Problem A. Angle Beats

Given n points $P_1, P_2, \cdots, P_n$ on 2D plane and q queries. In i-th query, a point A_i is given, and you should determine the number of tuples (u, v) that $1 \le u < v \le n$ and A_i, P_u, P_v form a non-degenerate right-angled triangle.

Input

The first line contains two positive integers n, q $(2 \le n \le 2\,000, 1 \le q \le 2\,000)$, denoting the number of given points and the number of queries.

Next n lines each contains two integers x_i, y_i $(|x_i|, |y_i| \le 10^9)$, denoting a given point P_i.

Next q lines each contains two integers x_i, y_i $(|x_i|, |y_i| \le 10^9)$, denoting a query point A_i.

It is guaranteed that the input $n + q$ points are all pairwise distinct.

Output

Output q lines each contains a non-negative integer, denoting the answer to corresponding query.

Example

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

Explanation

For query $(0, 0)$, the 4 right-angled triangles are

- $\{(0,0),(0,1),(1,0)\}$
- $\{(0,0),(0,1),(-1,0)\}$
- $\{(0,0),(0,-1),(1,0)\}$
- $\{(0,0),(0,-1),(-1,0)\}$

For query $(1, 1)$, the 3 right-angled triangles are

- $\{(1,1),(0,1),(1,0)\}$
- $\{(1,1),(0,1),(0,-1)\}$
- $\{(1,1),(1,0),(-1,0)\}$

Problem B. The Tree of Haruhi Suzumiya

This problem is written to commemorate the victims in Kyoto Animation arson attack on July 18, 2019.

The members in *SOS Dan (**S**ekai o **O**oini Moriageru Tame no **S**uzumiya Haruhi no Dan)* want to assemble a Christmas tree. You know, most Christmas trees are decorations instead of real trees.

The tree contains n vertices which are numbered from 1 to n, where vertex i is of weight w_i. The number of edges on the simple path from vertex i to 1 is denoted as d_i (vertex 1 is important as it is the top vertex). However, the members have various dislikes, so they start discussing:

- Haruhi says, "I dislike the vertex pairs (i, j) that i is an ancestor of j and $w_i > w_j$".
- Kyon says, "I dislike the vertex pairs (i, j) that i is an ancestor of j and $w_i < w_j$".
- Itsuki says, "I dislike the vertex pairs (i, j) that $i < j$ and neither i nor j is an ancestor of the other vertex".
- Mikuru says, "I dislike the vertices that are far away from vertex 1".
- Yuki says nothing.

Now the members are divided into two groups to assemble the tree. Haruhi, Itsuki and Mikuru are in group A while Kyon and Yuki are in group B. Both groups are to choose some vertices to assemble. Finally, each vertex should be chosen by exactly one of the two groups. Let's denote the vertex set chosen by group A as $V(A)$, the vertex set chosen by group B as $V(B)$. So $V(A) \bigcup V(B) = \{1, 2, \cdots, n\}$ and $V(A) \bigcap V(B) = \emptyset$ always hold.

The dislike level of group A (denoted by $D(A)$) is the number of vertex pairs (i, j) that are disliked by at least one of the two members (Haruhi, Itsuku) where $i, j \in V(A)$ add the sum of d_u (the dislike level of Mikuru) where $u \in V(A)$, while the dislike level of group B (denoted by $D(B)$) is the number of vertex pairs (i, j) that are disliked by Kyon where $i, j \in V(B)$.

Formally, $D(A)$ can be computed by definition with following formula:

$$\sum_{i \in V(A)} \sum_{j \in V(A)} [\ (i, j) \text{ is disliked by Haruhi or Itsuki }\ ] + \sum_{u \in V(A)} d_u$$

$D(B)$ can be computed by definition with following formula:

$$\sum_{i \in V(B)} \sum_{j \in V(B)} [\ (i, j) \text{ is disliked by Kyon }\ ]$$

Where $[X]$ equals 1 if the statement X is true, while $[X]$ equals 0 if false.

Yuki wants to know the minimum value of $D(A) + D(B)$ when $|V(B)| = 0, 1, 2, \cdots, n$ respectively.

Input

The first line contains an integer n ($1 \le n \le 500\,000$), denoting the number of vertices in the tree.

