,15]$, $8 \oplus 15 = 7$,
- $[7]$, $7 = 7$,

where $\oplus$ represents the bitwise XOR operation.The following scheme is invalid, since 8 , 15 do not appear:

- $[6,10,11]$, $6 \oplus 10 \oplus 11 = 7$,
- $[5,12,14]$, $5 \oplus 12 \oplus 14 = 7$,
- $[3,9,13]$, $3 \oplus 9 \oplus 13 = 7$,
- $[1,2,4]$, $1 \oplus 2 \oplus 4 = 7$,
- $[7]$, $7 = 7$.

The following scheme is invalid, since 3 appears twice, and 1 , 2 do not appear:

- $[6,10,11]$, $6 \oplus 10 \oplus 11 = 7$,
- $[5,12,14]$, $5 \oplus 12 \oplus 14 = 7$,
- $[3,9,13]$, $3 \oplus 9 \oplus 13 = 7$,
- $[3,4]$, $3 \oplus 4 = 7$,
- $[8,15]$, $8 \oplus 15 = 7$,
- $[7]$, $7 = 7$.

Input

Each test contains multiple test cases. The first line contains an integer t ($1 \le t \le 10^4$) — the number of test cases.

The first and the only line of each test case contains three integers n , k , x ($1 \le k \le n \le 2 \cdot 10^5$; $1 \le x \le 10^9$) — the length of the array, the number of subsequences and the required XOR.

It's guaranteed that the sum of n does not exceed $2 \cdot 10^5$.

Output

For each test case, if it is possible to partition the sequence, print 'YES' in the first line. In the i -th of the following k lines first print the length s_i of the i -th subsequence, then print s_i integers, representing the elements in the i -th subsequence. If there are multiple answers, print any. Note that you can print a subsequence in any order.

If it is not possible to partition the sequence, print ‘NO’.

Sample

standard input	*standard output*
7 15 6 7 11 4 5 5 3 2 4 1 4 6 1 7 11 5 5 11 6 5	YES 3 6 10 11 3 5 12 14 3 3 9 13 3 1 2 4 2 8 15 1 7 YES 2 1 4 2 2 7 2 3 6 5 5 8 9 10 11 NO YES 4 1 2 3 4 YES 6 1 2 3 4 5 6 NO NO

Note

In the first test case, we construct the following 6 subsequences:

- $[6,10,11]$, $6 \oplus 10 \oplus 11 = 7$,
- $[5,12,14]$, $5 \oplus 12 \oplus 14 = 7$,
- $[3,9,13]$, $3 \oplus 9 \oplus 13 = 7$,
- $[1,2,4]$, $1 \oplus 2 \oplus 4 = 7$,
- $[8,15]$, $8 \oplus 15 = 7$,
- $[7]$, $7 = 7$.

In the second test case, we construct the following 4 subsequences:

- $[1,4]$, $1 \oplus 4 = 5$,
- $[2,7]$, $2 \oplus 7 = 5$,
- $[3,6]$, $3 \oplus 6 = 5$,
- $[5,8,9,10,11]$, $5 \oplus 8 \oplus 9 \oplus 10 \oplus 11 = 5$.

The following solution is considered correct in this test case as well:

- $[1,4]$, $1 \oplus 4 = 5$,
- $[2,7]$, $2 \oplus 7 = 5$,
- $[5]$, $5 = 5$,
- $[3,6,8,9,10,11]$, $3 \oplus 6 \oplus 8 \oplus 9 \oplus 10 \oplus 11 = 5$.

Problem C. Computer Game

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

Ivan plays some computer game. There are n quests in the game. Each quest can be upgraded once, this increases the reward for its completion. Each quest has 3 parameters a_i , b_i , p_i : reward for completing quest before upgrade, reward for completing quest after upgrade ($a_i < b_i$) and probability of successful completing the quest.

Each second Ivan can try to complete one quest and he will succeed with probability p_i . In case of success Ivan will get the reward and opportunity to upgrade any one quest (not necessary the one he just completed). In case of failure he gets nothing. Quests do not vanish after completing.

Ivan has t seconds. He wants to maximize expected value of his total gain after t seconds. Help him to calculate this value.

Input

First line contains 2 integers n ($1 \le n \le 10^5$) and t ($1 \le t \le 10^{10}$) — number of quests and total time.

