The 2nd Universal Cup

Semifinals

Online Contest Session

June 23, 2024

This problem set should contain 13 problems on 20 numbered pages.

Problem A. Records in Chichén Itzá

Welcome to the scenic beaches and rich cultural heritage of Cancun, the site of The 2nd Universal Cup Summer Summit sponsored by Huawei! As you soak in the beauty of this coastal paradise, you are also transported back in time to the ancient city of Chichén Itzá, a renowned Mayan civilization site known for its magnificent pyramids and advanced understanding of astronomy and mathematics.

At Chichén Itzá, you discover some records of the Mayan civilization's study of graph theory. Among the ruins, you find some *degree sequences* of some trees recorded by the ancient Maya. Specifically, a degree sequence of a tree is the sorted list of the degrees of all the vertices of the tree.

Given a degree sequence, you need to determine if there exist two non-isomorphic trees that correspond to the given degree sequence.

Remind that we say $G_1(V_1, E_1)$ and $G_2(V_2, E_2)$ are isomorphic if and only if there exists a bijection between the vertex sets $\varphi : V_1 \mapsto V_2$ such that:

$$\forall x, y \in V_1,\ (x, y) \in E_1 \iff (\varphi(x), \varphi(y)) \in E_2$$

Input

There are multiple test cases. The first line of the input contains an integer T ($1 \le T \le 10^5$), representing the number of test cases. For each test case:

The first line contains an integer n ($2 \le n \le 10^5$), representing the length of the degree sequence of a tree.

The second line contains n integers $d_1, d_2, \ldots, d_n$, representing the degree sequence. It is guaranteed that there exists a tree that corresponds to the given degree sequence.

It is guaranteed that the sum of n over all test cases does not exceed 2×10^5.

Output

For each test case, print "`Yes`" (without quotes) if there exist two different trees that correspond to the given degree sequence, otherwise print "`No`" (without quotes).

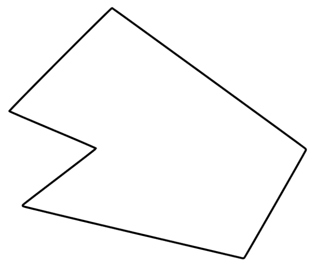
Example

standard input	standard output
3 6 1 1 1 1 3 3 5 1 1 2 2 2 10 1 1 1 1 2 2 2 2 3 3	No No Yes

Problem B. Almost Convex 2

This is a story about Kevin, a friend of Little Cyan Fish, again.

Kevin is the chief judge of the International Convex Polygon Championship (ICPC). He proposed a geometry task for the contest. However, since he is inexperienced in computational geometry, he couldn't generate a correct convex polygon for the tests of the task.



Kevin was very saddened by this. His good friend, Little Cyan Fish, consoled him by saying, "Although the data you generated is not a convex polygon, you can call it an almost-convex polygon!"

Given a set of points S (containing at least 3 points) in a two-dimensional plane, where no two points coincide and no three points are collinear, Little Cyan Fish calls a polygon P an almost-convex polygon if and only if:

- The polygon P is simple, i.e., its vertices are distinct and no two edges of the polygon intersect or touch, except that consecutive edges touch at their common vertex.
- The vertices of the polygon belong to S, and all points in S are either inside or on the boundary of the polygon.

Let $\mathbb{U}$ be the set consisting of all almost-convex polygons. It can be shown that $\mathbb{U}$ is a finite set and is not empty. Therefore, there exists a polygon R such that $|R|$ is the minimum among all polygons in $\mathbb{U}$ ($|R|$ is the number of vertices of polygon R).

Kevin and Little Cyan Fish want you to calculate the number of polygons $Q \in \mathbb{U}$ such that $|Q| \leq |R| + 2$.

Input

The first line contains an integer n ($3 \leq n \leq 500$), representing the number of points in S.

For the following n lines, the i-th line contains two integers x_i and y_i ($-10^4 \leq x_i, y_i \leq 10^4$), representing a point (x_i, y_i) in S.

It is guaranteed that no two points in S coincide and no three points are collinear.

Output

Output one line containing one integer representing the number of polygons Q.

