Problem A. Hi Young Guangzhou

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	1024 megabytes

Guangzhou, also known as Canton and the "City of Rams", is a vibrant metropolis with over 2,200 years of history. As a vital starting point of the ancient Maritime Silk Road, it has been a major port for international trade since the Qin and Han dynasties.

In the year just passed, the 15th National Games concluded successfully in Guangzhou. The spirit of hard work of the athletes still thrives in this city.

Picture 1: "Dawanji" at Guangzhou Baiyun International Airport

Now, you and your teammate are making a plan to travel around Guangzhou. There are n tourist attractions you want to visit. Also, you plan to stay in Guangzhou for m days.

There are $n-1$ relationships among the attractions. Each relationship connects two attractions, indicating that they are closely related in history or scenery. Additionally, any two different attractions can reach each other through one or more relationships, meaning that the relationships form a tree.

You decide to visit each attraction **exactly once**. Let b_i be the day on which the i-th attraction is visited. Your plan should satisfy:

- The hotel you booked on the i-th day is near the attraction a_i. That means you should visit the attraction a_i on the day i. In other words, $b_{a_i} = i$ for every $i = 1, 2, \ldots, m$.
- For each attraction i ($1 \le i \le n$), the number of relationships connecting it to other attractions that are visited on the same day (the b_i-th day) is no less than the number of such relationships for any other day. In other words, for every $u = 1, 2, \ldots, n$ and every $t = 1, 2, \ldots, m$, the following must hold $\sum_{v:(u,v)\in E}[b_v = b_u] \ge \sum_{v:(u,v)\in E}[b_v = t]$, where E is the set of all relationships.

So, now make your plan! Construct any possible $b_1, b_2, \ldots, b_n$ satisfying the above conditions, or show that it is impossible to do so.

Input

Each test contains multiple test cases. The first line contains one integer t ($1 \le t \le 10^5$), indicating the number of test cases. The description of the test cases follows.

The first line contains two integers n, m ($1 \le m \le n \le 10^5$, $1 \le \sum n, \sum m \le 10^6$), indicating the number of attractions and the number of days you will stay.

The second line contains m integers $a_1, a_2, \ldots, a_m$ ($1 \le a_i \le n$, $a_i \ne a_j$ for $1 \le i < j \le m$), indicating you must visit attraction a_i on the day i.

The next $n - 1$ lines, each line contains two integers u_i, v_i ($1 \le u_i, v_i \le n$), indicating a relationship.

Output

For each test case, if you can't make a valid plan, output `NO`.

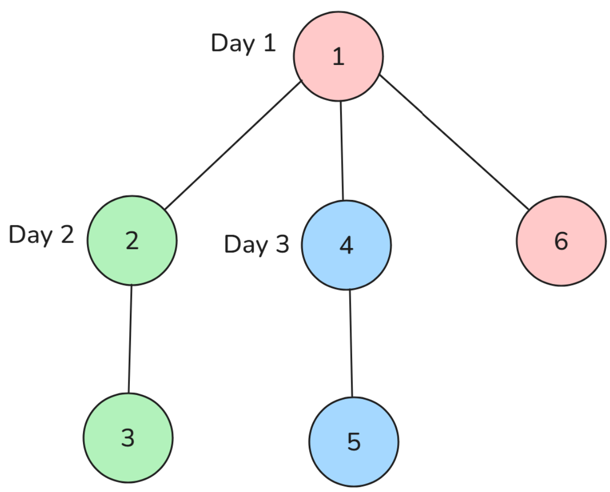
Otherwise, output two lines. The first line contains one word `YES`, and the second line contains $b_1, b_2, \ldots, b_n$ ($1 \le b_i \le m$) indicating your plan.

If there are multiple valid answer, you may output any of them.

Examples

standard input	standard output
4 1 1 1 4 2 2 3 1 2 2 3 3 4 3 2 1 3 1 2 2 3 6 3 1 2 4 1 2 2 3 1 4 4 5 1 6	YES 1 YES 1 1 2 2 NO YES 1 2 2 3 3 1

Note



Picture 2: Example figure for Test case 4

Problem B. Welcome to Math Kingdom!

Input file:	`standard input`
Output file:	`standard output`
Time limit:	4 seconds
Memory limit:	1024 megabytes

You know you are now in HSFZ, the famous math kingdom. So you have to solve some math problems.

You are given two positive integers p, q. Your task is to construct positive integers a, b, c, d such that $1 \le a, b, c, d \le 10^7$ and $\frac{a^3 + b^3}{c^3 + d^3} = \frac{p}{q}$.