Following n lines contain description of quests. Each description is 3 numbers a_i b_i p_i ($1 \le a_i < b_i \le 10^8$, $0 < p_i < 1$) — reward for completing quest before upgrade, reward for completing quest after upgrade and probability of successful completing of quest. a_i and b_i are integers. All probabilities are given with at most 9 decimal places.

Output

Print the expected value.

Your answer will be accepted if absolute or relative error does not exceed 10^{-6} . Formally, let your answer be a , and the jury's answer be b . Your answer is considered correct if $\frac{|a-b|}{\max(b,\ 1)} \le 10^{-6}$.

Sample

standard input	*standard output*
3 2 3 1000 0.5 1 2 0.48 3 20 0.3	252.2500000000000
2 2 1 1000 0.1 2 3 0.2	20.7200000000000

Problem D. Uniformly Branched Trees

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

A tree is a connected graph without cycles.

Two trees, consisting of n vertices each, are called isomorphic if there exists a permutation $p\colon 1, \ldots, n \to 1, \ldots, n$ such that the edge (u, v) is present in the first tree if and only if the edge (p_u, p_v) is present in the second tree.

Vertex of the tree is called internal if its degree is greater than or equal to two.

Count the number of different non-isomorphic trees, consisting of n vertices, such that the degree of each internal vertex is exactly d . Print the answer over the given prime modulo mod .

Input

The single line of the input contains three integers n , d and mod ($1 \le n \le 1000$, $2 \le d \le 10$, $10^8 \le mod \le 10^9$) — the number of vertices in the tree, the degree of internal vertices and the prime modulo.

Output

Print the number of trees over the modulo mod .

Sample

standard input	*standard output*
`5 2 433416647`	`1`
`10 3 409693891`	`2`
`65 4 177545087`	`910726`

Problem D. Centroids

Input file: *standard input*
Output file: *standard output*
Time limit: 4 second
Memory limit: 512 megabytes

Tree is a connected acyclic graph. Suppose you are given a tree consisting of n vertices. The vertex of this tree is called centroid if the size of each connected component that appears if this vertex is removed from the tree doesn't exceed $\frac{n}{2}$.

You are given a tree of size n and can perform no more than one edge replacement. Edge replacement is the operation of removing one edge from the tree (without deleting incident vertices) and inserting one new edge (without adding new vertices) in such a way that the graph remains a tree. For each vertex you have to determine if it's possible to make it centroid by performing no more than one edge replacement.

Input

The first line of the input contains an integer n ($2 \le n \le 400000$) — the number of vertices in the tree. Each of the next $n-1$ lines contains a pair of vertex indices u_i and v_i ($1 \le u_i, v_i \le n$) — endpoints of the corresponding edge.

Output

Print n integers. The i -th of them should be equal to 1 if the i -th vertex can be made centroid by replacing no more than one edge, and should be equal to 0 otherwise.

Sample

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

Note

In the first sample each vertex can be made a centroid. For example, in order to turn vertex 1 to centroid one have to replace the edge (2,3) with the edge (1,3) .

Problem E. Parallel Swaps Sort

Input file: *standard input*
Output file: *standard output*
Time limit: 7 second
Memory limit: 1024 megabytes

You are given a permutation $p_1, p_2, ..., p_n$ of $[1,2,...,n]$. You can perform the following operation some (possibly 0) times:

- choose a subarray $[l, r]$ of even length;
- swap a_l , a_{l+1} ;
- swap a_{l+2} , a_{l+3} (if $l + 3 \le r$);
- ...
- swap a_{r-1} , a_r .

Sort the permutation in at most 10^6 operations. You do not need to minimize the number of operations.

Input

The first line contains a single integer n ($2 \le n \le 3 \cdot 10^5$) — the length of the permutation.

The second line contains n integers $p_1, p_2, ..., p_n$ ($1 \le p_i \le n$, the p_i are distinct) — the permutation before performing the operations.

Output

Output your operations in the following format.

The first line should contain an integer k ($0 \le k \le 10^6$) — the number of operations.

The next k lines represent the k operations in order. Each of these k lines should contain two integers l and r ($1 \le l < r \le n$, $r - l + 1$ must be even) — the corresponding operation consists in choosing the subarray $[l, r]$ and swapping its elements according to the problem statement.