The second line contains n integers w_i ($1 \le w_i \le 500\,000$), denoting the weight of vertex i.

Next $n - 1$ lines each contains two integers u, v ($1 \le u, v \le n, u \ne v$), denoting an edge between vertex u and v.

It is guaranteed that input graph forms a tree.

Output

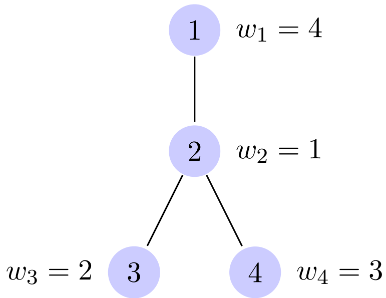
Output $n + 1$ lines, where the i-th line contains an integer denoting the answer when $|V(B)| = i - 1$.

Example

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

Explanation

The input tree is as follows:



The possible best schemes when $|V(B)| = 0, 1, \cdots, n$ are as follows respectively:

- $|V(B)| = 0$: $A = \{1,2,3,4\}, B = \emptyset$, where $D(A) = 9, D(B) = 0, \text{sum} = 9$
- $|V(B)| = 1$: $A = \{1,2,4\}, B = \{3\}$, where $D(A) = 5, D(B) = 0, \text{sum} = 5$
- $|V(B)| = 2$: $A = \{1,2\}, B = \{3,4\}$, where $D(A) = 2, D(B) = 0, \text{sum} = 2$
- $|V(B)| = 3$: $A = \{2\}, B = \{1,3,4\}$, where $D(A) = 1, D(B) = 0, \text{sum} = 1$
- $|V(B)| = 4$: $A = \emptyset, B = \{1,2,3,4\}$, where $D(A) = 0, D(B) = 2, \text{sum} = 2$

Problem C. Sakurada Reset

A huge revolution is coming up in Sakurada. In order to avoid the revolution, Asai Kei is going to take some actions. The director of the revolution will also take some actions to defend Asai Kei. In this problem, each action can be described as a positive integer.

Specifically, Asai Kei will choose a non-empty subsequence as the actual actions to take from a given sequence a. Meanwhile, the director will similarly choose a non-empty subsequence as the defending actions from a given sequence b. And the influence value of a sequence of actions $p_1, p_2, \cdots, p_k$ of length k is defined as follows:

$$\sum_{i=1}^{k} p_i \times 1000^{k-i}$$

Now given action sequences a and b, Asai Kei will choose a non-empty subsequence A from a and the director will choose a non-empty subsequence B from b. Asai Kei can beat the director if and only if the influence value of A is strictly greater than B.

You want to know the number of different schemes for them to choose subsequences A and B so that Asai Kei can beat the director. Two schemes $(A_1, B_1), (A_2, B_2)$ are considered to be different if $A_1 \neq A_2$ or $B_1 \neq B_2$, and two sequences p, q are considered to be different if their influence values are different. Print the answer modulo 998244353 in a single line.

Input

The first line contains two positive integers n, m ($1 \leq n, m \leq 5\,000$), denoting the length of a and the length of b.

The next line contains n positive integers a_i ($1 \leq a_i \leq 100$), denoting the sequence a.

The next line contains m positive integers b_i ($1 \leq b_i \leq 100$), denoting the sequence b.

Output

Output a single line containing a non-negative integer, denoting the answer modulo 998244353.

Example

standard input	standard output
3 5 2 1 2 1 2 2 1 2	22

Explanation

Asai Kei can choose 6 subsequences: (1), (2), (1,2), (2,1), (2,2), (2,1,2), and their influence values are 1, 2, 1002, 2001, 2002, 2001002 respectively. The director can choose 11 subsequences: (1), (2), (1,1), (1,2), (2,1), (2,2), (1,1,2), (1,2,1), (1,2,2), (2,1,2), (2,2,2), and their influence values are 1, 2, 1001, 1002, 2001, 2002, 1001002, 1002001, 1002002, 2001002, 2002002 respectively.

There are $0 + 1 + 3 + 4 + 5 + 9 = 22$ schemes in total that Asai Kei can beat the director.

