Problem A. Array

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	512 mebibytes

Given n, find any array $a_1, a_2, \ldots, a_n$ of integers such that all of the following conditions hold:

- $1 \le a_i \le 10^9$ for every i from 1 to n.
- $a_1 < a_2 < \ldots < a_n$
- For every i from 2 to n, a_i isn't divisible by a_{i-1}

It can be shown that such an array always exists under the constraints of the problem.

Input

The first line contains the number of test cases t ($1 \le t \le 100$). Description of the test cases follows.

The only line of each test case contains a single integer n ($1 \le n \le 1000$).

It is guaranteed that the sum of n over all test cases does not exceed 10^4.

Output

For each test case print n integers $a_1, a_2, \ldots, a_n$ — the array you found. If there are multiple arrays satisfying all the conditions, print any of them.

Example

standard input	standard output
3 1 2 7	1 2 3 7 8 9 10 11 12 13

Note

In the first test case, array [1] satisfies all the conditions.

In the second test case, array $[2, 3]$ satisfies all the conditions, as $2 < 3$ and 3 is not divisible by 2.

In the third test case, array $[111, 1111, 11111, 111111, 1111111, 11111111, 111111111]$ satisfies all the conditions, as it's increasing and a_i isn't divisible by a_{i-1} for any i from 2 to 7.

Problem B. Build the Permutation

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	512 mebibytes

You are given three integers n, a, b. Determine if there exists a permutation $p_1, p_2, \ldots, p_n$ of integers from 1 to n, such that:

- There are exactly a integers i with $2 \le i \le n-1$ such that $p_{i-1} < p_i > p_{i+1}$ (in other words, there are exactly a local maximums).
- There are exactly b integers i with $2 \le i \le n-1$ such that $p_{i-1} > p_i < p_{i+1}$ (in other words, there are exactly b local minimums).

If such permutations exist, find any such permutation.

Input

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

The only line of each test case contains three integers n, a and b ($2 \le n \le 10^5$, $0 \le a, b \le n$).

The sum of n over all test cases doesn't exceed 10^5.

Output

For each test case, if there is no permutation with the requested properties, output -1.

Otherwise, print the permutation that you are found. If there are several such permutations, you may print any of them.

Example

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

Note

In the first test case, one example of such permutations is $[1, 3, 2, 4]$. In it $p_1 < p_2 > p_3$, and 2 is the only such index, and $p_2 > p_3 < p_4$, and 3 the only such index.

One can show that there is no such permutation for the third test case.

Problem C. Cheating Game Master

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	512 mebibytes

n players are playing a game.

There are two different maps in the game. For each player, we know his strength on each map. When two players fight on a specific map, the player with higher strength on that map always wins. No two players have the same strength on the same map.

You are the game master and want to organize a tournament. There will be a total of $n-1$ battles. While there is more than one player in the tournament, choose a map and two remaining players to fight on it. The player who loses will be eliminated from the tournament.

In the end, exactly one player will remain, and he is declared the winner of the tournament. For each player determine if he can win the tournament.

Input

The first line contains a single integer t ($1 \le t \le 100$) — the number of test cases. The description of test cases follows.

The first line of each test case contains a single integer n ($1 \le n \le 10^5$) — the number of players.

The second line of each test case contains n integers $a_1, a_2, \ldots, a_n$ ($1 \le a_i \le 10^9$, $a_i \ne a_j$ for $i \ne j$), where a_i is the strength of the i-th player on the first map.

The third line of each test case contains n integers $b_1, b_2, \ldots, b_n$ ($1 \le b_i \le 10^9$, $b_i \ne b_j$ for $i \ne j$), where b_i is the strength of the i-th player on the second map.

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

Output

For each test case print a string of length n. i-th character should be "1" if the i-th player can win the tournament, or "0" otherwise.

Example

standard input	standard output
3 4 1 2 3 4 1 2 3 4 4 11 12 20 21 44 22 11 30 1 1000000000 1000000000	NO NO NO YES YES YES YES YES YES

Note

In the first test case, the 4-th player will beat any other player on any game, so he will definitely win the tournament.

In the second test case, everyone can be a winner.

In the third test case, there is only one player. Clearly, he will win the tournament.