After all the operations, $a_i = i$ must be true for each i ($1 \le i \le n$).

Sample

standard input	*standard output*
5 2 5 4 1 3	5 1 4 1 2 2 5 1 4 4 5
9 1 2 3 4 5 6 7 8 9	0
10 6 4 2 3 8 10 9 1 5 7	15 1 8 6 9 1 8 3 10 1 10 1 10 1 6 6 9 6 9

standard input	*standard output*
	(continue) 2 7 9 10 5 10 1 6 2 9 1 10

Note

In the first test:

- At the beginning, $p = [2,5,4,1,3]$.
- In the first operation, you can choose $[l, r] = [1,4]$. Then, (a_1, a_2) are swapped and (a_3, a_4) are swapped. The new permutation is $p = [5,2,1,4,3]$.
- In the second operation, you can choose $[l, r] = [1,2]$. Then, (a_1, a_2) are swapped. The new permutation is $p = [2,5,1,4,3]$.
- In the third operation, you can choose $[l, r] = [2,5]$. Then, (a_2, a_3) are swapped and (a_4, a_5) are swapped. The new permutation is $p = [2,1,5,3,4]$.
- In the fourth operation, you can choose $[l, r] = [1,4]$. Then, (a_1, a_2) are swapped and (a_3, a_4) are swapped. The new permutation is $p = [1,2,3,5,4]$.
- In the fifth operation, you can choose $[l, r] = [4,5]$. Then, (a_4, a_5) are swapped. The new permutation is $p = [1,2,3,4,5]$, which is sorted.

In the second test, the permutation is already sorted, so you do not need to perform any operation.

Problem F. Cube Snake

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

You've got an $n \times n \times n$ cube, split into unit cubes. Your task is to number all unit cubes in this cube with positive integers from 1 to n^3 so that:

- each number was used as a cube's number exactly once;
- for each $1 \leq i < n^3$, unit cubes with numbers i and $i+1$ were neighbouring (that is, shared a side);
- for each $1 \leq i < n$ there were at least two different subcubes with sizes $i \times i \times i$, made from unit cubes, which are numbered with consecutive numbers. That is, there are such two numbers x and y , that the unit cubes of the first subcube are numbered by numbers x , $x+1$, $\ldots$, $x+i^3-1$, and the unit cubes of the second subcube are numbered by numbers y , $y+1$, $\ldots$, $y+i^3-1$.

Find and print the required numeration of unit cubes of the cube.

Input

The first line contains a single integer n $(1 \leq n \leq 50)$ — the size of the cube, whose unit cubes need to be numbered.

Output

Print all layers of the cube as n $n \times n$ matrices. Separate them with new lines. Print the layers in the order in which they follow in the cube. See the Sample for clarifications.

It is guaranteed that there always is a solution that meets the conditions given in the problem statement.

Sample

standard input	*standard output*
3	1 4 17 2 3 18 27 26 19 8 5 16 7 6 15 24 25 20 9 12 13 10 11 14 23 22 21

Note

In the sample the cubes with sizes $2 \times 2 \times 2$ are numbered with integers $1, \ldots, 8$ and $5, \ldots, 12$.

Problem G. Linova and Kingdom

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

Writing light novels is the most important thing in Linova's life. Last night, Linova dreamed about a fantastic kingdom. She began to write a light novel for the kingdom as soon as she woke up, and of course, she is the queen of it.

There are n cities and $n-1$ two-way roads connecting pairs of cities in the kingdom. From any city, you can reach any other city by walking through some roads. The cities are numbered from 1 to n , and the city 1 is the capital of the kingdom. So, the kingdom has a tree structure.

As the queen, Linova plans to choose exactly k cities developing industry, while the other cities will develop tourism. The capital also can be either industrial or tourism city.

A meeting is held in the capital once a year. To attend the meeting, each industry city sends an envoy. All envoys will follow the shortest path from the departure city to the capital (which is unique).

Traveling in tourism cities is pleasant. For each envoy, his happiness is equal to the number of tourism cities on his path.

In order to be a queen loved by people, Linova wants to choose k cities which can maximize the sum of happinesses of all envoys. Can you calculate the maximum sum for her?

