Problem A. ZYB Loves Graph

Input file: `standard input`
Output file: `standard output`

You are given an undirected graph with N vertices and M edges. The edges are numbered from 1 to M. Denote the set $S(X)$ as: All the vertices we can reach from vertex X by **exactly** one edge.

You are supposed to deal with Q operations of the following two types:

- $(1, l, r)$ – **reverse** the status of edges numbered between l to r $(1 \le l \le r \le M)$. i.e. Delete the edge if it is in the graph, otherwise add it to the graph.
- $(2, u, v)$ – ask whether $S(u)$ and $S(v)$ are exactly the same $(1 \le u, v \le N)$.

Note that all the M edges are in the graph at the beginning.

Input

The input contains multiple cases. The first line of the input contains a single positive integer T, the number of cases.

For each case, the first line contains two integers N $(1 \le N \le 100000)$ and M $(1 \le M \le 200000)$, the number of vertices and edges in the graph. In the following M lines, the i-th line contains two integers u_i, v_i $(1 \le u_i, v_i \le N)$, describing the the i-th edge (u_i, v_i). Each edge appears in the input at most once. The $(M + 2)$-th line contains a integer Q $(1 \le Q \le 200000)$, the number of operations. In the following Q lines, each line contains three integers, describing an operation.

The total sum of N over all cases does not exceed 150000. The total sum of M over all cases does not exceed 800000. The total sum of Q over all cases does not exceed 600000.

Output

For each case, print a string in a line. The length of the string should equal the number of operations of type 2. If the answer is yes, the i-th character of the string should be '1', otherwise it should be '0'. Check the samples for more details.

Example

<table>
<tr><th>standard input</th><th>standard output</th></tr>
<tr><td><pre>1
5 4
1 2
1 3
4 2
4 3
3
2 1 4
1 1 1
2 1 2</pre></td><td><pre>10</pre></td></tr>
</table>

Problem B. Crazy Binary String

Input file:	`standard input`
Output file:	`standard output`

ZYB loves binary strings (strings that only contains '0' and '1'). And he loves *equal binary strings* more, where the number of '0' and the number of '1' in the string are equal.

ZYB wants to choose a substring from an original string T so that it is an *equal binary string* with the longest length possible. He also wants to choose a subsequence of T which meets the same requirements.

A string v is a substring of a string w if v is empty, or there are two integers l and r ($1 \le l \le r \le |w|$) such that $v = w_l w_{l+1} \cdots w_r$. A string v is a subsequence of a string w if it can be derived from w by deleting any number (including zero) of characters without changing the order of the remaining characters.

For simplicity, you only need to output the maximum possible length. Note that the empty string is both a substring and a subsequence of any string.

Input

The first line of the input contains a single integer N ($1 \le N \le 100000$), the length of the original string T. The second line contains a binary string with exactly N characters, the original string T.

Output

Print two integers A and B, denoting the answer for substring and subsequence respectively.

Example

standard input	standard output
8 01001001	4 6

Problem C. Guessing ETT

Input file: `standard input`
Output file: `standard output`

ZYB is a smart guy. One day he learns a new method for representing trees: Euler tour technique (ETT). You can find more details about ETT on this web page: `https://en.wikipedia.org/wiki/Euler_tour_technique`.

If we use vertices rather than edges in ETT, then any tree with N vertices corresponds to a sequence of length $2N - 1$, let's call it the vertex-ETT sequence.

In the beginning, ZYB generates a tree and writes down its vertex-ETT sequence. However, he spilt ink onto the paper by mistake and some numbers were covered in ink. Can you help him to restore the sequence? For convenience, it is guaranteed that at least one valid sequence exists.

Input

There are multiple cases. The first line of the input contains a single positive integer T, the number of cases.

For each case, the first line contains a single integer N ($1 \le N \le 2.5 \times 10^5$), while the second line contains an integer sequence a ($1 \le a_i \le N$ or $a_i = -1$, which means this number was covered by the ink) of length $2N - 1$, the vertex-ETT sequence.

It's guaranteed that the sum of N over all test cases does not exceed 500000. Due to the large size of the input, it's recommended to use a fast way to read the input.