Problem D. Dominoes

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	512 mebibytes

You are given n dominoes. Each domino has a left and a right cell. Each cell can be colored either black or white. Some cells are already colored, while some aren't yet.

The coloring is said to be **valid** if and only if it is possible to rearrange the dominoes in some order such that for each $1 \le i \le n$ the color of the right cell of the i-th domino is different from the color of the left cell of the $((i \bmod n) + 1)$-st domino.

Note that you can't rotate the dominoes, so the left cell always remains the left cell, and the right cell always remains the right cell.

Count the number of valid ways to color the yet uncolored cells of dominoes. Two ways are considered different if there is a cell that is colored white in the one way, and black in the other.

As this number can be very big, output it modulo $998\,244\,353$.

Input

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

The next n lines describe dominoes. Each line contains two characters which represent the left and the right cell. Character `B` means that the corresponding cell is black, character `W` means that the corresponding cell is white, and `?` means that the cell is yet to be colored.

Output

Print a single integer — the answer to the problem.

Examples

standard input	standard output
1 ?W	1
2 ?? W?	2
4 BB ?? W? ??	10

Note

In the first test case, there is only one domino, and we need the color of its right cell to be different from the color of its right cell. There is only one way to achieve this.

In the second test case, there are only 2 such colorings:

`BB WW` and `WB WB`.

Problem E. Escapable Submatrices

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	512 mebibytes

You are given the matrix $m \times n$, each cell of the matrix contains one arrow — direction of the movement (right, left, up or down).

The submatrix is called *escapable*, if, starting from any of its cells and following the arrows, you will leave this submatrix.

Your task is to find any escapable submatrix of maximal area (number of cells).

Input

The first line of the input contains two integers r and c — the number of rows and columns in the matrix, respectively ($1 \le r, c$; $1 \le r \times c \le 150\,000$).

Each of the following r lines contains exactly c characters: '<' (ASCII 60) for the left arrow, '>' (ASCII 62) for the right arrow,

'^' (ASCII 94) for the up arrow and 'v' (ASCII 118) for the down arrow, respectively.

Output

Print the coordinates of the escapable submatrix of maximal area: four integers row_a, col_a, row_b and col_b ($1 \le row_a \le row_b \le r$; $1 \le col_a \le col_b \le c$).

First two integers are coordinates of the upper left cell in the submatrix, last two are coordinates of the lower right cell.

Examples

standard input	standard output
2 2 >v ^<	1 1 1 2
2 3 >>< <<<	1 1 2 2

Problem F. Funny Deque

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	512 mebibytes

You have an array v, which is initially empty, and N queries. Array elements are numbered starting from 0. Each query belongs to one of the following types:

1. Insert the element x at the end of v;
2. Insert the element x at the beginning of v;
3. Remove an element from the end of v;
4. Remove an element from the beginning of v;
5. Apply *XOR Permutation* with the parameter x to the array v.
6. Get the maximum number that is currently contained in v.

XOR Permutation with the parameter x is a permutation of the form $p_i = i \oplus x$, where $\oplus$ is XOR (Exclusive OR) operation. It's guaranteed that all permutations are correct. Also, when performing a deletion operation, you should output the value of the element that was taken from the array.

Input

The first line of the input contains one integer N — the number of queries ($1 \le n \le 5 \cdot 10^5$).

Then descriptions of the queries follow. Each query starts with one integer denoting the query type, then for the queries of type 1, 2 and 5 the integer x follows ($1 \le x \le 5 \cdot 10^5$).

Output

For each query of types 3, 4 and 6 print the result of the query (the value of the deleted element or the requested maximum).

Example

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

Problem G. Guess The Integer

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

This is an interactive problem.

The jury program is secretly written down some integer N ($1 \le N \le 10^5$). You may ask the query — tell the integer K ($1 \le K \le 4 \cdot 10^4$) and the jury program will receive the value of $N^N \bmod K$. Your task is to guess N in no more than two queries.

Interaction Protocol

In the beginning, the jury program tells you one integer T ($1 \le T \le 5000$) — the number of the games to be played.

In each game, the interaction is started by your program by sending the integer K. The query format is "`?` K", where $1 \le K \le 4 \cdot 10^4$ is integer. The jury program will answer the value $N^N \bmod K$ or -1 if the query has an invalid format, K is out of range, or if you asked more than twice.