Input

The first line contains two integers n and k ($2 \le n \le 2 \cdot 10^5$, $1 \le k < n$) — the number of cities and industry cities respectively.

Each of the next $n-1$ lines contains two integers u and v ($1 \le u, v \le n$), denoting there is a road connecting city u and city v .

It is guaranteed that from any city, you can reach any other city by the roads.

Output

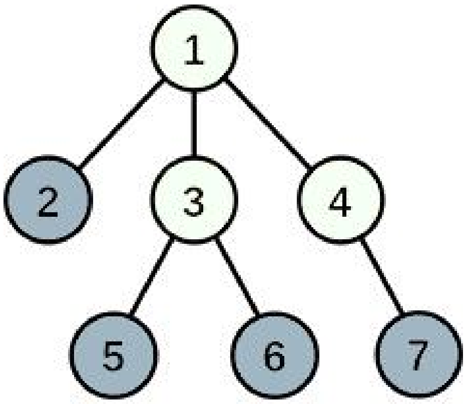
Print the only line containing a single integer — the maximum possible sum of happinesses of all envoys.

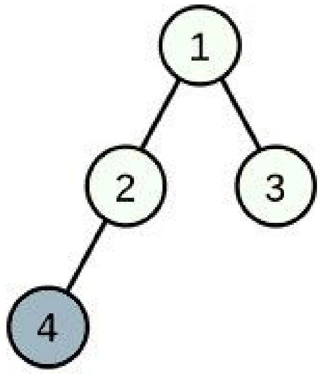
Sample

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

standard input	*standard output*
(continue) 8 5 7 5 1 7 6 1 3 7 8 3 2 1 4 5	9

Note





In the first example, Linova can choose cities 2 , 5 , 6 , 7 to develop industry, then the happiness of the envoy from city 2 is 1 , the happiness of envoys from cities 5 , 6 , 7 is 2 . The sum of happinesses is 7 , and it can be proved to be the maximum one.

In the second example, choosing cities 3 , 4 developing industry can reach a sum of 3 , but remember that Linova plans to choose exactly k cities developing industry, then the maximum sum is 2 .

Problem H. Jellyfish and Inscryption

Input file:	*standard input*
Output file:	*standard output*
Time limit:	2 second
Memory limit:	1024 megabytes

Jellyfish loves playing a game called 'Inscryption' which is played on a directed acyclic graph with n vertices and m edges. All edges $a \to b$ satisfy $a < b$.

You need to move from vertex 1 to vertex n along the directed edges, and then fight with the final boss.

You will collect cards and props in the process.

Each card has two attributes: HP and damage. If a card's HP is a and its damage is b , then the power of the card is $a \times b$.

Each prop has only one attribute: power.

In addition to vertex 1 and vertex n , there are some vertices that trigger special events. The special events are:

1. You will get a card with a HP, and b damage.
2. If you have at least one card, choose one of your cards and increase its HP by x .
3. If you have at least one card, choose one of your cards and increase its damage by y .
4. You will get a prop with w power.

When you get to vertex n , you can choose at most one of your cards and multiply its damage by 10^9 .

The final boss is very strong, so you want to maximize the sum of the power of all your cards and props. Find the maximum possible sum of power of all your cards and props if you play the game optimally.

Input

The first line contains two integers n and m $(2 \le n \le 200$, $n-1 \le m \le \min\left(\frac{n(n-1)}{2}, 2000\right))$ — the number of the vertices and the number of the edges.

In the following n lines, the i -th $(1 \le i \le n)$ line describes the special event that will trigger on vertex i :

- 0 — no special events.
- 1 a b $(1 \le a, b \le 200)$ — you will get a card with a HP, and b damage.
- 2 x $(1 \le x \le 200)$ — if you have at least one card, choose one of your cards and increase its HP by x .
- 3 y $(1 \le y \le 200)$ — if you have at least one card, choose one of your cards and increase its damage by y .
- 4 w $(1 \le w \le 10^6)$ — you will get a prop with w power.

In the following m lines, each line contains two integers u and v $(1 \le u < v \le n)$ representing a directed edge from vertex u to vertex v .

It is guaranteed that every edge (u, v) appears at most once.

It is guaranteed that there are no special events on vertex 1 and vertex n .