Examples

standard input	standard output
7 0 2 1 0 1 3 2 5 3 0 3 3 4 2	26
5 4 0 0 0 2 1 3 3 3 1	13
3 0 0 3 0 0 3	1

Problem C. Space Station

In the distant future, Little Cyan Fish is the commander of a high-tech space station on the frontier of human civilization. His station is under constant threat from an alien enemy that attacks at unpredictable intervals. These attacks vary in intensity and can cause severe damage to his station's infrastructure.

As the commander, he has access to advanced defensive systems. Each time an enemy attack is imminent, he can choose to activate the defensive shields, nullifying the attack, or let the attack through, dealing with the damage afterward. However, activating the shields comes at a cost.

Specifically, Little Cyan Fish will be attacked n times. Before each attack, he can choose to spend m dollars to activate the shield, making the attack ineffective (i.e., dealing no damage). After deciding whether to activate the shield for a particular attack, he will learn the current attack's intensity b_i. If Little Cyan Fish did not activate the shield for this attack, he must pay b_i dollars to repair the damage caused by the attack. (Note that activating the shield only makes the current attack ineffective. If he chooses to activate the shield for the next attack, he has to pay the cost again.)

However, the damage b_i caused by each attack is not yet known. Little Cyan Fish only knows that b_i is a random permutation of another known sequence $a_1, a_2, \cdots, a_n$. In other words, $b_i = a_{\sigma_i}$, where $\sigma_1, \sigma_2, \cdots, \sigma_n$ is a permutation of $\{1, 2, \ldots, n\}$ chosen uniformly at random from all $n!$ possible permutations. Note that after each attack, Little Cyan Fish will know the damage of the attack, whether he activated the shield or not.

Given the sequence $a_1, a_2, \cdots, a_n$, determine the expected minimum cost of handling the attacks using the optimal strategy, modulo 998 244 353. More formally, represent the expected minimum cost as an irreducible fraction $E = p/q$. Then, there exists a unique integer $r \in [0, 998\,244\,353)$ such that $r \times q \equiv p \pmod{998\,244\,353}$, so output this integer r.

Input

The first line of the input contains two integers n and m ($1 \le n \le 100$, $1 \le m \le 100$).

The next line of the input contains n integers $a_1, a_2, \cdots, a_n$ ($1 \le a_i \le 200$).

Output

Output a single line containing a single integer, representing the answer.

Examples

standard input	standard output
3 3 2 3 4	499122185
5 1 10 20 30 40 50	5

Note

In the first example, one of the optimal strategies is:

- Activate the shield before the first attack. This will cost 3 dollars.
  - There is a 1/3 of chance that $a_1 = b_1 = 2$. In this case, Little Cyan Fish should activate the shield for all remaining rounds of attacks. This will cost 3×2 more dollars. So the cost in this case will be $3 + 3 \times 2 = 9$

- There is a 1/3 of chance that $a_1 = b_2 = 3$. In this case, Little Cyan Fish should NOT activate the shield in the second round of attack.
  * There is a 1/2 of chance that $a_2 = b_1 = 2$. In this case, Little Cyan Fish needs to pay 2 dollars to repair the damage. Then, Little Cyan Fish should activate the shield in the third round of attack, which will cost 3 dollars. So the cost in this case will be $3 + 2 + 3 = 8$.
  * There is a 1/2 of chance that $a_2 = b_3 = 4$. Little Cyan Fish needs to pay 4 dollars to repair the damage. Then, Little Cyan Fish should NOT activate the shield in the third round of attack, which will cost $a_3 = b_1 = 2$ dollars to repair the damage. So the cost in this case will be $3 + 4 + 2 = 9$.
- There is a 1/3 of chance that $a_1 = b_3 = 4$. In this case, Little Cyan Fish should NOT activate the shield for all remaining rounds of attacks. This will cost $2 + 3$ more dollars to fix the damage in the remaining two rounds of attacks. So the cost in this case will be $3 + 2 + 3 = 8$.

So, the answer in the first test case will be $9\times\frac{1}{3}+8\times\frac{1}{6}+9\times\frac{1}{6}+8\times\frac{1}{3} = \frac{17}{2} \equiv 499\,122\,185 \pmod{998\,244\,353}$.

In the second example, the optimal strategy is to activate the shield for all rounds.

Problem D. Solar Panel Grid Optimization

In the not-so-distant future, the world is managed by advanced robots capable of performing complex tasks to maintain global systems. One such system is responsible for managing a massive grid of solar panels that generate renewable energy for an entire city. These panels are arranged in an $N \times N$ matrix, where each cell in the matrix can either be in an active (1) or inactive (0) state.