Problem D. Decimal

Given a positive integer n, determine if $\frac{1}{n}$ is an infinite decimal in decimal base. If the answer is yes, print "Yes" in a single line, or print "No" if the answer is no.

Input

The first line contains one positive integer T ($1 \le T \le 100$), denoting the number of test cases.

For each test case:

Input a single line containing a positive integer n ($1 \le n \le 100$).

Output

Output T lines each contains a string "Yes" or "No", denoting the answer to corresponding test case.

Example

standard input	standard output
2 5 3	No Yes

Explanation

$\frac{1}{5} = 0.2$, which is a finite decimal.

$\frac{1}{3} = 0.333\cdots$, which is an infinite decimal.

Problem E. Escape

Given a maze of size $n \times m$. The rows are numbered $1, 2, \cdots, n$ from top to bottom while the columns are numbered $1, 2, \cdots, m$ from left to right, which means that $(1, 1)$ is the top-left corner and that (n, m) is the bottom-right corner. And for each cell of size 1×1, it is either blank or blocked.

There are a robots above the maze. For i-th robot, it is initially positioned exactly above the cell $(1, p_i)$, which can be described as $(0, p_i)$. And the initial moving direction of the robots are all downward, which can be written as $(1, 0)$ in the vector form.

Also, there are b exits below the maze. For i-th exit, it is positioned exactly below the cell (n, e_i), which can be described as $(n + 1, e_i)$.

Now, you want to let the robots escape from the maze by reaching one of the exits. However, the robots are only able to go straight along their moving directions and can't make a turn. So you should set some turning devices on some blank cells in the maze to help the robots make turns. There are 4 types of turning devices:

- "NE-devices" : make the robots coming from above go rightward, and make the robots coming from right go upward. Coming from left or below is illegal.
- "NW-devices" : make the robots coming from above go leftward, and make the robots coming from left go upward. Coming from right or below is illegal.
- "SE-devices" : make the robots coming from below go rightward, and make the robots coming from right go downward. Coming from left or above is illegal.
- "SW-devices" : make the robots coming from below go leftward, and make the robots coming from left go downward. Coming from right or above is illegal.

For each cell, the number of turning devices on it can not exceed 1. And collisions between the robots are ignored, which allows multiple robots to visit one same cell even at the same time.

You want to know if there exists some schemes to set turning devices so that all the a robots can reach one of the b exits after making a finite number of moves without passing a blocked cell or passing a turning device illegally or going out of boundary(except the initial position and the exit). If the answer is yes, print "Yes" in a single line, or print "No" if the answer is no.

Input

The first line contains one positive integer T $(1 \le T \le 10)$, denoting the number of test cases.

For each test case:

The first line contains four positive integers n, m, a, b $(1 \le n, m \le 100, 1 \le a, b \le m)$, denoting the number of rows and the number of columns in the maze, the number of robots and the number of exits respectively.

Next n lines each contains a string of length m containing only "0" or "1", denoting the initial maze, where cell (i, j) is blank if the j-th character in i-th string is "0", while cell (i, j) is blocked if the j-th character in i-th string is "1".

The next line contains a integers p_i $(1 \le p_i \le m)$, denoting the initial positions $(0, p_i)$ of the robots.

The next line contains b integers e_i $(1 \le e_i \le m)$, denoting the positions $(n + 1, e_i)$ of the exits.

It is guaranteed that all p_is are pairwise distinct and that all e_is are also pairwise distinct.