It is guaranteed that for all i , there exists a path from vertex 1 to vertex n that passes through vertex i .

Output

Output a single integer — the maximum sum of the power of all your cards and props.

Sample

<table>
<tr><th>standard input</th><th>standard output</th></tr>
<tr><td><pre>6 8
0
1 2 10
1 6 6
2 8
3 10
0
1 2
1 3
2 4
2 5
3 4
3 5
4 6
5 6</pre></td><td><pre>100000000000</pre></td></tr>
<tr><td><pre>4 3
0
4 1000000
4 1000000
0
1 2
2 3
3 4</pre></td><td><pre>2000000</pre></td></tr>
<tr><td colspan="2">For sample 3~6, you can see them in the website.</td></tr>
</table>

Note

In the first example, we will play the game in the following order:

- move from vertex 1 to vertex 2 , get a card with 2 HP, and 10 damage.
- move from vertex 2 to vertex 4 , choose the card we get from vertex 2 and increase its HP by 8 .
- move from vertex 4 to vertex 6 , choose the card we get from vertex 2 and multiply its damage by 10^9 .

In the end, we will have a card with $(2+8)=10$ HP and $10\times 10^9=10^{10}$ damage, It's power is $10\times 10^{10}=10^{11}$. Because we only have the card we get from vertex 2 , so the sum of power of all your cards and props is 10^{11} .

Problem I. Quadratic Set

Input file:	*standard input*
Output file:	*standard output*
Time limit:	4 second
Memory limit:	256 megabytes

Let's call a set of positive integers $a_1, a_2, ..., a_k$ quadratic if the product of the factorials of its elements is a square of an integer, i. e. $\prod_{i=1}^{k} a_i! = m^2$, for some integer m .

You are given a positive integer n .

Your task is to find a quadratic subset of a set $1,2,...,n$ of maximum size. If there are multiple answers, print any of them.

Input

A single line contains a single integer n ($1 \le n \le 10^6$).

Output

In the first line, print a single integer — the size of the maximum subset. In the second line, print the subset itself in an arbitrary order.

Sample

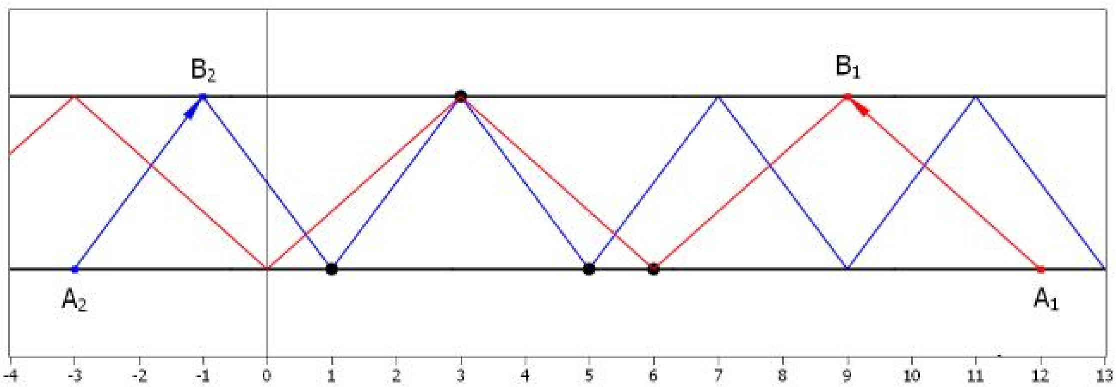
standard input	*standard output*
1	1 1
4	3 1 3 4
7	4 1 4 5 6
9	7 1 2 4 5 6 7 9

Problem J. Ray in the tube

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

You are given a tube which is reflective inside represented as two non-coinciding, but parallel to Ox lines. Each line has some special integer points — positions of sensors on sides of the tube.

You are going to emit a laser ray in the tube. To do so, you have to choose two integer points A and B on the first and the second line respectively (coordinates can be negative): the point A is responsible for the position of the laser, and the point B — for the direction of the laser ray. The laser ray is a ray starting at A and directed at B which will reflect from the sides of the tube (it doesn't matter if there are any sensors at a reflection point or not). A sensor will only register the ray if the ray hits exactly at the position of the sensor.