Figure 1: By AleSpa - Own work, CC BY-SA 3.0

To optimize energy production, the robots need to transform the current state matrix A of the solar panel grid into a desired state matrix B. The transformation process involves two types of operations:

- Shift an entire row to the left, i.e., select a positive integer i between 1 and N and transform $(A_{i,1}, A_{i,2}, \ldots, A_{i,N})$ into $(A_{i,2}, A_{i,3}, \ldots, A_{i,N}, A_{i,1})$.
- Shift an entire column downwards, i.e., select a positive integer i between 1 and N and transform $(A_{1,i}, A_{2,i}, \ldots, A_{N,i})$ into $(A_{N,i}, A_{1,i}, A_{2,i}, \ldots, A_{N-1,i})$, and then flip the state of the top cell, making the new $A_{1,i} \leftarrow 1 - A_{1,i}$.

You are tasked with finding a sequence of operations that can transform the grid from its current state A to the desired state B by using no more than 10^3 operations.

Input

The first line of the input contains a single integer N ($3 \leq N \leq 20$).

The next N lines describe the initial matrix A. The j-th character on the i-th line of these lines will be either 0 or 1, representing the value of $A_{i,j}$.

After these N lines, there will be an empty line.

The next N lines describe the target matrix B. The j-th character on the i-th line of these lines will be either 0 or 1, representing the value of $B_{i,j}$.

Output

The first line of the output should be an integer T ($0 \leq T \leq 10^3$), representing the number of the operations you used.

The next T lines of the output describe the operations. The i-th line of these lines should be one of the following:

- `row` i: Perform an operation on the i-th row.
- `column` i: Perform an operation on the i-th column.

If there are multiple possible solutions, you may print any of them.

Example

standard input	standard output
4 1011 1100 0100 1011 1001 0110 0110 1001	4 row 2 row 2 row 2 column 3

Problem E. Fast Median Transform

The *median* in an array of length k is the element that occupies the position $\lfloor \frac{k+1}{2} \rfloor$ after we sort the elements in non-decreasing order. For example, the median of $[5, 1, 2, 3]$ is 2, and the median of $[2, 1, 1]$ is 1.

Little Cyan Fish gives you an array a of length n, and an array b of length m. Here, the index of an array starts from 0.

For an integer X_0, we define the "Fast Median Transform" of (a, b, X_0) (called $\mathrm{FMT}(a, b, X_0)$):

- Let X be a variable, initially $X = X_0$.
- For each integer i from 0 to $nm - 1$, the following operations happen in order:
  - Calculate the median of the array $[a_{i \bmod n}, b_{i \bmod m}, X]$. Let the median be Y.
  - Assign Y to X.
- We define $\mathrm{FMT}(a, b, X_0)$ as **the value of X at the end of the process**.

You will be given q queries. Each query consists of three integers x, y', X'_0. For each query, perform the following operations:

1. Update a_x to $y' \oplus lastans$, where $\oplus$ denotes the bitwise XOR operation and $lastans$ is the result of the previous query (initially zero).
2. Compute and print $\mathrm{FMT}(a, b, X'_0 \oplus lastans)$.

Note: The queries are not independent; they occur in order, and the effect of each query persists.

Input

The first line of the input contains three integers n, m, and q ($1 \le n, m, q \le 3 \times 10^5$), indicating the length of a, the length of b, and the number of queries, respectively.

The second line contains n non-negative integers $a_0, a_1, \cdots, a_{n-1}$ ($0 \le a_i < 2^{29}$).

The third line contains m non-negative integers $b_0, b_1, \cdots, b_{m-1}$ ($0 \le b_i < 2^{29}$).

Then q lines follow.

The i-th line contains three non-negative integers x, y', and X'_0 ($0 \le x < n, 0 \le y', X'_0 < 2^{29}$), indicating the i-th query.

Output

Output q lines. The i-th line contains a single integer indicating the answer $\mathrm{FMT}(a, b, X'_0 \oplus lastans)$ of the i-th query.

Example

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

Problem F. Colorful Graph 3

Having delved into the complex theory of quantum chromodynamics once again, Little Cyan Fish has become fascinated with the concept of color charge. To test your understanding of this theory, he has proposed the following task to you.