Output

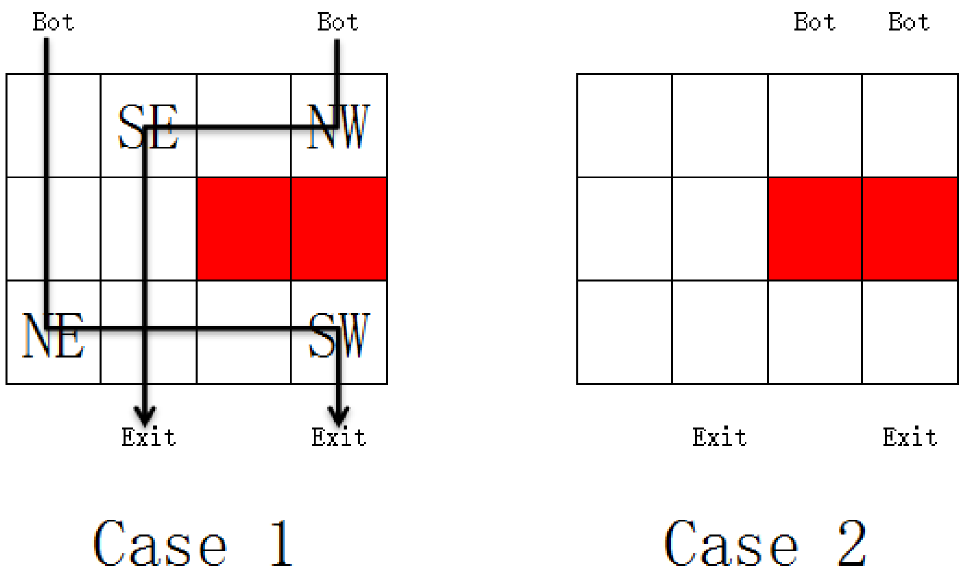
Output T lines each contains a string "Yes" or "No", denoting the answer to corresponding test case.

Example

standard input	standard output
2 3 4 2 2 0000 0011 0000 1 4 2 4 3 4 2 2 0000 0011 0000 3 4 2 4	Yes No

Explanation

Please look at the following illustration.



Problem F. Forest Program

The kingdom of Z is fighting against desertification these years since there are plenty of deserts in its wide and huge territory. The deserts are too arid to have rainfall or human habitation, and the only creatures that can live inside the deserts are the cactuses. In this problem, a cactus in desert can be represented by a cactus in graph theory.

In graph theory, a cactus is a connected undirected graph with no self-loops and no multi-edges, and each edge can only be in at most one simple cycle. While a tree in graph theory is a connected undirected acyclic graph. So here comes the idea: just remove some edges in these cactuses so that the remaining connected components all become trees. After that, the deserts will become forests, which can halt desertification fundamentally.

Now given an undirected graph with n vertices and m edges satisfying that all connected components are cactuses, you should determine the number of schemes to remove edges in the graph so that the remaining connected components are all trees. Print the answer modulo 998244353.

Two schemes are considered to be different if and only if the sets of removed edges in two schemes are different.

Input

The first line contains two non-negative integers n, m ($1 \le n \le 300\,000, 0 \le m \le 500\,000$), denoting the number of vertices and the number of edges in the given graph.

Next m lines each contains two positive integers u, v ($1 \le u, v \le n, u \ne v$), denoting that vertices u and v are connected by an undirected edge.

It is guaranteed that each connected component in input graph is a cactus.

Output

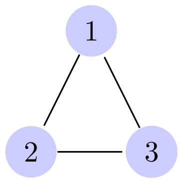
Output a single line containing a non-negative integer, denoting the answer modulo 998244353.

Example

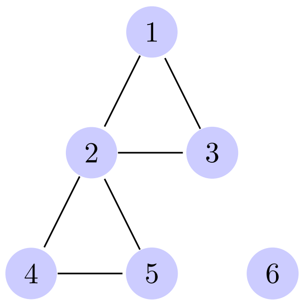
standard input	standard output
3 3 1 2 2 3 3 1	7
6 6 1 2 2 3 3 1 2 4 4 5 5 2	49

Explanation

For the first case, at least one edge should be removed, so the answer is $2^3 - 1 = 7$. The graph is as follows:



For the second case, the graph is as follows:



Problem G. Game on Chessboard

Given a chessboard of size $n \times n$. The rows are numbered $1, 2, \cdots, n$ from top to bottom while the columns are numbered $1, 2, \cdots, n$ from left to right, which means that $(1, 1)$ is the top-left corner and that (n, n) is the bottom-right corner.

There are some chesses in the chessboard. Each chess has a color, white or black, and a cost value w. The number of chesses in one cell does not exceed 1.

You want to remove these chesses. In each time, you can choose one white chess u and one black chess v and then remove them with cost $|w_u - w_v|$. Moreover, one chess can be chosen in this operation if and only if there are no other chesses in its bottom-left area. Formally, one chess positioned at (x, y) can be chosen if and only if there are no other chesses (x_1, y_1) that $x_1 \geq x$ and $y_1 \leq y$.

