,

Problem A. So I'll Max Out My Constructive Algorithm Skills

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

BaoBao the Witch is stuck in a maze with n rows and n columns, where the height of the cell in the i-th row and the j-th column is $h_{i,j}$. To get out of the maze, BaoBao has to find a path which passes through each cell exactly once. Each time she can only move into the neighboring cell sharing a same edge with the current one. But as we know, BaoBao is super lazy, so every time when she climbs up (that is to say, moving from a cell with a smaller height to another with a larger height) her happiness value will decrease. As her helping hand, your task is to find a valid path so that when moving along the path, the number of times BaoBao climbs up will not be more than the number of times she climbs down.

More formally, you need to find a sequence $(x_1, y_1), (x_2, y_2), \cdots, (x_{n^2}, y_{n^2})$ such that:

- For all $1 \le i \le n^2$, $1 \le x_i, y_i \le n$;
- For all $1 \le i, j \le n^2, i \ne j$, $(x_i, y_i) \ne (x_j, y_j)$;
- For all $2 \le i \le n^2$, $|x_i - x_{i-1}| + |y_i - y_{i-1}| = 1$;
- $\sum_{i=2}^{n^2} [h_{x_{i-1},y_{i-1}} < h_{x_i,y_i}] \le \sum_{i=2}^{n^2} [h_{x_{i-1},y_{i-1}} > h_{x_i,y_i}]$, where $[P]$ equals 1 when P is true, and equals 0 when it is false.

Additionally, you discover that the heights in all cells are a permutation of n^2, so you just need to output the height of each cell in a valid path.

Input

There are multiple test cases. The first line of the input contains an integer T ($1 \le T \le 100$) indicating the number of test cases. For each test case:

The first line contains an integer n ($2 \le n \le 64$) indicating the size of the maze.

For the following n lines, the i-th line contains n integers $h_{i,1}, h_{i,2}, \cdots, h_{i,n}$ ($1 \le h_{i,j} \le n^2$) where $h_{i,j}$ indicates the height of the cell in the i-th row and the j-th column. It's guaranteed that all integers in the input make up a permutation of n^2.

Output

For each test case output one line containing n^2 separated by a space indicating the heights of each cell in a valid path. If there are multiple valid answers you can output any of them. It's easy to prove that an answer always exists.

Please, DO NOT output extra spaces at the end of each line, or your answer may be considered incorrect!

Example

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

,

Problem B. Laser Trap

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	256 megabytes

BaoBao is playing the famous game *Elden Ring* these days. It's an open-world game in which you can control your character to travel from places to places. However, your character could also enter a trap and you need to figure out how to escape. Right now, BaoBao's character is stuck in a 2-dimensional plane with deadly lasers. There are n laser generators (each can be regarded as a point) shooting laser beams between every pair of them (so there are $\frac{n(n-1)}{2}$ laser beams in total). The beams start and end at generator points and do not stretch to infinity.

Starting at point $(0, 0)$, BaoBao wants to escape to point $(10^{10^{10^{10^{10}}}}, 10^{10^{10^{10^{10}}}})$ without touching any laser beam or generator. In order to do so, BaoBao can ask her friend DreamGrid to remove any number of laser generators, together with any laser beam that starts or ends at these generators. Output the minimum number of laser generators that need to be erased for the escape.

Note that BaoBao does not need to move in a specific direction to escape. Her escaping route can even be a curve if necessary.

Input

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

The first line contains an integer n ($1 \le n \le 10^6$) indicating the number of laser generators.

For the following n lines, the i-th line contains two integers x_i and y_i ($-10^9 \le x_i, y_i \le 10^9$) indicating the location of the i-th laser generator.

It is guaranteed that no two generators coincide, and no laser beam or generator will touch $(0, 0)$.

It is also guaranteed that the sum of n of all test cases will not exceed 10^6.

Output

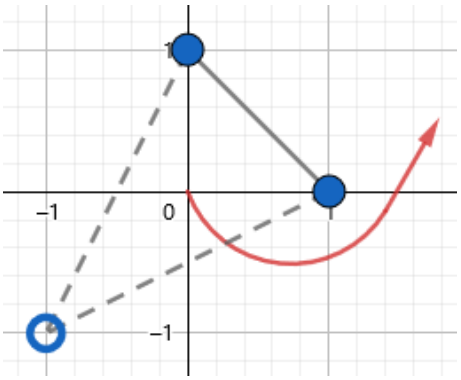
For each test case output one line containing one integer indicating the minimum number of generators that need to be removed.