Given an integer n, and k non-negative integers $c_1, c_2, \cdots, c_k$, construct an undirected graph with n vertices satisfying the following conditions:

- Each edge has a color w_i which is an integer in $[1, k]$.
- For all $1 \le i, j \le n$ and all $1 \le t \le k$, there exists a path connecting the vertices i and j such that the number of edges with color t on the path does not exceed c_t. (Note that the paths for different t's need not be the same.)
- The number of edges in the constructed graph is minimized.

It is guaranteed that there is at least one possible solution.

Input

The first line of the input contains a single integer T ($1 \le T \le 5 \times 10^4$), indicating the number of test cases.

For each test case, the first line of the input contains two integers n, k ($2 \le n, k \le 10^5$).

The next line of the input contains k integers c_i ($0 \le c_i \le n$).

It is guaranteed that there is at least one possible solution, and the sum of n and the sum of k over all test cases do not exceed 10^5 respectively.

Output

For each test case, output an integer m on its own line, indicating the number of edges in your constructed graph.

The next m lines of the output describe your graph. The i-th line of these lines contains three integers u_i, v_i, and w_i ($1 \le u_i, v_i \le n$, $1 \le w_i \le k$), indicating an edge connecting the vertex u_i and v_i with color w_i.

Example

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

Problem G. CNOI Knowledge

This is an interactive problem.

CNOI, or Chinese Olympiad in Informatics, is generally used to refer to a series of Olympiad in Informatics competitions in China. Competitors in China excel at solving intricate data structure problems and complex combinatorial counting problems. Furthermore, students are diverse in their interests and personalities, and the community embraces this diversity.

Mr. Wuwuwu, an anonymous teacher, is a grandmaster in CNOI techniques. He really enjoys different kinds of data structure problems and often wonders if he can get something new out of them. Today, he is teaching one of his students Little Cyan Fish how to solve the following problem.

Figure 2: The yellow hibiscus, also known as **pua alo alo**, is Hawaii's state flower.

Original Problem
You are given a string S of length n consisting of positive integers in $[1, 10^9]$. You need to handle q queries. In each query, you are given two integers l and r. You need to count the number of distinct substrings in the string $S[l \cdots r]$.

This problem first appeared in China about 10 years ago and has since become well-known worldwide. However, Mr. Wuwuwu thought this problem was too easy for the year 2024, so he presented a slightly different version:

New Problem
There is a string S of length n consisting of positive integers in $[1, 10^9]$. You can do the following queries for at most 10^4 times:

- *? l r*: Ask for the number of distinct substrings in the string $S[l \cdots r]$ $(1 \le l \le r \le n)$.

Your task is to recover the original string S by using these queries. If there are multiple possible strings , you may print any of them. In particular, a string T will be considered as a correct answer if and only if the number of distinct substrings in the string $S[l \cdots r]$ equals to the number of distinct substrings in the string $T[l \cdots r]$ for all $1 \le l \le r \le n$.

Can you show Mr. Wuwuwu that you can solve this new problem?

Interaction Protocol

The first line of the input contains a single integer n $(2 \le n \le 10^3)$.

Then, the interaction begins. You may perform at most 10^4 queries in the interaction process. To perform a query, you need to print a single line "`?` l r" $(1 \le l \le r \le n)$, indicating a query. Then, you need to read the result of your query from the standard input.

To give your answer, you need to print "`!` s_1 s_2 $\cdots$ s_n" $(1 \le s_i \le 10^9)$. If there are multiple possible answers, you may print any of them. Printing the answer is not considered a query and does not count toward the limit (10^4) on the number of queries. After printing your answer, you need to terminate your program immediately.

After printing a query, do **NOT** forget to output an end of line and flush the output. To do this, use "`fflush(stdout)`" or "`cout.flush()`" in C++, "`System.out.flush()`" in Java, "`flush(output)`" in Pascal, or "`stdout.flush()`" in Python.

In this problem, it is guaranteed that the interactor is **non-adaptive**. That is, the string s is decided before the interaction process. They will not be changed based on your queries.