Output

For each case, print $2N-1$ space-separated integers, the recovered sequence. If there are multiple solutions, print any of them.

Example

standard input	standard output
2 3 1 2 1 -1 1 3 -1 2 3 -1 1	1 2 1 3 1 1 2 3 2 1

Problem D. Big Integer

Input file: `standard input`
Output file: `standard output`

For little pupils, a very large number usually means an integer with many many digits. Let's define a class of big integers which consists only of the digit one ($11\cdots1$). The first few integers in this class are $1, 11, 111, 1111\cdots$. Denote $A(n)$ as the n-th smallest integer in this class. To make it even larger, we consider integers in the form of $A(a^b)$. Now, given a prime number p, how many pairs (i, j) are there such that $i \le n,\ j \le m,\ A(i^j) \equiv 0(mod\ p)$.

Input

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

For each case, the input consists of a single line, which contains 3 positive integers p, n, m $(p, n, m \le 10^9)$.

Output

Print the answer, a single integer, in one separate line for each case.

Example

standard input	standard output
2 11 8 1 7 6 2	4 2

Problem E. Trees in the Pocket II

Input file: `standard input`
Output file: `standard output`

DreamGrid has just found two trees, both consisting of n vertices, in his right pocket. Each edge in each tree has its own weight. The i-th edge in the first tree has a weight of a_i, while the i-th edge in the second tree has a pair of weight, denoted by (b_i, c_i).

Let ${}^1u$ be the u-th vertex in the first tree, and ${}^2u$ be the u-th vertex in the second tree. Let $\mathbb{E}_1({}^1u, {}^1v)$ be the set containing the **indices** of all the edges on the path between the u-th and the v-th vertex in the first tree. Similarly, let $\mathbb{E}_2({}^2u, {}^2v)$ be the set containing the **indices** of all the edges on the path between the u-th and the v-th vertex in the second tree. Define the following values:

- $A_{\min}({}^1u, {}^1v) = \min\{a_k | k \in \mathbb{E}_1({}^1u, {}^1v)\}$
- $B_{\max}({}^2u, {}^2v) = \max\{b_k | k \in \mathbb{E}_2({}^2u, {}^2v)\}$
- $C_{\max}({}^2u, {}^2v) = \max\{c_k | k \in \mathbb{E}_2({}^2u, {}^2v)\}$

As DreamGrid is bored, he decides to count the number of good indices. DreamGrid considers an index i ($1 \le i \le n$) is good, if for all $1 \le j \le n$ and $j \ne i$, $A_{\min}({}^1i, {}^1j) \ge \min(B_{\max}({}^2i, {}^2j), C_{\max}({}^2i, {}^2j))$. Please help DreamGrid figure out all the valid indices.

You may have seen this problem before, but this time **we need you to have an $O(N \log N)$ solution**, or it may not pass.

Input

There are multiple test cases. The first line contains an integer T, indicating the number of test cases. For each test case:

The first line contains an integer n ($2 \le n \le 10^5$), indicating the number of vertices in both trees.

For the following $(n-1)$ lines, the i-th line contains three integers ${}^1u_i$, ${}^1v_i$ and a_i ($1 \le {}^1u_i, {}^1v_i \le n$, $1 \le a_i \le 10^9$), indicating that there is an edge, whose weight is a_i, connecting vertex u_i and v_i in the first tree.

For the following $(n-1)$ lines,the i-th line contains four integers ${}^2u_i$, ${}^2v_i$, b_i and c_i ($1 \le {}^2u_i, {}^2v_i \le n$, $1 \le b_i^2, c_i^2 \le 10^9$), indicating that there is an edge, whose weight is (b_i, c_i), connecting vertex u_i and v_i in the second tree.

It's guaranteed that the sum of n of all test cases does not exceed 1.5×10^5.

Output

For each test case output k integers (k is the number of valid indices) in one line separated by a space in increasing order, indicating the indices of the valid vertices.

Note that if there is no valid vertex, you should print "-1" instead.