Input

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

For each test cases, only one line contains two integers p, q ($1 \le p, q \le 10^4$).

Output

For each test case, output one line contains four integers a, b, c, d ($1 \le a, b, c, d \le 10^7$).

It can be proved that a solution always exist under above constraints.

Examples

standard input	standard output
2 1 1 9 28	1 1 1 1 1 2 1 3

Problem C. Shiroi Album II

Input file: `standard input`
Output file: `standard output`
Time limit: 6 seconds
Memory limit: 1024 megabytes

Kamome is currently researching triangular paths on **weighted undirected graphs**. More specifically, a path is called *triangular* if and only if:

- The length of this path is no more than 2, or
- For any three different edges e_1, e_2, and e_3 on this path, it is always true that $w(e_1)$, $w(e_2)$, and $w(e_3)$ can form the three sides of a triangle.

Recall that positive integers x, y, and z can form the sides of a triangle if and only if $x < y + z$, $y < z + x$, and $z < x + y$.

Kamome has provided you with a weighted undirected graph. You need to determine for each pair of points (u, v) whether there exists a triangular path starting from u and ending at v. We assume that there is always a triangular path from each node to itself.

Picture 3: A part of the "triangle" at Peking University

Input

Each test contains multiple test cases. The first line contains one integer t ($1 \le t \le 10^5$), indicating the number of test cases. The description of the test cases follows.

The first line contains two integers n, m ($1 \le n, m \le 3000$, $1 \le \sum n^2, \sum m^2 \le 3000^2$), indicating the number of nodes and edges of this graph.

The next m lines, each line contains three integers u_i, v_i, w_i ($1 \le u_i, v_i \le n$, $1 \le w_i \le 10^9$), indicating edge (u_i, v_i) with weight w_i.

The graph is **maybe disconnected** or include **self-loop** or **multiple edge**.

Output

For each test case, output n lines, each line contains a string of n characters, where the j-th characters is $s_{i,j}$, if it is 0 indicating there is no triangular path start from i and end at j, 1 otherwise.

Examples

standard input	standard output
2 4 3 1 2 2 2 3 2 3 4 3 8 7 1 2 3 2 3 4 3 4 1 3 5 2 2 6 2 6 7 4 6 8 5	1111 1111 1111 1111 11101110 11111111 11111111 01111000 11111000 11100111 11100111 01100111

Problem D. Judgment of Mahou Shoujo

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	1024 megabytes

There are n Mahou Shoujo sitting in a circle, numbered clockwise from 1 to n. Among them, some are actually $Majo$.

For the next $n - 3$ days, the following events occur one by one:

- In the night, **exactly one** Majo wakes up and kills the first living Mahou Shoujo on her left or right (this person could also be a Majo);
- In the morning, everyone wakes up and discovers the dead Mahou Shoujo.

Kamome, as the judge of the Mahou Shoujo, needs to find, after each morning when a dead Mahou Shoujo is discovered, the minimum number of Majo that could have been present on the first day.

Picture 4: The Judgement

Input

Each test contains multiple test cases. The first line contains one integer t ($1 \le t \le 10^5$), indicating the number of test cases. The description of the test cases follows.

The first line contains two integers n ($4 \le n \le 2 \times 10^5$, $1 \le \sum n \le 10^6$), indicating the number of Mahou Shoujo.

The second line contains $n - 3$ integers p_i ($1 \le p_i \le n$, $p_i \ne p_j$ for $1 \le i < j \le n - 3$) indicating the Mahou Shoujo who died on the i-th day.

Output

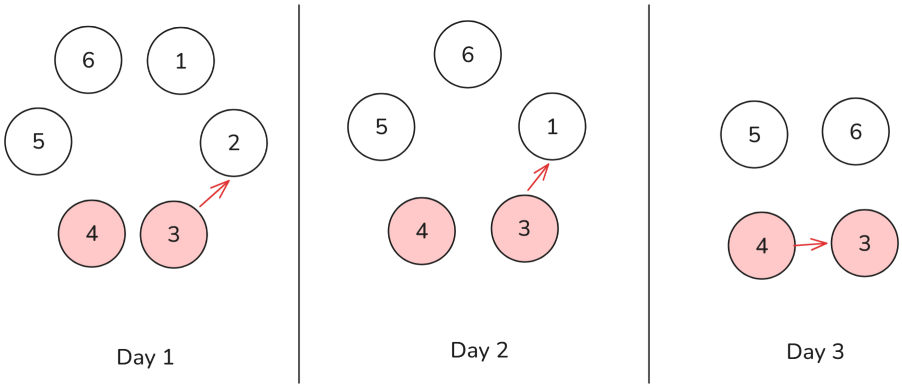
For each test case, output one line containing $n - 3$ integers, indicating the minimum number of Majo that could have been present on the first day after discovering the dead Mahou Shoujo on the i-th day.