Example

standard input	standard output
12	
	? 1 1
1	
	? 1 2
3	
	? 1 12
72	
	? 6 12
25	
	? 5 10
19	
	! 6 12 15 23 5 18 12 5 20 20 5 18

Problem H. Exchanging Kubic 2

Little Cyan Fish is conducting a social experiment with Prof. Kubic. In this experiment, there are n people initially located at positions $a_1, a_2, \cdots, a_n$. You need to perform exactly n operations, and each person can be chosen only once. In each of n operations, you must choose one person, and all other people will move one step towards this chosen person.

Specifically, if you choose the i-th person, then for each j-th person ($j \neq i$):

- If $a_i > a_j$, then $a_j \leftarrow a_j + 1$.
- If $a_i < a_j$, then $a_j \leftarrow a_j - 1$.
- If $a_i = a_j$, then a_j remains unchanged.

We define the value of an array $a = (a_1, a_2, \cdots, a_n)$ as the minimum possible distance between the two farthest people after all operations.

Now, you are given n and n sets $S_1, S_2, \ldots, S_n$. An array $a = (a_1, a_2, \ldots, a_n)$ is valid if and only if:

1. $a_i \leq a_{i+1}$ for all $1 \leq i < n$
2. $a_i \in S_i$ for all $1 \leq i \leq n$

You need to find the sum of values of all valid arrays a. The answer should be taken modulo 998 244 353.

Input

The first line of the input contains one integer n ($1 \leq n \leq 400$).

The next n lines of the input describe each set S_i. The first integer of the i-th line of these lines is an integer $|S_i|$ ($0 \leq |S_i| \leq 800 + 1$). Then follow $|S_i|$ distinct integers in the range $[0, 800]$, representing the set.

Output

Output a single line containing a single integer, representing the answer.

Examples

standard input	standard output
5 3 1 2 3 1 2 0 4 0 2 3 4 2 2 3	0
5 4 1 2 3 7 4 5 7 8 9 4 2 3 6 9 5 0 1 4 7 9 8 0 1 2 3 6 7 8 9	16

Problem I. Nightmare

Is this a programming contest?

Given a prime p and an $n \times n$ **non-zero** symmetric matrix G in which each element is an integer in $[0, p)$.

Little Cyan Fish asks you to find the **smallest** integer m satisfying:

- There exists n arrays of length m, $v_1, \ldots, v_n$, in which each element is an integer in $[0, p)$, such that for all $1 \leq i, j \leq n$, we have:

$$G_{i,j} = \left(\sum_{k=1}^{m} v_{i,k} v_{j,k} \right) \bmod p.$$

Input

The first line contains two integers n and p ($1 \leq n \leq 500$, $2 \leq p \leq 10^6$).

Each of the next n lines contains n integers. The j-th integer in the i-th line is $G_{i,j}$ ($0 \leq G_{i,j} < p$, $G_{i,j} = G_{j,i}$ for all $1 \leq i, j \leq n$, $G \neq \mathbf{0}$).

It is guaranteed that p is a prime number, and that there exists at least one possible solution.

Output

The first line of the output contains an integer m, indicating the smallest possible integer m.

Each of the next n lines contains m integers. The j-th integer on the i-th line is $v_{i,j}$.

If there are multiple possible solutions, you may print any of them.

Examples

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

Problem J. Guess The Sequence 2

Little K has a **random** permutation $p_1, p_2, \ldots, p_n$ of $1 \sim n$. And he is going to give his friend Little Cyan Fish a quiz.

Little K can choose a set of subsegments of the permutation and provide the maximum value of each subsegment to Little Cyan Fish. Formally, he can choose a set $\{(l_1, r_1), (l_2, r_2), \ldots, (l_k, r_k)\}$ where $1 \le l_i \le r_i \le n$ for every $1 \le i \le k$, and Little Cyan Fish will receive k tuples $(l_1, r_1, m_1), (l_2, r_2, m_2), \cdots, (l_k, r_k, m_k)$ where $m_i = \max_{j=l_i}^{r_i} p_j$, indicating the maximum value of each subsegment Little K chooses.

Little Cyan Fish has to guess what Little K's permutation is. Since Little K is Little Cyan Fish's best friend, he needs to make it possible to correctly guess the permutation based on the information he provided. Therefore, there should be only one permutation satisfying the information Little K provides.

Little K is curious about the number of different sets of subsegments that can be provided to Little Cyan Fish, so that he can uniquely determine the permutation. Unfortunately, Little K cannot figure this problem out himself, so he asks you for help. Since the answer is really big, print it modulo 998 244 353.