However, you may be not able to choose two chesses holding the restriction sometimes. So you can also choose exactly one chess x arbitrarily and remove it with cost w_x. Moreover, you can do this operation in any situation as long as there is at least one chess on chessboard.

Now you want to remove all the chesses with the minimum total cost. Print the mimimum cost in one line.

Input

The first line contains one positive integers n $(1 \leq n \leq 12)$, denoting the size of the chessboard.

Next n lines each contains a string of length n containing only "."", "W" or "B", denoting the initial chessboard, where cell (i, j) is blank if the j-th character in i-th string is ".", while "W" represents a white chess and "B" represents a black chess.

Next n lines each contains n non-negative integers $w_{i,j}$ $(0 \leq w_{i,j} \leq 10^6)$, denoting the cost values, where the j-th number in i-th line represents the cost value of the chess positioned at (i, j).

It is guaranteed that $w_{i,j}$ is positive if and only if there exists a chess at (i, j).

Output

Output a single line containing a non-negative integer, denoting the mimimum cost.

Example

standard input	standard output
4 WBB. BWBW WBWW BBBW 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1	3

Explanation

One possible scheme is as follows:

1. remove $(4, 1)$ with cost 1

W	B	B	
B	W	B	W
W	B	W	W
	B	B	W

2. remove $(2, 1)$ with cost 1

W	B	B	
	W	B	W
W	B	W	W
	B	B	W

3. remove $(2, 4)$ with cost 1

W	B	B	
	W	B	
W	B	W	W
	B	B	W

4. remove $(4, 2)$ and $(3, 1)$ with cost $|1 - 1| = 0$

W	B	B	
	W	B	
	B	W	W
		B	W

5. remove $(1, 1)$ and $(3, 2)$ with cost $|1 - 1| = 0$

	B	B	
	W	B	
		W	W
		B	W

6. remove $(2, 2)$ and $(4, 3)$ with cost $|1 - 1| = 0$

	B	B	
		B	
		W	W
			W

7. remove $(1, 2)$ and $(3, 3)$ with cost $|1 - 1| = 0$

		B	
		B	
			W
			W

8. remove $(2, 3)$ and $(4, 4)$ with cost $|1 - 1| = 0$

		B	
			W

9. remove $(1, 3)$ and $(3, 4)$ with cost $|1 - 1| = 0$

The total cost is 3.

Problem H. Houraisan Kaguya

Houraisan Kaguya is fighting against Fujiwara no Mokou over the moon. Suddenly, Mokou launches a spell “Imperishable Shooting”(just a programming problem, believe it or not) to attack Kaguya, which is as follows.

Given a prime number p and n positive integers $a_1, a_2, \cdots, a_n$ which are strictly less than p.

For two integers a, b $(0 \le a, b < p)$, we say a is representable by b if and only if there exists a positive integer x that $b^x \equiv a (\bmod\ p)$. Furthermore, we define $f(a, b)$ as the minimum positive integer u that a^u modulo p is representable by b. If no such u exists, $f(a, b)$ is specially defined as 0.

The problem is to determine the value of following formula.

$$\left(\sum_{i=1}^{n} \sum_{j=1}^{n} f(a_i, a_j) \times f(a_j, a_i) \right) \mod p$$

Please help Kaguya solve it so that Kaguya can give Mokou the sixth puzzle in the next round.

Input

The first line contains two positive integers n, p $(1 \le n \le 100\,000, 2 \le p \le 10^{18})$, denoting the number of given integers and the given prime number.

The next line contains n positive integers a_i $(1 \le a_i < p)$, denoting the given integers.

Output

Output a single line containing a non-negative integer, denoting the answer.

Example

standard input	standard output
4 5 1 2 3 4	4

Explanation

When p equals 5, the values of f are as follows, where a is enumerated by rows and b is by columns.

f b / a	1	2	3	4
1	1	1	1	1
2	4	1	1	2
3	4	1	1	2
4	2	1	1	1

So the sum is $1 + 4 + 4 + 2 + 4 + 1 + 1 + 2 + 4 + 1 + 1 + 2 + 2 + 2 + 2 + 1 = 34$, and the answer to this problem is $34 \bmod 5 = 4$.