Examples

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

Note

For the second test case, on the third day, at least 2 Majo could have been present on the first day. A possible example is 3 and 4 being the Majo, where 3 kills 2 on the first day, 3 kills 1 on the second day, and 4 kills 3 on the third day.



Picture 5: A possible example

Problem E. Hoshizora

Input file: standard input
Output file: standard output
Time limit: 6 seconds
Memory limit: 1024 megabytes

Kamome is very interested in observing meteors. He created an infinitely large two-dimensional model that simulates the movement of meteors (even though it doesn't actually simulate anything). There are n meteor models in this simulation, and the weight of the i-th meteor model is m_i. Each meteor model moves at a constant speed along a fixed straight line (it might also be stationary).

It has been a long time since Kamome ran this model. One day, Kamome had a sudden idea and took two photos of the current model. Taking Kamome's position as the origin, for each photo i ($i = 1, 2$), the j-th meteor model was located at the position $(x_{i,j}, y_{i,j})$ on the plane.

You can calculate the trajectory of each meteor model, but that's not enough! Kamome wants you to calculate, for each meteor model i, the maximum gravitational force that any other meteor model exerts on it during the entire time the model has been running (you can consider this as an infinite amount of time before and after the photos were taken).

Note that the usual gravitational formula can be rewritten as $F = G\frac{m_1 m_2}{r^2} = \frac{G}{\frac{1}{m_1}\frac{1}{m_2}r^2}$. To avoid dealing with some corner cases, Kamome instead gives you $m'_i = \frac{1}{m_i}$ as input, and you only need to output the result of dividing the answer by G and then taking the reciprocal. That is, for each i, compute $\min_{j \neq i} \min_{t \in \mathbb{R}} m'_i m'_j D_t^2(i, j)$, where $D_t(i, j)$ is the distance between meteor models i and j at time t, and t can be any real number (negative, positive, or zero).

Picture 6: Hoshizora

Input

Each test contains multiple test cases. The first line contains one integer t ($1 \leq t \leq 10^4$), indicating the number of test cases. The description of the test cases follows.

The first line contains two integers n ($2 \leq n, \sum n \leq 10^5$), indicating the number of meteor models.

The second line contains n integers m'_i ($1 \leq m'_i \leq 100$), its definition is as stated above.

The next n lines, each line contains four integers $x_{1,i}, y_{1,i}, x_{2,i}, y_{2,i}$ ($-10^8 \le x_{1,i}, y_{1,i}, x_{2,i}, y_{2,i} \le 10^8$, $0 \le |x_{1,i} - x_{2,i}|, |y_{1,i} - y_{2,i}| \le 5$), indicating the position of the i-th meteor model in the two photos.

Output

For each test case, output one line of n real numbers, as defined above.

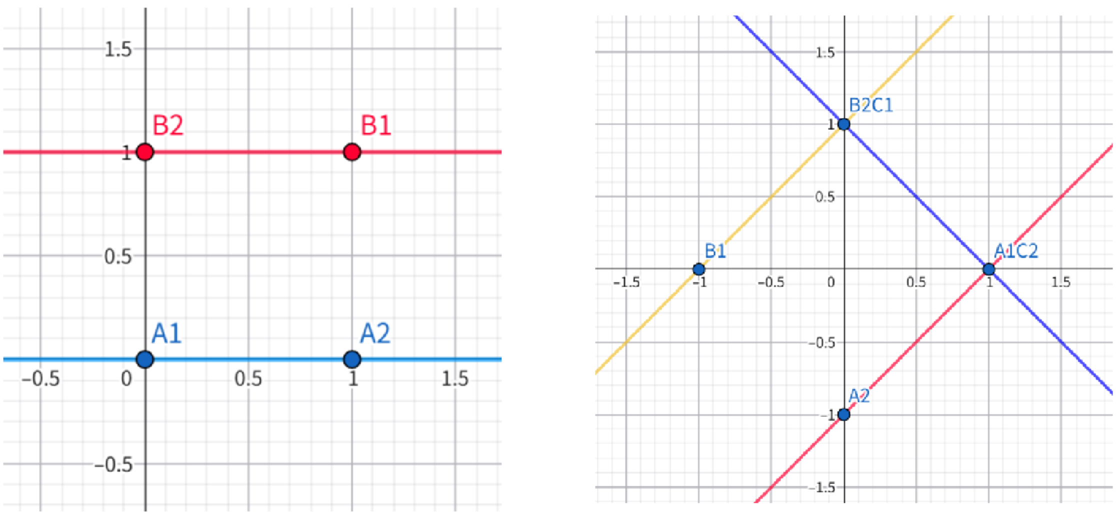
Assume your answer be x and jury's answer be y, your answer is considered correct if and only if $\frac{|x-y|}{\max(y,1)} \le 10^{-6}$.