Input

The first line of the input contains a single integer n ($1 \le n \le 5 \times 10^5$).

The next line of the input contains n integers $p_1, p_2, \cdots, p_n$ ($1 \le p_i \le n$, $p_1, p_2, \cdots, p_n$ is a permutation of $1 \sim n$). **It is guaranteed that the permutation is uniformly randomly chosen from all permutations of length n.**

Output

Output a single line contains a single integer, indicating the answer modulo 998 244 353.

Examples

standard input	standard output
2 1 2	6
4 1 4 2 3	532

Problem K. Game: Battle of Menjis

Huangmenji is a popular Chinese dish known for its rich flavor and aromatic qualities. Originating from Shandong province, it has gained widespread popularity throughout China and beyond. The dish primarily consists of chicken braised in a savory and slightly sweet soy sauce-based sauce, accompanied by various vegetables such as mushrooms, bell peppers, and bamboo shoots.

Figure 3: A huangmenji with rice. CC-BY-SA-4.0

Little Cyan Fish recently found a game called *Battle of Menjis*. Now, Alice and Bob are playing this game.

The game starts with a sequence of n non-negative integers, $a_1, \cdots, a_n$. Alice and Bob take turns, Alice goes first.

In Alice's turn, she can choose an index i ($1 \le i \le n$), and replace a_i by $a_i + 1$.

In Bob's turn, he can choose an index i ($1 \le i \le n$) satisfying $a_i > 0$ (we can prove that such i must exist), and replace a_i by $a_i - 1$.

The game ends when both Alice and Bob have taken k turns.

Alice wants to maximize $\bigoplus_{i=1}^{n} a_i$, and Bob wants to minimize $\bigoplus_{i=1}^{n} a_i$. Here $\bigoplus_{i=1}^{n} a_i$ represents the bitwise exclusive or of the sequence a.

It can be proved that an optimal strategy for each of the two players exists. Please determine $\bigoplus_{i=1}^{n} a_i$ when the two players play optimally.

Input

The first line contains a single integer T ($1 \le T \le 10^5$), representing the number of testcases.

For each testcase, the first line contains two integers n, k ($1 \le n \le 10^5, 1 \le k \le 10^9$), and the second line contains n integers $a_1 \dots a_n$ ($0 \le a_i \le 10^9$).

It is guaranteed that the sum of n in all testcases does not exceed 5×10^5.

Output

For each testcase, output one integer, $\bigoplus_{i=1}^{n} a_i$ when the two players play optimally, in a single line.

Example

standard input	standard output
4 2 3 1 1 4 4 0 0 0 0 4 1 1 2 4 8 13 5 1 1 4 5 1 4 1 9 1 9 8 1 0	0 0 9 11

Problem L. Slay the Spire

Little Cyan Fish is playing the game "Slay the Spire". At any given moment, his character is in one of the m possible states. Initially, his character starts in the state s.

In the game, he has access to n cards. The effect of the i-th card is as follows:

- If the current state of your character is a_i, then deal b_i damage to the enemy and change the state to c_i.
- Otherwise, only the state will be changed to c_i.

Additionally, he has k potions. The effect of the i-th potion is to change the current state to x_i.

Now, the game starts. At any moment, he can choose to use any card or any potion. He may use these items in any order, but each item (card or potion) could be used at most once. His goal is to determine the maximum possible damage you can deal to the enemy.

Input

The first line of the input contains a single integer T ($1 \le T \le 5\,000$), indicating the number of test cases.

In each test case, the first line contains four integers n, m, k and s ($1 \le n \le 10^3$, $1 \le m, k \le 500$, $1 \le s \le m$), representing the number of cards, the number of states, the number of potions, and the initial state, respectively.

The next n lines of the input describe all the cards. The i-th line of these lines contains three integers a_i, b_i and c_i ($1 \le a_i, c_i \le m$, $1 \le b_i \le 10^9$), representing the effect of the i-th card.

The next line of the input contains k integers $x_1, x_2, \cdots, x_k$ ($1 \le x_i \le m$), describing the effects of all the potions.

It is guaranteed that over all test cases, none of the sum of n, the sum of m, and the sum of k exceeds $5\,000$.

Output

For each test case, output a line with a single integer, representing the answer.