Example

standard input	standard output
2 2 1 2 1 1 2 2 3 4 1 2 7 1 3 8 1 4 12 1 2 8 8 2 3 9 7 2 4 6 4	-1 3 4

Problem F. Planting Trees

Input file: `standard input`
Output file: `standard output`

The semester is finally over and the summer holiday is coming. However, as part of your university's graduation requirement, you have to take part in some social service during the holiday. Eventually, you decided to join a volunteer group which will plant trees in a mountain.

To simplify the problem, let's represent the mountain where trees are to be planted with an $N \times N$ grid. Let's number the rows 1 to N from top to bottom, and number the columns 1 to N from left to right. The elevation of the cell in the i-th row and j-th column is denoted by $a_{i,j}$. Your leader decides that trees should be planted in a rectangular area within the mountain and that the maximum difference in elevation among the cells in that rectangle should not exceed M. In other words, if the coordinates of the top-left and the bottom-right corners of the rectangle are (x_1, y_1) and (x_2, y_2), then the condition $|a_{i,j} - a_{k,l}| \leq M$ must hold for $x_1 \leq i, k \leq x_2,\ y_1 \leq j, l \leq y_2$. Please help your leader calculate the maximum possible number of cells in such a rectangle so that he'll know how many trees will be planted.

Input

The input contains multiple cases. The first line of the input contains a single integer T $(1 \leq T \leq 500)$, the number of cases.

For each case, the first line of the input contains two integers N $(1 \leq N \leq 1000)$ and M $(0 \leq M \leq 10^5)$. The following N lines each contain N integers, where the j-th integer in the i-th line denotes $a_{i,j}$ $(1 \leq a_{i,j} \leq 10^5)$.

It is guaranteed that the sum of N^3 over all cases does not exceed $25 \cdot 10^7$.

Output

For each case, print a single integer, the maximum number of cells in a valid rectangle.

Example

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

Problem G. Removing Stones

Input file: `standard input`
Output file: `standard output`

Summer vacation is coming and Mark has returned home from his university having successfully survived the exam week. Today, he is very bored. So his friend Alice challenges him to play a game with stones which is invented by her. Alice gives Mark N piles of stones numbered from 1 to N, and there are a_i stones in the i-th pile. The rules of the game are simple: Mark will try to remove all stones. In each move, Mark chooses two different non-empty piles and removes one stone from each of those two piles. Mark can perform any number of moves. If all the piles are empty after some number of moves, Mark wins the game. If he can't make a valid move but not all piles are empty, he loses the game. Obviously, if the total number of stones is odd, then Mark is not able to win the game. So there is an additional rule: if initially, the total number of stones is odd, then Mark removes a single stone from the pile with the fewest stones before starting the game. If there are multiple piles with the smallest number of stones, Mark chooses one among them to remove a stone.

Mark found the optimal strategy for Alice's game very quickly and gets bored again. Also, he noticed that for some configuration of stones there is no way to win. So he challenges you to solve this problem: count the number of integer pairs (l, r) $(1 \le l < r \le N)$ such that it is possible for Mark to win the game if the game is played using only the piles numbered from l to r.

Input

The input contains multiple cases. The first line of the input contains a single positive integer T, the number of cases.

The first line of each case contains a single integer N $(2 \le N \le 300000)$, the number of piles. The following line contains N space-separated integers, where the i-th integer denotes a_i $(1 \le a_i \le 10^9)$, the number of stones in the i-th pile.

It is guaranteed that the sum of N over all cases does not exceed 1000000.

Output

For each case, print a single integer on a single line, the number of pairs satisfying the required property.

Example

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

Problem H. Magic Line

Input file: `standard input`
Output file: `standard output`

There are always some problems that seem simple but is difficult to solve.

ZYB got N distinct points on a two-dimensional plane. He wants to draw a magic line so that the points will be divided into two parts, and the number of points in each part is the same. There is also a restriction: this line can not pass through any of the points.

Help him draw this magic line.

Input

There are multiple cases. The first line of the input contains a single integer T ($1 \le T \le 10000$), indicating the number of cases.

For each case, the first line of the input contains a single even integer N ($2 \le N \le 1000$), the number of points. The following N lines each contains two integers x_i, y_i ($|x_i, y_i| \le 1000$), denoting the x-coordinate and the y-coordinate of the i-th point.