Examples

standard input	standard output
2 2 1 2 0 0 1 0 1 1 0 1 3 1 2 3 1 0 0 -1 -1 0 0 1 0 1 1 0	2.000000000 2.000000000 3.000000000 4.000000000 3.000000000

Note

For the first test case, the maximum gravitational force of meteor model 1 and meteor model 2 occurs when they reach $(0.5, 0)$ and $(0.5, 1)$ respectively. At this point, the answer is $m'_1 m'_2 d^2_{\min} = 2$.



Picture 7: Two test cases

Problem F. Sakura no Mahou

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	1024 megabytes

Una is walking on a path with n cherry trees, fragments of spring falling gently from the sky. On the i-th cherry tree, there are a_i blossoms.

Una really misses her best friend, Kamome. She remembers that, on the day Kamome left this town, Kamome gave Una an integer sequence $b_1, \cdots, b_n$. Kamome told Una that, when $a_i \equiv b_i \pmod 2$ for every $i \in [1, n]$, the magic power in the cherry trees will be activated and they will meet again.

Una's wish was so strong that her imagination became reality. She gained the power of magic to make flowers bloom. On each wave of her wand, she can choose a **contiguous subsequence** of cherry trees $[l, r]$, and change a_i to $\sum_{j=l}^{i} a_j$ for every $i \in [l, r]$.

Gifted by magic, Una can perform magic **any number of times** every day, but she must not perform magic on the same tree **twice or more** on the same day, otherwise the tree will die due to an excessive amount of magic power. (In other words, the subsequences chosen on the same day must not intersect.)

Una has not seen Kamome for ten years, and she does not want to wait any longer. Una wants to know the minimum number of days for them to meet (i.e., make $a_i \equiv b_i \pmod 2$ for each $i \in [1, n]$), and the magic she should perform on each day to achieve this minimum. Note that this might be impossible, in which case you should tell her so. (Please refer to the Output section for more details.)

Input

Each test contains multiple test cases. The first line contains the number of test cases t ($1 \le t \le 2.5 \times 10^4$). The description of the test cases follows.

The first line of each test case contains a single integer n ($1 \le n \le 10^3$, $\sum n^2 \le 10^6$) , the number of cherry trees.

The second line of each test case contains n integers $a_1, \cdots, a_n$ ($0 \le a_i \le 1$), the number of blossoms on each tree currently.

The third line of each test case contains n integers $b_1, \cdots, b_n$ ($0 \le b_i \le 1$), the sequence Kamome gave to Una.

Output

For each test case, if it is impossible to make Kamome and Una meet, print `-1`. Otherwise, print a single integer, k, representing the minimum number of days required for Una to meet Kamome.

On the i-th line of the next k lines, you should give instructions to Una on the magic she should perform on the i-th day. You should print $2c + 1$ integers $c_i, l_{i,1}, r_{i,1}, l_{i,2}, r_{i,2}, \cdots, l_{i,c}, r_{i,c}$, where c is the number of magic operations she should perform on the i-th day, and $[l_{i,j}, r_{i,j}]$ represents the j-th contiguous subsequence she should choose on the i-th day. You should guarantee that $r_i < l_{i+1}$ for every $1 \le i \le c - 1$ and $l_i \le r_i$ for every $1 \le i \le c$.

If there are multiple solutions, you may output any one of them.

Examples

standard input	standard output
3	-1
5	1
1 0 1 0 1	1 2 3
0 1 0 1 0	2
3	2 1 4 6 9
1 1 0	2 1 2 6 7
1 1 1	
10	
1 0 0 1 0 1 0 0 1 0	
1 0 1 0 0 1 0 1 0 0	

Note

In the third test case, at the end of the first day, the sequence a is $[\underline{1,1,1,2},0,\underline{1,1,1,2},0]$.

At the end of the second day, the sequence a is $[\underline{1,2},1,2,0,\underline{1,2},1,2,0]$.

It is clear that after two days, $a_i \equiv b_i \pmod 2$. It can be proven that this is not achievable within one day.

Problem G. Stop Plang!

Input file:	`standard input`
Output file:	`standard output`
Time limit:	6 seconds
Memory limit:	1024 megabytes

This is an interactive problem.