After answering the query, you shall name the integer N. To do that, you shall print the string "`!` N", where N is the guessed value. If the jury program prints 1, then the answer is guessed correctly and you may start the next game, if the jury program prints -1, it means that your guess was wrong.

In case when you got -1 after the query or after the answer, your program shall terminate immediately, otherwise, you may get a random error code instead of Wrong Answer.

Example

standard input	standard output
1	
	? 2021
27	
	! 3

Note

To avoid Wall-Time-Limit error, please print the newline character after the query or answer, and flush the output buffer using the following functions:

- In Pascal: **flush(output)**;
- In C/C++: **fflush(stdout)** or **cout.flush()**;
- In Java: **System.out.flush()**;
- In Python: **sys.stdout.flush()** from the library **sys**;

Problem H. Hardware

Input file:	*standard input*
Output file:	*standard output*
Time limit:	5 seconds
Memory limit:	512 mebibytes

There are N computers connected by M wires. Wire i connects computers U_i and V_i, and can transmit a packet of size at most M_i bytes. All M_i are distinct integers. Cost of transmitting one packet through i-th wire is C_i. Data of size K can be transmitted through the i-th wire by splitting it in $\lceil \frac{K}{M_i} \rceil$ packets.

You should answer Q queries. For each query i you are given an integer K_i. The answer to i-th query is the minimum cost needed to send data of size K_i bytes to all computers starting from computer 1. When a computer receives the data it can pass it through multiple wires.

Input

The first line contains two integers N and M ($2 \le N \le 10^5$, $N - 1 \le M \le 10^5$).

The next M lines each contain 4 integers. The i-th line contains integers U_i, V_i, M_i and C_i ($1 \le U_i, V_i \le N$, $U_i \ne V_i$, $1 \le M_i, C_i \le 10^6$). All M_i are distinct.

The next line contains an integer Q ($1 \le Q \le 10^5$).

The next Q lines each contain an integer K_i ($1 \le K_i \le 10^6$).

Output

Print the answer to each query, in separate lines.

Example

standard input	standard output
3 3 1 2 1 3 1 3 2 1 2 3 3 7 2 1 3	4 9

Problem I. Investigation For Quadropalindromes

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	512 mebibytes

Let's call the string *quadropalindrome* if:

- The string is a palindrome and
- it is possible either split the string by two **palindromes** of **equal** length, either add one letter and then split the resulting string by two **palindromes** of **equal** length.

For example, "aaa" is the quadropalindrome (it is the palindrome, and after inserting of the letter 'a' in any place of the string it can be split by two palindromes 'aa" and 'aa' of equal length; "zz" is the quadropalindrome (it is the palindrome and it can be split by two palindromes 'z' and 'z' of equal length). The strings "abba" nor "abaabaaba" are not quadropalindromes.

Given the string. Your task is to remove a minimal possible number of letters and reorder the rest to obtain the quadropalindrome. If several quadropalindromes are possible to obtain, choose **lexicographically minimal** one.

Input

The first line of the input contains one non-empty string S ($1 \le |S| \le 10^5$), consisting of lowercase English letters.

Output

Print one string — the answer to the problem.

Example

standard input	standard output
zyyzzzzxx	zxzzxz

Problem J. Jeopardy

Input file:	*standard input*
Output file:	*standard output*
Time limit:	5 seconds
Memory limit:	512 mebibytes

Bytica is organizing the tournament on Jeopardy. One game in Jeopardy is played for four players. There are 16 players in the final part of the tournament. Bytica plans to schedule the competition such as every two players are played together in exactly one game (of course, each game contains four different players).

But Bytica comes to the work after the previous competition director, Byteasar, who did not worry much about the schedule, already assigned players to several games that were already played.

Given the ids of players in the played games, check if it is possible to add some games to the schedule such as each player will play exactly one game versus any other player, and if it is possible, find the minimal set of games that should be added to the schedule.

Input

The first line of the input contains one integer N ($1 \leq N \leq 25$) — the number of the games that are already played.

Then N lines follow; i-th of them is containing four pairwise distinct integers between 1 and 16, inclusively — the id's of the players that are involved in the i-th of the played games.