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

Output

For each case, print four integers x_1, y_1, x_2, y_2 in a line, representing a line passing through (x_1, y_1) and (x_2, y_2). Obviously the output must satisfy $(x_1, y_1) \ne (x_2, y_2)$.

The absolute value of each coordinate must not exceed 10^9. It is guaranteed that at least one solution exists. If there are multiple solutions, print any of them.

Example

standard input	standard output
1 4 0 1 -1 0 1 0 0 -1	-1 999000000 1 -999000001

Problem I. Median

Input file:	`standard input`
Output file:	`standard output`

JSB has an integer sequence $a_1, a_2, \ldots, a_n$. He wants to play a game with SHB.

For each $1 \le i \le n-2$, JSB calculates the median of $\{a_i, a_{i+1}, a_{i+2}\}$, denoted by b_i. SHB is given the sequence $b_1, b_2, \ldots, b_{n-2}$. Can you help him restore the sequence a?

Recall that the median of three numbers is the second largest one of them.

Input

There are multiple cases. The first line of the input contains a single positive integer T, indicating the number of cases.

For each case, the first line of the input contains a single integer n ($3 \le n \le 10^5$), the length of the sequence a. The second line contains $n-2$ integers $b_1, b_2, \ldots, b_{n-2}$ ($0 \le b_i \le 10^9$).

It's guaranteed that the sum of n over all cases does not exceed 10^6.

Output

For each case print one line containing n integers, indicating the sequence $a_1, a_2, \ldots, a_n$. Your output must satisfy $0 \le a_i \le 10^9$ for each $1 \le i \le n$.

If there are multiple valid answers, you may print any of them. If there is no valid answer, print "-1" (without quotes) instead.

Example

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

Problem J. ZYB Loves LRU

Input file: `standard input`
Output file: `standard output`

ZYB has finished his computer course recently. He is very interested in the LRU algorithm for cache management.

To simplify the problem, assume that a block contains a name (which is a string) and a set of data (which is a number). ZYB wants to implement the LRU algorithm with an array.

His array can hold at most M elements at any time. In the beginning, the array is empty. In each operation, the CPU may access a block s. ZYB will search for it in his array by brute force. If this block is present in his array (which means he can find a block with the same name), he takes it out of the array and puts it back at the end of the array. Otherwise, he simply adds it to the end of the array. If at any time, the size of the array exceeds the capacity, he will remove the first block in his array (the block at the front of the array).

Seems boring? Well, sometimes ZYB may ask the data of a block in, before or after a certain block. Could you help him to write a program to achieve his goal?

Input

There are multiple cases. The first line of the input contains a single positive integer T, indicating the number of cases.

For each case, the first line of the input contains two integers Q and M ($1 \leq Q,\ M \leq 500000$), denoting the number of operations and the capacity of the array. The following Q lines each contain an integer opt ($0 \leq opt \leq 1$), a string s ($1 \leq |s| \leq 10$) and an integer v, separated by single spaces, describing an operation.

- If $opt = 0$, then $|v| < 10$, and the operation is that the CPU wants to access a block. If this access misses (which means you can't find a block in the array with the name s), add a block at the end of the array with the name s and the data v, and the result of the operation is v. If this access hits, the result of the operation is the data of the block you found (ignore v in this case). Don't forget to move that block to the end of the array.
- If $opt = 1$, then v will be -1, 0 or 1, and the operation is that you should answer ZYB's question. Let k be the index of the block with the name s in the array. Then the result of this operation is the data of the block with the index $k + v$ in the array. If such block doesn't exist, please output "Invalid" (without quotation marks) instead.

Note that ZYB's questions (operations of type 1) don't count as access and don't cause the array to be updated.

It is guaranteed that the sum of Q over all cases does not exceed 1000000, and that s only contains digits (i.e. '0' - '9').

Output

For each case, print the result of each operation in a line as defined in the input section of the statement.

Example

standard input	standard output
1 8 3 0 0101010 1 0 0101011 2 1 0101010 1 0 1100000 3 0 0101011 -1 0 1111111 4 1 0101011 -1 1 0101010 0	1 2 2 3 2 4 3 Invalid