Problem I. Invoker

In dota2, there is a hero named Invoker. He has 3 basic skills in the game, which are Quas, Wex and Exort. Once he launches a basic skill, he will gain the corresponding element, where Quas gives "Q", Wex gives "W" and Exort gives "E".

Invoker can't have more than 3 elements simultaneously. If he launches a basic skill when he already owns 3 elements, he will get the corresponding element and lose the element he gained the earliest.

As can be seen, there are 10 unordered combinations of 3 elements in 3 types, each represents a special skill, which are as follows:

- Cold Snap: unordered element combination "QQQ", denoted by "Y"
- Ghost Walk: unordered element combination "QQW", denoted by "V"
- Ice Wall: unordered element combination "QQE", denoted by "G"
- EMP: unordered element combination "WWW", denoted by "C"
- Tornado: unordered element combination "QWW", denoted by "X"
- Alacrity: unordered element combination "WWE", denoted by "Z"
- Sun Strike: unordered element combination "EEE", denoted by "T"
- Forge Spirit: unordered element combination "QEE", denoted by "F"
- Chaos Meteor: unordered element combination "WEE", denoted by "D"
- Deafening Blast: unordered element combination "QWE", denoted by "B"

When Invoker owns 3 elements, he can launch the invoking skill, denoted by "R", to gain the special skill according to the elements he currently owns. After invoking, the elements won't disappear, and the chronological order of the 3 elements won't change.

Now given a sequence of special skills, you want to invoke them one by one with using the minimum number of basic skills(Q,W,E) and invoking skill(R). Print the minimum number in a single line.

At the beginning, Invoker owns no elements. And you should re-invoke the special skills even if you have already invoked the same skills just now.

Input

Input a single line containing a string s $(1 \le |s| \le 100\,000)$ that only contains uppercase letters in $\{B, C, D, F, G, T, V, X, Y, Z\}$, denoting the sequence of special skills.

Output

Output a single line containing a positive integer, denoting the minimum number of skills to launch.

Example

standard input	standard output
XDTBVV	15

Explanation

One possible scheme is QWWREERERWQRQRR.

Q	W	W	R	E	E	R	E	R	W	Q	R	Q	R	R
Q	QW	QWW	X	WWE	WEE	D	EEE	T	EEW	EWQ	B	WQQ	V	V

Problem J. MUV LUV EXTRA

One day, Kagami Sumika is stuck in a math problem aiming at calculating the length of a line segment with given statements and constraints. Since Sumika has no idea about it, she takes out a ruler and starts to measure the length. Unfortunately, the answer is an infinite decimal and she only got the first some digits of the answer from the ruler.

Sumika guesses that the answer is a rational number, which means that there exists two integers p, q that the answer equals $\frac{q}{p}$. In this situation, the answer can be expressed as an infinite repeated decimal. For example, $\frac{1}{2} = 0.500\cdots$, $\frac{1}{3} = 0.333\cdots$, $\frac{9}{10} = 0.8999\cdots$, $\frac{36}{35} = 1.0285714285714\cdots$. Sumika wants to guess the original number from the digits she got. Note that a number may has more than one way to be expressed such as $1.000\cdots = 0.999\cdots$. Sumika won't transform the digits she got to another form when guessing the original number.

Furthermore, Sumika relizes that for a repeating part, either too long or the appeared length too short will make the result unreliable. For example, if the decimal she measured is 1.0285714285714, it is obviously unreliable that the repeating part is "0285714285714", since it is too long, or "428571", since the appeared length is too short, which equals 7, the length of "4285714". In this case, the best guess is "285714", whose length is 6 and the appeared length is 12. So formally, she defines the reliability value of a repeating part, whose length is l and the appeared length is p, as the following formula:

$$a \times p - b \times l$$

Where a and b are given parameters.

Last but not least, you can ignore the integer parts of the decimal. It is just for restoring the scene. And the repeating part you guess should be completely repeated at least once and is still repeating at the end currently.

Please help Sumika determine the maximum reliability value among all repeating parts.

Input

The first line contains two positive integers a, b ($1 \le a, b \le 10^9$), denoting the parameters.

The next line contains a string s ($1 \le |s| \le 10^7$) in decimal form, denoting the first some digits of the accurate result.