Example

standard input	standard output
1 6 5 2 1 1 100 2 1 100 4 1 100 5 1 200 5 2 100 3 3 100 1 1 5	600

Problem M. Puzzle: Summon

Little Cyan Fish is a fan of logic puzzles. Today, he is playing a special version of the classic puzzle "Summon".

Consider a grid with 2 rows and $2n$ columns. Let (x, y) denote the cell on the x-th row and y-th column. The grid was divided into n regions, where each region is a 2×2 square. Specifically, for each $1 \le i \le n$, the i-th region R_i contains exactly 4 cells: $(1, 2i-1)$, $(1, 2i)$, $(2, 2i-1)$, and $(2, 2i)$.

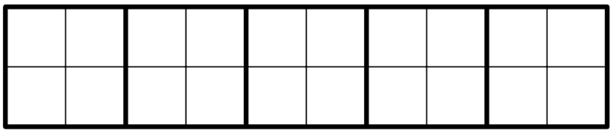


Figure 4: A sample grid for $n = 5$. The borders of each region was marked in bold lines.

The task of the puzzle is to fill digits from 1 to 2 in some of the cells according to the following rules:

- For each region R_i $(1 \le i \le n)$, it must contain each digit from 1 to 2 exactly once.
- Cells with the same digit don't touch, not even diagonally.

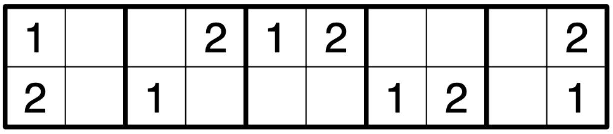


Figure 5: A valid plan to fill the integers

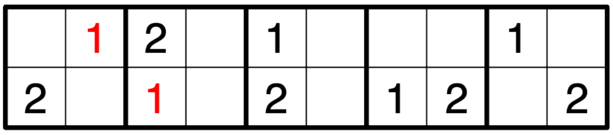


Figure 6: An invalid plan: the digits marked red touch each other

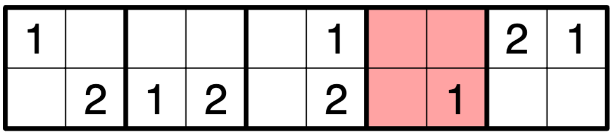


Figure 7: An invalid plan: the region R_4 doesn't contain the digit 2.

The value of a plan will be defined as the sum of the formed numbers in the first row. Specifically, the connected blocks of digits in the first row form numbers by reading left to right. And the sum of all such numbers will be the the value of the plan.

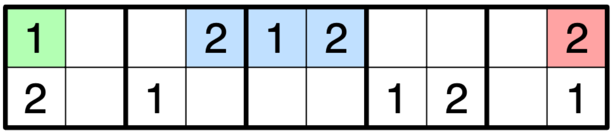


Figure 8: In this plan, the value will be $1 + 212 + 2 = 215$

Now, Little Cyan Fish gives you an unfinished puzzle: a grid of $2 \times 2n$ with some pre-filled digits. Your task is to fill some digits in some empty cells to get a valid solution of the puzzle with the maximum value.

Input

There are multiple test cases in a single test file.

The first line of the input contains a single integer T ($1 \le T \le 100\,000$), indicating the number of the test cases.

For each test case, the first line of the input contains one single integer n ($1 \le n \le 100\,000$).

Each of the next two lines contains $2n$ characters, representing the initial state of the puzzle. Each character is either a digit from 1 to 2 (indicating a pre-filled digit), or a character `?` (indicating an empty cell).

It is guaranteed that the sum of n in all testcases does not exceed $1\,000\,000$.

Output

For each test case, if there is no possible plan, output a single line containing the string "impossible" (without quotes).

Otherwise, the first line of the output contains a single integer, indicating the maximum value of the plan.

The next two lines of the output describes the plan you found. Each line should contain $2n$ characters, indicating your solution. Each character is either a digit from 1 to 2 (indicating a cell with a digit filled) or 0 (indicating an empty cell). If there are multiple possible solutions with the maximum value, you may output any of them.

Example

standard input	standard output
5 2 ?1?? ??1? 4 1??????? ???2???2 5 ?????????? ?????????? 6 1212???????? ????????1212 7 ?2?1?????1?2?? ?????1???????2	impossible 242 12101210 00020002 2121212121 2121212121 0000000000 12121212 121212120000 000000001212 21 12010202010201 00020101020102