In the Plang Province, $n = 300$ students just participated in Plang OI 2026. Among them, some won gold medals while others did not.

Students in Plang Province have a strange habit. If they didn't win gold medal, they will honestly say that they haven't won. However, if they have won, they will say they have won with $\frac{1}{2}$ probability, while the other $\frac{1}{2}$ probability is that they say they haven't. Note that each response is random and has no relation to the previous one.

You, as the coach of Plang Province, want to know for each student whether they have won gold. Then you can let Una perform the following actions:

- Choose some distinct students and ask Una to ask each of them whether they have won the gold medal. Una will tell you the number of people who answered "Yes".

The gold medal line of Plang OI 2025 is 571, so you should find the answer in no more than 568 queries **on average**.

Due to the limitations of the Plang OI, you may assume that there're exactly 48 students won gold medal.

Picture 8: A coach asking a student how many points he has hidden.

Interaction Protocol

This is an interactive problem. Remember to flush the output buffer after every print. To flush your output, you can use:

- `fflush(stdout)` or `cout.flush()` in C/C++;
- `System.out.flush()` in Java and Kotlin;
- `sys.stdout.flush()` in Python.

First, you should read a integer T ($T = 20$) indicating the number of test cases.

For each test case, you should read an integer n ($n = 300$), indicating the number of students.

To make a query, you should output one line formatted as "`?` k p_1 $p_2 \ldots$ p_k" ($1 \le p_i \le n$, $p_i \neq p_j$ for $1 \le i < j \le k$).

Then, read an integer indicating the answer Una found. If your query is invalid, or if you make more than $568T$ queries in total, the jury program will output -1. After reading -1, you should exit immediately to avoid undefined behavior.

If you get the answer, you should output one line formatted as "! $s_1 s_2 \ldots s_n$", where $s_i = 1$ indicating the i-th student get gold, 0 otherwise.

Then, read one word OK or WA indicating whether your answer is correct. If your program reading WA, you should exit immediately to avoid undefined behavior.

Note that the grader is **not adaptive**, that is the answer won't change after your queries. There are at most 10 tests.

Examples

standard input	standard output
1	
2	
	? 2 1 2
0	
	? 1 1
0	
	? 1 2
1	
	! 01
OK	

Note

Note that the example is for reference only, it does not satisfy $T = 20$ and $n = 300$, and it will not appear in the final tests.

A testing tool is provided to help contestants develop and test their solutions. You can download this tool from the attachments. Executing the tool with a "-h" option should describe how to use the tool. The testing tool will only implement some test scenarios and only some functionality of the real judge program.

Problem H. Shiratama

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	1024 megabytes

Kamome, as a big fan of Shiratama, is trying to create some paintings in the style of Shiratama.

... Well, the problem setter is too lazy to write a story, so let's get straight to the problem.

Kamome's painting can be abstracted into $p_1, p_2, \ldots, p_{2n}$, which is a permutation of $1 \sim 2n$. You need to sort p by performing no more than $2n$ operation 1 and no more than $2n^2$ operation 2.

- Operation 1: For each $i = 1, 3, ..., 2n-1$, **simultaneously** swap p_i and p_{i+1}.
- Operation 2: Kamome selects an **even** number i ($1 \le i \le 2n$) and swaps p_i and p_{i+1}. Note that we consider $p_{2n+1} = p_1$.

However, sometimes it is impossible to sort this permutation using the above operations, if so, report it.

Input

Each test contains multiple test cases. The first line contains one integer t ($1 \le t \le 10^5$), indicating the number of test cases. The description of the test cases follows.

The first line contains a single integer n ($1 \le n \le 100$, $\sum n^2 \le 10^6$), indicating the length of the permutation.

The second line contains $2n$ integers $p_1, p_2, \ldots, p_{2n}$ ($1 \le p_i \le 2n$, $p_i \ne p_j$ for $1 \le i < j \le 2n$), indicating the permutation.

Output

For each test case, if it is impossible to sort the permutation, output one word `NO`.

Otherwise, your output contains three lines. The first line contains one word `YES`, the second line contains one integer k ($0 \le k \le 2n^2 + 2n$), and the third line contains k integers $a_1, a_2, \ldots, a_k$ ($a_i \in \{1\} \cup \{2, 4, \ldots, 2n\}$), indicating the operations you perform. If $a_i = 1$ that means you perform an operation 1, otherwise that means you perform an operation 2 and choose $x = a_i$.

Note that you can't use more than $2n$ operation 1 and $2n^2$ operation 2.