It is guaranteed that there is exactly one decimal point in s and s is a legal non-negative decimal without leading "-"(the minus sign).

Output

Output a single line containing an integer, denoting the maximum reliability value.

Example

standard input	standard output
5 3 1.1020	9
2 1 12.1212	6

Explanation

For the first case, all possible repeating parts are as follows:

repeating part	length	appeared length	reliability value
0	1	1	$5 \times 1 - 3 \times 1 = 2$
20	2	2	$5 \times 2 - 3 \times 2 = 4$
02	2	3	$5 \times 3 - 3 \times 2 = 9$
020	3	3	$5 \times 3 - 3 \times 3 = 6$
1020	4	4	$5 \times 4 - 3 \times 4 = 8$

For the second case, all possible repeating parts are as follows:

repeating part	length	appeared length	reliability value
2	1	1	$2 \times 1 - 1 \times 1 = 1$
12	2	4	$2 \times 4 - 1 \times 2 = 6$
21	2	3	$2 \times 3 - 1 \times 2 = 4$
212	3	3	$2 \times 3 - 1 \times 3 = 3$
1212	4	4	$2 \times 4 - 1 \times 4 = 4$

Problem K. MUV LUV UNLIMITED

There are few entertainments in *United Nations 11th Force, Pacific Theater, Yokohama Base*, the only pastime for squad 207 is gathering in PX to play games after supper. However, whatever they play, Shirogane Takeru is always the loser. So he decides to use the game theory knowledge from another world to become the winner. According to the knowledge he has learnt, Takeru introduces his army friends a game:

Given a rooted tree of size n, whose root is vertex 1. Two players do operations on the tree alternately. In each operation, a player should choose several (at least one) leaf vertices (which have no children vertices) and remove them from the tree. As can be seen, there might be some new leaf vertices after one operation. The player who cannot make a move in his/her turn loses the game.

But unfortunately, Takeru doesn't master the knowledge skillfully, so he has no idea whether the first player will win if the two players are playing optimally. Please help him determine that.

Assume that the two players are playing optimally to make themselves win, print "Takeru" in a single line if the first player will win, or print "Meiya" otherwise.

Input

The first line contains one positive integer T, denoting the number of test cases.

For each test case:

The first line contains one positive integer n ($2 \le n \le 10^6$), denoting the size of the given tree.

The next line contains $n-1$ positive integers p_i ($1 \le p_i \le n$), where i-th integer denotes the parent vectex of vectex $i+1$.

It is guaranteed that the sum of n among all cases in one test file does not exceed 10^6.

Output

Output T lines each contains a string "Takeru" or "Meiya", denoting the answer to corresponding test case.

Example

standard input	standard output
2 3 1 1 4 1 2 3	Takeru Meiya

Explanation

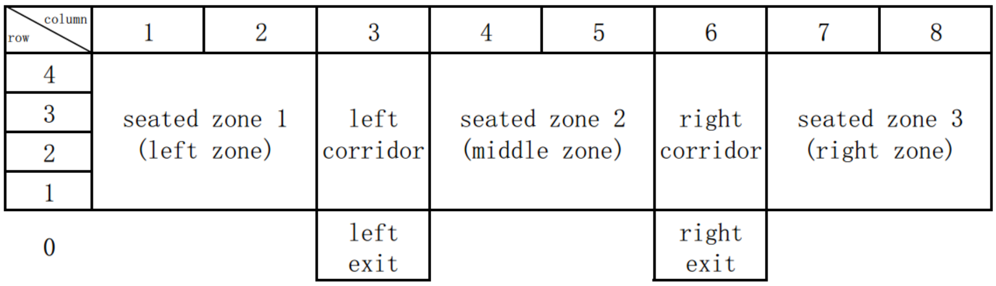
For the first case, the first player can remove vertex 3 firstly. As a result, the second player has no choice but to remove vertex 2. Finally, the first player removes vertex 1 and becomes the winner.

For the second case, the two players can only remove exactly one vertex in each operation alternately. As can be seen, the winner, who removes vertex 1, will be the second player.