Examples of laser rays. Note that image contains two examples. The 3 sensors (denoted by black bold points on the tube sides) will register the blue ray but only 2 will register the red.Calculate the maximum number of sensors which can register your ray if you choose points A and B on the first and the second lines respectively.

Input

The first line contains two integers n and y_1 ($1 \leq n \leq 10^5$, $0 \leq y_1 \leq 10^9$) — number of sensors on the first line and its y coordinate.

The second line contains n integers $a_1, a_2, ..., a_n$ ($0 \leq a_i \leq 10^9$) — x coordinates of the sensors on the first line in the ascending order.

The third line contains two integers m and y_2 ($1 \leq m \leq 10^5$, $y_1 < y_2 \leq 10^9$) — number of sensors on the second line and its y coordinate.

The fourth line contains m integers $b_1, b_2, ..., b_m$ ($0 \leq b_i \leq 10^9$) — x coordinates of the sensors on the second line in the ascending order.

Output

Print the only integer — the maximum number of sensors which can register the ray.

Sample

standard input	standard output
3 1 1 5 6 1 3 3	3

Note

One of the solutions illustrated on the image by pair A_2 and B_2 .

Problem K. Mahou Shuojo Site

Input file: *standard input*
Output file: *standard output*
Time limit: 4 second
Memory limit: 64 megabytes

I think about dying every day.

Why don't you just die?

A magical girl website? What's this?

“It's unfortunate. It's unfortunate.”

“To you who are like this...”

“the power of magic.”

Amaya Sayu.

It's a pain in the ass to introduce myself.

You collapsed after using your wand too much, and have been unconscious ever since.

That's why I healed you.

On the future August 11th at 19:23 hours

On that day, tempest will strike-

Tempest will bring an end to this decaying world and open the door to a new world But in the process, most humans are expected to die

Tempest's arrival is inevitable, but there's only one way to gain salvation--

That is to fill the King's belly

The King feeds on the negative energy of humans, and by offering this energy, he can avoid the deaths caused by Tempest.

By using the wand given by the magical girl's website, you can store negative energy in the wand.

Only by using your wand as much as possible, and offering more negative energy than other magical girls, can you become a Tempest-exempted person.

If you want to survive, then give your wand filled with negative energy to us on the day of the Tempest!

Finally, I wish you all a happy ending.

Remove all the negativity from human beings.

Tempest will sweep the world.

Problem Statement

Tempest has three integer arrays a, b, c of length n. We have $1 \le a_i, b_i, c_i \le n$ for all i.

In order to accelerate his potato farming, he organizes his farm in a manner based on these three arrays. He is now going to complete m operations to count how many potatoes he can get. Each operation will be in one of the two following forms:

1. Tempest reorganizes his farm and makes the k-th element of the array a equal to x. In other words, perform the assignment $a_k := x$.

2. Given a positive integer r, Tempest receives a potato for each triplet (i, j, k), such that $1 \le i < j < k \le r$, and $b_{a_i} = a_j = c_{a_k}$. Count the number of potatoes that he receives; that is, the number of such triplets.

Help Tempest complete these operations.

Input

The first line contains two integers n , m $(1 \le n \le 2 \cdot 10^5$, $1 \le m \le 5 \cdot 10^4)$.

The second line contains n integers $a_1, a_2, ..., a_n$ $(1 \le a_i \le n)$.

The third line contains n integers $b_1, b_2, ..., b_n$ $(1 \le b_i \le n)$.

The fourth line contains n integers $c_1, c_2, ..., c_n$ $(1 \le c_i \le n)$.

The next m lines describe operations, the i -th line describes the i -th operation in one of these two formats:

- $1\ k\ x$ $(1 \le k, x \le n)$, representing an operation of the first type.
- $2\ r$ $(1 \le r \le n)$, representing an operation of the second type.

It is guaranteed that there is at least one operation of the second type.

Output

For each operation of the second type print the answer.

Sample

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

Note

For the first operation, the triplets are:

- $i = 1$, $j = 2$, $k = 3$
- $i = 2$, $j = 3$, $k = 4$
- $i = 3$, $j = 4$, $k = 5$

There is no satisfying triplet for the third operation.

For the fourth operation, the triplets are:

- $i = 2$, $j = 4$, $k = 5$
- $i = 3$, $j = 4$, $k = 5$