If there are multiple possible answer, you may output any of them.

Examples

standard input	standard output
3 2 1 3 4 2 2 2 4 1 3 3 1 2 4 3 5 6	NO YES 2 2 1 YES 7 2 4 1 6 1 2 4

Note

For test case 2, the change of permutation is shown as follows:

$$[2, \underline{4, 1}, 3] \to [\underline{2, 1}, \underline{4, 3}] \to [1, 2, 3, 4]$$

Problem I. Embedding Trees

Input file:	`standard input`
Output file:	`standard output`
Time limit:	12 seconds
Memory limit:	1024 megabytes

Note the unusual definition of subtree and diameter in this problem.

Here are some definitions:

- The size of a tree T is the number of vertices in it, that is, $|V(T)|$.
- The degree of a vertex i in tree T is the number of edges connected to it, denoted by deg_i.
- A tree T' is a subtree of tree T if and only if:
  - For all vertices $v \in V(T')$, $v \in V(T)$.
  - For all edges $e \in E(T')$, $e \in E(T)$.
- Let S_T be the set of subtrees of T.
- The diameter of a tree T is the size of the largest subtree of T such that $\max(deg_i) \leq 2$ in the subtree.
- Let R_x be the set of the largest trees such that:
  - $\max(deg_i) \leq 3$.
  - The diameter of the tree is $2x$.
- For a tree T, let $f(T)$ be the largest k such that $S_T \cap R_k \neq \emptyset$. If there's no such k, $f(T) = 0$.

You are given a tree T with n nodes. Count sum of $f(T')$ for all subtrees T' of T modulo 998244353. Two subtrees are different if and only if they have different vertex set or edge set.

Input

Each test contains multiple test cases. The first line contains one integer t ($1 \leq t \leq 5 \times 10^4$), indicating the number of test cases. The description of the test cases follows.

The first line contains one integer n ($1 \leq n \leq 10^5$, $\sum n \leq 10^6$), indicating the number of vertices in the tree T.

Each of the following $n-1$ lines contains two integers u, v ($1 \leq u \neq v \leq n$), indicating an edge on the tree T.

Output

For each test case, print one integer, indicating the sum of $f(T')$, modulo 998244353.

Examples

standard input	standard output
2 5 1 2 1 3 2 4 2 5 8 1 2 8 7 1 3 8 6 1 4 8 5 1 8	12 94

Note

For the first test case, there are 17 different subtrees. 5 of the subtrees contain only one vertex and thus $f(T') = 0$. All the other subtrees satisfy $f(T') = 1$. Thus, the answer is 12.

Problem J. Hiking Trip

Input file:	`standard input`
Output file:	`standard output`
Time limit:	4 seconds
Memory limit:	2048 megabytes

Una and Kamome plans to go hiking in the suburb of Guangzhou.

There are n views from west to east in the mountain, numbered from 1 to n. The i-th view has an altitude h_i. Una decides to choose an interval $[l, r]$ $(1 \le l \le r \le n)$ and travel from the l-th view to the r-th view.

However, Una dislike valleys, so she doesn't want any $l < i < r$ to be a valley, that is, $h_{i-1} > h_i < h_{i+1}$. At the same time, she thinks that flat roads are boring, so she hopes that for all $l \le i < r$, $h_i \ne h_{i+1}$. Una will enjoy the trip if the interval $[l, r]$ satisfies these two conditions.

Kamome made some research before the hiking trip and find out the altitudes of some of the views. She wants to know that if the altitudes of all the other views are independent random integers in $[1, m]$, how many different intervals $[l, r]$ can be chosen to help Una enjoy the trip. Help Kamome find the expected value, modulo $10^9 + 7$.

Input

Each test contains multiple test cases. The first line contains one integer t $(1 \le t \le 10^5)$, indicating the number of test cases. The description of the test cases follows.

The first line contains two integers n, m $(1 \le n \le 10^6, \sum n \le 10^7, 1 \le m \le 10^9)$, indicating number of views in the mountain and the altitude range.

The second line contains n integers $h_1, h_2, \ldots, h_n$ $(1 \le h_i \le m$ or $h_i = -1)$, indicating the result of the research Kamome made. If $h_i \ne -1$, h_i means the actual altitude of view i. Otherwise, it means that Kamome didn't find any information about the altitude of view i and regard it as a random integer in $[1, m]$.

Output

For each test case, print one integer, indicating the expected number of intervals $[l, r]$ such that Una is enjoyed, modulo $10^9 + 7$.