Problem L. MUV LUV ALTERNATIVE

In order to cover *the Tactical Armored Squadron* withdrawing, *the IJN Combined Fleet's 2nd Squadron* is attracting attacks from *Lux*. After a while, the fleet is heavily damaged and some ships have been on fire. The soldiers on one fired ship are trying escaping from the cabin and reaching the board.

The cabin can be abstracted into a grid of $n \times m$, where rows are numbered $1, 2, \cdots, n$ from bottom to top and columns are numbered $1, 2, \cdots, m$ from left to right. The cabin is divided into 3 seated zones by 2 vertical corridors of width 1, where the left zone is of width l_1, the middle zone is of width l_2 and the right zone is of width l_3. Also, for the two corridors, one is between the left zone and the middle zone and the other one is between the middle zone and the right zone so that $l_1 + l_2 + l_3 + 2 = m$ always holds. Let's use (x, y) to denote the location of the cell in the x-th row and the y-th column. Below the two corridors are the left exit and the right exit, which can be considered to be positioned at $(0, l_1 + 1), (0, l_1 + l_2 + 2)$ respectively. Following is the illustration when $n = 4, l_1 = l_2 = l_3 = 2$.



There are k soldiers inside the cabin, each soldier is in a unique cell (x_i, y_i) in one of the three seated zones initially. A soldier in seated zones can only move leftward or rightward, while one in corridors can move not only leftward or rightward, but also upward or downward. And one soldier can stay still or move to an adjacent cell in one of the allowed directions according to its current position at every moment. Two cells are adjacent if and only if they share an edge. Each cell mustn't contain more than one soldier after each moment's movement.

For maintaining the order of escape, the soldiers who are initially in the left zone can only go to the left exit, while ones who are initially in the right zone can only go to the right exit. For ones who are initially in the middle zone, they can go to either the left exit or the right exit.

You want to know the minimum possible time that all the soldiers have reached one of the two exits.

Input

The first line contains five positive integers n, l_1, l_2, l_3, k $(1 \le k \le 100\,000, 1 \le n, l_1, l_2, l_3 \le 10^9)$, denoting the number of rows, the three widths of the left zone, the middle zone, the right zone, and the number of soldiers respectively.

Next k lines each contains three positive integers a_i, x_i, y_i $(a_i \in \{1, 2, 3\}, 1 \le x_i \le n, 1 \le y_i \le l_1 + l_2 + l_3 + 2, y_i \ne l_1 + 1, y_i \ne l_1 + l_2 + 2)$, denoting the initial position of a soldier, where a_i denotes the zone index and x_i, y_i denotes the row index and the column index respectively.

It is guaranteed that input k positions are pairwise distinct and that (x_i, y_i) is indeed in the zone a_i.

Output

Output a single line containing a positive integer, denoting the minimum possible time.

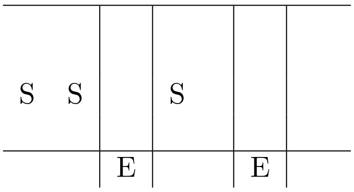
Example

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

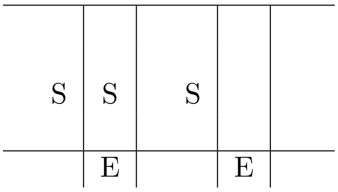
Explanation

One possible scheme is as follows, where "S", "E" denote soldiers and exits respectively:

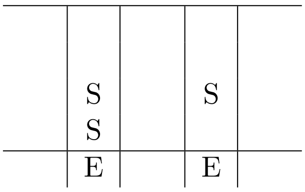
1. Moment 0: Initial status



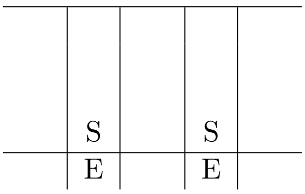
2. Moment 1: three soldiers move to $(2, 2), (2, 3), (2, 5)$ respectively.



3. Moment 2: three soldiers move to $(2, 3), (1, 3), (2, 6)$ respectively.



4. Moment 3: three soldiers move to $(1, 3), (0, 3), (1, 6)$ respectively, where the second soldier has reached the left exit.



5. Moment 4: the remaining two soldiers move to $(0, 3), (0, 6)$ respectively, where the three soldiers have all reached the exits. So the cost time is 4 during the escape.