Output

If it is impossible to add some (possibly zero) games such as each player will play exactly one game versus any other player, print -1. Otherwise, print in the first line number of the games k that should be added to the schedule, then in the following k lines print the added games in the same format as the input file. If there is more than one answer, print any of them.

Examples

standard input	standard output
19 12 15 9 10 8 9 13 2 14 2 10 4 2 7 12 1 9 4 16 1 12 13 16 11 11 5 15 2 1 11 3 10 13 1 14 5 16 5 8 10 14 6 9 11 14 15 16 7 2 6 3 16 12 8 14 3 7 4 11 8 6 12 5 4 3 9 7 5 6 15 8 1 10 6 13 7	3 15 4 13
4 1 2 3 4 5 6 7 8 9 10 11 12 14 15 1 3	-1

Problem K. K Swaps

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	512 mebibytes

You are given a permutation p of size n.

Let the function $F(p)$ be the minimum number of swaps needed to sort the permutation p. If you perform exactly k swaps to the permutation p and obtain the permutation q, what are the maximum and minimum $F(q)$ possible?

Notice that you are **not allowed** to swap an element with itself.

Input

The first line of the input contains two integers n and k ($2 \le n \le 10^5$, $0 \le k \le 10^9$).

Then the permutation p of size n is placed at the second line.

Output

Print two integers — the maximum and minimum values of $F(q)$.

Examples

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

Problem L. Letters On Tree

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	512 mebibytes

You are given m strings and a tree on n nodes. Each edge has some letter written on it.

You have to answer q queries. Each query is described by 4 integers u, v, l and r. The answer to the query is the total number of occurrences of $str(u, v)$ in strings with indices from l to r. $str(u, v)$ is defined as the string that is made by concatenating letters written on the edges on the shortest path from u to v (in order that they are traversed).

Input

The first line of the input contains three integers n, m and q ($2 \le n \le 10^5$, $1 \le m, q \le 10^5$).

The i-th of the following $n - 1$ lines contains two integers u_i, v_i and a lowercase Latin letter c_i ($1 \le u_i, v_i \le n$, $u_i \ne v_i$), denoting the edge between nodes u_i, v_i with a character c_i on it.

It's guaranteed that these edges form a tree.

The following m lines contain the strings consisting of lowercase Latin letters. The total length of those strings does not exceed 10^5.

Then q lines follow, each containing four integers u, v, l and r ($1 \le u, v \le n$, $u \ne v$, $1 \le l \le r \le m$), denoting the queries.

Output

For each query print one integer — the answer to the query.

Examples

standard input	standard output
2 5 3 1 2 a aab abab aaa b a 2 1 1 5 1 2 1 3 2 1 3 5	8 7 4
9 5 6 1 2 a 2 7 c 1 3 b 3 4 b 4 6 b 3 5 a 5 8 b 5 9 c ababa cabbb bac bbbac abacaba 2 7 1 4 2 5 1 5 6 3 4 4 6 9 4 5 5 7 3 5 5 3 1 5	3 4 2 1 1 10

Problem M. Marked Cells

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	512 mebibytes

On an endless checkered sheet of paper, n cells are chosen and colored in three colors, where n is divisible by 3. It turns out that there are exactly $\frac{n}{3}$ marked cells of each of three colors!

Find the largest such k that it's possible to choose $\frac{k}{3}$ cells of each color, remove all other marked cells, and then select three rectangles with sides parallel to the grid lines, so that the following conditions hold:

- No two rectangles share a common cell (but they can share a side or its part).
- The i-th rectangle contains all the chosen cells of the i-th color and no chosen cells of other colors, for $i = 1, 2, 3$.

Input

The first line of the input contains one integer n — the number of the marked cells ($3 \le n \le 3 \times 10^5$, n is divisible by 3).

i-th of the following n lines contains description of one marked cell with index i — three integers x_i, y_i and c_i, denoting the coordinates of the cell and its color, respectively ($|x_i|, |y_i| \le 10^9$; $1 \le c_i \le 3$).

You may assume that number of marked cells of each color is the same (and equals $n/3$). You may assume that each cell is marked at most once.

Output

In the first line print one integer k — the number of the chosen cells.

In the second line print the space-separated list of the indices of the cells that are chosen.

Examples

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