Examples

standard input	standard output
10 4 4 1 4 2 3 3 3 -1 2 -1 4 2 -1 -1 -1 1 4 3 1 -1 -1 3 5 5 -1 2 -1 4 -1 6 4 2 -1 1 -1 -1 1 5 5 1 -1 4 -1 1 8 4 -1 2 -1 -1 2 -1 4 -1 9 7 4 -1 2 -1 -1 6 -1 4 -1 20 20 -1 -1 -1 5 -1 -1 13 -1 10 -1 -1 -1 -1 12 -1 3 -1 -1 -1 18	8 666666676 875000012 555555566 872000016 750000017 400000014 554687520 973046972 216066617

Note

For the first test case, Una enjoys the trip if she chooses intervals $[1, 1]$, $[2, 2]$, $[3, 3]$, $[4, 4]$, $[1, 2]$, $[2, 3]$, $[3, 4]$ or $[1, 3]$.

For the second test case, the answer is $\frac{14}{3}$.

Problem K. 0721 Master

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	1024 megabytes

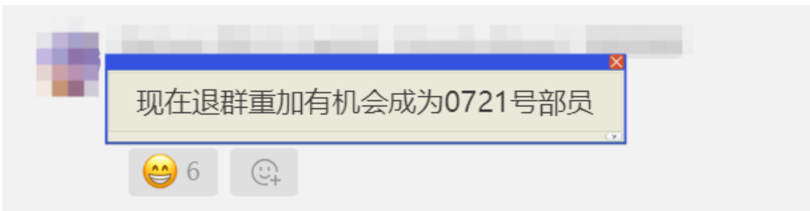
In the ancient times of a distant eastern country, there existed a mysterious QQ group. The people there had a unique custom: they possessed a special "Divine Number" k, and they would worship the group member who had the k-th earliest join time among the current members in the chat. Of course, if the current number of members in the group was strictly less than k, no one would be worshipped.

Now, let us introduce the history of this QQ group. Legend has it that the group was initially empty. Subsequently, n people joined it one after another. Ancient texts recorded the names and the joining order of these n individuals. However, for the sake of the mental health of the contestants participating in this competition, we will simply number these n people from 1 to n **in chronological order** of their joining.

As a QQ group with a long history, its roster of members was by no means static. A person who had joined the group could leave at any arbitrary moment. This could happen before or after any other person joined, or it might never happen at all. However, due to a mysterious rule of the chat group, once a person left, they could never join the group again.

Finally, the ancient texts provide a 01-string S of length n. The i-th character of S is '1' if and only if the member numbered i was worshipped at least at one moment throughout the history of the group chat.

Given n, k, and the string S, can you help us reconstruct a possible history of this group? However, the ancient texts might be flawed. Therefore, if there is no way to reconstruct any valid history, you simply need to tell us that it is impossible.



Picture 9: The ancient text.

Input

Each test contains multiple test cases. The first line contains one integer t ($1 \le t \le 5 \times 10^3$), indicating the number of test cases. The description of the test cases follows.

The first line contains two integers n, k ($1 \le k \le n$, $1 \le \sum n \le 5000$), indicating the number of people and the "Divine Number".

The second line contains the 01-string S, indicating whether each person was worshipped at least at one moment throughout the history of the group chat.

Output

For each test case, if it is impossible to construct a history that satisfies the conditions, output `-1`.

Otherwise, output a positive integer c on the first line, representing the total number of join or leave events. You must ensure that $n \le c \le 2n$.

The following c lines describe each event in chronological order. Each line must be one of the following two formats:

- `I x`: denotes that the member numbered x joins the group. You must ensure that member x has never joined the group before this event occurs. Furthermore, after all events are processed, all members numbered $1 \sim n$ must have joined the group at some point.
- `O x`: denotes that the member numbered x leaves the group. You must ensure that member x is currently in the group when this event occurs.

Examples

standard input	standard output
2 4 2 0111 5 3 00110	6 I 1 I 2 I 3 O 1 I 4 O 2 6 I 1 I 2 I 3 I 4 O 2 I 5

Note

For the first test case, the reconstructed history of the group chat proceeds as follows:

- Initially, the group is empty.
- `I 1`: Member 1 joins. The current members are $\{1\}$. Since there is strictly less than $k = 2$ members, no one is worshipped.
- `I 2`: Member 2 joins. The current members are $\{1, \mathbf{2}\}$.
- `I 3`: Member 3 joins. The current members are $\{1, \mathbf{2}, 3\}$.
- `O 1`: Member 1 leaves. The current members are $\{2, \mathbf{3}\}$.
- `I 4`: Member 4 joins. The current members are $\{2, \mathbf{3}, 4\}$.
- `O 2`: Member 2 leaves. The current members are $\{3, \mathbf{4}\}$.