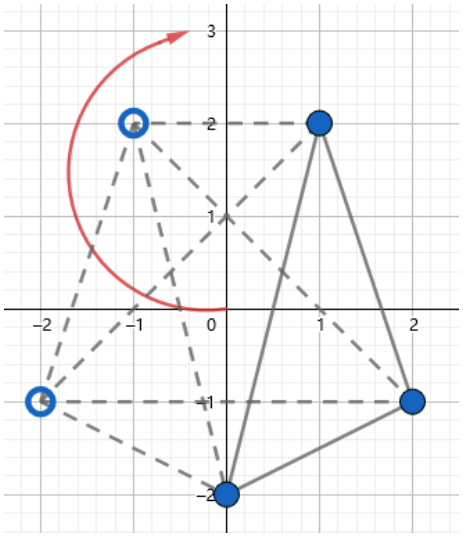
Example

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

Note

The second and the third sample test cases are shown below. Solid dots and lines represent the remaining laser generators and beams, while hollow dots and dashed lines represent the removed laser generators and beams. The arrow is the escaping route.





Problem C. Sandpile on Clique

Input file:	standard input
Output file:	standard output
Time limit:	1 second
Memory limit:	256 megabytes

The *Abelian Sandpile Model* is a famous dynamical system displaying self-organized criticality. It has been studied for decades since it was introduced by Per Bak, Chao Tang and Kurt Wiesenfeld in a 1987 paper. The sandpile prediction is of wide interest in physics, computer science, and mathematics, both for its beautiful algebraic structure and for its relevance to applications like load balancing and derandomization of models like internal diffusion-limited aggregation. The sandpile model is related to many other models and physical phenomena, like the rotor-routing model, avalanche models.

In the sandpile model, we are given an undirected graph G whose vertices are indexed from 1 to n. We're also given n integers $a_1, a_2, \cdots, a_n$ where a_i indicates that there are a_i chips placed on vertex i initially. Each turn we will pick an arbitrary vertex v such that the number of chips on v is not smaller than the number of edges connecting v, denoted as d_v. For each neighbor of v, it will receive one chip from v. Therefore, v will lost d_v chips. This process is called firing or toppling. Firing will keep happening until no vertex v has at least d_v chips.

It can be proven that the order of firing doesn't affect the result. Meanwhile, it is also possible that the firing will never terminate. This instance is described as "recurrent". Now you are given a clique and the initial number of chips. Determine whether this instance is a recurrent one. If not, please output the final number of chips for each node respectively.

A clique (also called a complete graph) is a graph where every two vertices are connected with an edge.

Input

There is only one test case in each test file.

The first line of the input contains an integer n ($2 \le n \le 5 \times 10^5$) indicating the size of the clique.

The second line contains n integers $a_1, a_2, \cdots, a_n$ ($0 \le a_i \le 10^9$) where a_i indicates the initial number of chips placed on vertex i.

Output

Output one line. If the given sandpile instance will terminate, output n integers separated by a space where the i-th integer indicates the final number of chips on the i-th vertex. Otherwise output "Recurrent" (without quotes) instead.

Please, DO NOT output extra spaces at the end of each line or your solution may be considered incorrect!

Examples

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

Note

For the first sample test case:

- We can only select vertex 1 at the beginning. The number of chips becomes $\{1, 1, 4, 1, 4\}$.
- We can now select vertex 3 or 5 because both of them have at least 4 chips. We select vertex 3 and the number of chips becomes $\{2, 2, 0, 2, 5\}$. Selecting vertex 5 will lead to the same result.

- We now select vertex 5. The number of chips becomes $\{3, 3, 1, 3, 1\}$. There is no vertex with at least 4 chips so the firing terminates.

For the second sample test case, we can select vertex 1 and 2 repeatedly. The firing never terminates.