Throughout the history of the group chat, members 2, 3, and 4 were worshipped at least once, while member 1 was never worshipped. So it matches $S =$ `0111`.

Problem L. Senren Banka

Input file:	`standard input`
Output file:	`standard output`
Time limit:	12 seconds
Memory limit:	1024 megabytes

Kamome is making efforts to train the AI to play games. She is having the AI train the game "Senren Banka".

This game consists of n choice branches in each round. You are considered to pass the game if and only if you select each branch correctly in one round. Each round starts with branch 1, and for each $i = 1, 2, \ldots, n-1$, You can select the $i+1$-th branch only after selecting the i-th branch correctly. If you make a wrong choice, you will enter the Bad Ending and have to start over from the beginning. This is regarded as a new round of the game.

Kamome's AI can learn from past choices. More specifically, if this is the j-th time choosing the i-th branch, then the AI has a probability of $p_{i,\min(m,j)}$ of choosing the correct option.

Kamome wants to know the number of expected rounds it would take for this AI to pass the game for the first time. Output the result modulo 998244353.

Input

The first line contains two integers n, m ($1 \le n \le 20$, $1 \le m \le 5 \times 10^4$), indicating the number of choice branches and the threshold of this AI.

The next n lines, each line contains m integer $p'_{i,j}$ ($1 \le p'_{i,j} \le 100$), indicating $p_{i,j} = \frac{p'_{i,j}}{100}$.

Output

Output one line contains one integer, indicating the answer modulo 998244353.

Examples

standard input	standard output
1 1 50	2
2 2 25 50 50 25	499122183
10 10 1 2 3 4 5 6 8 7 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100	473598335

Problem M. Put Happiness in your Pocket

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	1024 megabytes

This is an interactive problem.

Sea, you again. Wish you happiness next summer holiday!

Long, long ago, Una and Kamome were already good friends.

Picture 10: Una and Kamome travelling in Qingdao

They liked playing guessing games. Last time, the game Una invented was quickly cracked by Kamome's excellent strategy. This time, Una brought a new game.

Una has a tree in mind with n nodes. Kamome can make no more than n queries:

- Kamome can tell Una a permutation $p_1, p_2, \ldots, p_n$ of $0 \sim n-1$, Una will tell Kamome the length of the weighted diameter of the tree if node i is assigned a weight of 2^{p_i}.

But sometimes, Kamome finds that Una is being a troublemaker. More specifically, sometimes no matter how many times she asks, she still can't determine the tree with certainty. In such cases, Kamome can report that there is no solution.

Interaction Protocol

This is an interactive problem. Remember to flush the output buffer after every print. To flush your output, you can use:

- `fflush(stdout)` or `cout.flush()` in C/C++;
- `System.out.flush()` in Java and Kotlin;
- `sys.stdout.flush()` in Python.

First, you should read an integer T ($1 \le T \le 10^4$) indicating the number of test cases.

For each test cases, you should read an integer n ($2 \le n \le 100$, $\sum n^2 \le 10^6$), indicating the size of the tree.

To make a query, you should output one line formatted as "`?` p_1 p_2 ... p_n" ($0 \le p_i \le n-1$, $p_i \ne p_j$ for $1 \le i < j \le n$).

Then, read a string that contains n characters indicating the weight of diameter of the tree in binary representation. If your query is invalid, or you make more than n queries in one test case, jury program will output -1. If your program reading -1, you should exit immediately to avoid undefined behavior.

If you determine that it is impossible to uniquely determine the tree, output "`! NO`".

Otherwise, you should output one line formatted as "`! YES` u_1 v_1 u_2 v_2 ... u_{n-1} v_{n-1}" indicating this tree contains edges (u_i, v_i). You can output them in any order.

Then, read one word `OK` or `WA` indicating whether your answer is correct. If your program reading `WA`, you should exit immediately to avoid undefined behavior.

If the tree is impossible to determine it and you output `YES`, even though you output the correct answer, your answer will also be considered `Wrong Answer`, because Una thinks you are cheating.

Note that the grader is **not adaptive**, that is the answer won't change after your queries.

Examples

standard input	standard output
2	
4	? 0 1 2 3
1110	? 2 1 0 3
1110	! YES 1 4 2 4 3 4
OK	
3	? 0 1 2
111	! NO
OK	

Note

A testing tool is provided to help contestants develop and test their solutions. You can download this tool from the attachments. Executing the tool with a "-h" option should describe how to use the tool. The testing tool will only implement some test scenarios and only some functionality of the real judge program.