Problem D. Link-Cut Tree

Input file:	standard input
Output file:	standard output
Time limit:	2 seconds
Memory limit:	256 megabytes

BaoBao just learned how to use a data structure called link-cut tree to find cycles in a graph and decided to give it a try. BaoBao is given an undirected graph with n vertices and m edges, where the length of the i-th edge equals 2^i. She needs to find a simple cycle with the smallest length.

A simple cycle is a subgraph of the original graph containing k ($3 \leq k \leq n$) vertices $a_1, a_2, \cdots, a_k$ and k edges such that for all $1 \leq i \leq k$ there is an edge connecting vertices a_i and $a_{(i \mod k)+1}$ in the subgraph. The length of a simple cycle is the total length of the edges in the cycle.

Input

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

The first line contains two integers n and m ($3 \leq n \leq 10^5$, $1 \leq m \leq 10^5$) indicating the number of vertices and edges in the original graph.

For the following m lines, the i-th line contains two integers u_i and v_i ($1 \leq u_i, v_i \leq n$) indicating an edge connecting vertices u_i and v_i with length 2^i. There are no self loops nor multiple edges. Note that the graph is not necessarily connected.

It's guaranteed that neither the sum of n nor the sum of m of all test cases will exceed 10^6.

Output

For each test case output one line. If there are no simple cycles in the graph output "-1" (without quotes); Otherwise output k integers separated by a space in increasing order indicating the indices of the edges in the simple cycle with the smallest length. It can be shown that there is at most one answer.

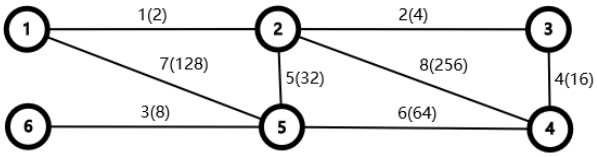
Please, DO NOT output extra spaces at the end of each line, or your solution may be considered incorrect!

Example

standard input	standard output
2 6 8 1 2 2 3 5 6 3 4 2 5 5 4 5 1 4 2 4 2 1 2 4 3	2 4 5 6 -1

Note

The first sample test case is shown below. The integers beside the edges are their indices (outside the parentheses) and lengths (inside the parentheses). The simple cycle with the smallest length consists of edges 2, 4, 5 and 6 with a length of $2^2 + 2^4 + 2^5 + 2^6 = 116$.


1
2
3
4
5
6
1(2)
2(4)
7(128)
5(32)
8(256)
4(16)
3(8)
6(64)

Problem E. Colourful Tree

Input file:	standard input
Output file:	standard output
Time limit:	6 seconds
Memory limit:	512 megabytes

Your task is to maintain a colourful tree and process queries.

At the beginning, there is only one vertex numbered 1 with colour C on the tree. Then there are q operations of two types coming in order:

- $0\ x\ c\ d$: Add a new vertex indexed $(n+1)$ with colour c to the tree, where n is the current number of existing vertices. An edge connecting vertex x and $(n+1)$ with length d will also be added to the tree.
- $1\ x\ c$: Change the colour of vertex x to c.

After each operation, you should find a pair of vertices u and v $(1 \le u, v \le n)$ with **different** colours in the current tree so that the distance between u and v is as large as possible.

The distance between two vertices u and v is the length of the shortest path from u to v on the tree.

Input

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

The first line of the input contains two integers q and C $(1 \le q \le 5 \times 10^5,\ 1 \le C \le q)$ indicating the number of operations and the initial colour of vertex 1.

For the following q lines, each line describes an operation taking place in order with 3 or 4 integers.

- If the i-th line contains 4 integers 0, x_i, c_i and d_i $(1 \le x_i \le n,\ 1 \le c_i \le q,\ 1 \le d \le 10^9)$, the i-th operation will add a new vertex $(n+1)$ with colour c_i to the tree and connect it to vertex x_i with an edge of length d_i.
- If the i-th line contains 3 integers 1, x_i and c_i $(1 \le x_i \le n,\ 1 \le c_i \le q)$, the i-th operation will change the colour of vertex x_i to c_i.

It's guaranteed that the sum of q of all test cases will not exceed 5×10^5.

Output

For each operation output the maximum distance between two vertices with different colours. If no valid pair exists output 0 instead.

Example

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