Problem A. Square of Rectangles

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

Aryan is an ardent lover of squares but a hater of rectangles (Yes, he knows all squares are rectangles). But Harshith likes to mess with Aryan. Harshith gives Aryan three rectangles of sizes $l_1 \times b_1$, $l_2 \times b_2$, and $l_3 \times b_3$ such that $l_3 \leq l_2 \leq l_1$ and $b_3 \leq b_2 \leq b_1$. Aryan, in order to defeat Harshith, decides to arrange these three rectangles to form a square such that no two rectangles overlap and the rectangles are aligned along edges. Rotating rectangles is **not** allowed. Help Aryan determine if he can defeat Harshith.

Input

Each test contains multiple test cases. The first line contains the number of test cases t ($1 \leq t \leq 100$). The description of the test cases follows.

Each test case contains a single line with 6 space-separated integers $l_1, b_1, l_2, b_2, l_3,$ and b_3 ($1 \leq l_3 \leq l_2 \leq l_1 \leq 100$, $1 \leq b_3 \leq b_2 \leq b_1 \leq 100$) — the dimensions of the three rectangles.

Output

For each testcase, print "`YES`" if the rectangles can be arranged to form a square; otherwise, "`NO`".

You can output the answer in any case (upper or lower). For example, the strings "`yEs`", "`yes`", "`Yes`", and "`YES`" will be recognized as positive responses.

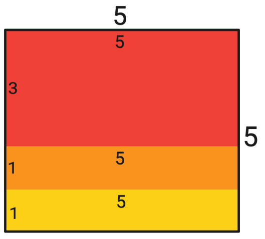
Example

standard input	standard output

5
100 100 10 10 1 1
5 3 5 1 5 1
2 3 1 2 1 1
8 5 3 5 3 3
3 3 3 3 2 1
	NO
	YES
	YES
	NO
	NO

Note

In the second test case, the three rectangles 5×3, 5×1, and 5×1 can be arranged as follows to form a square.



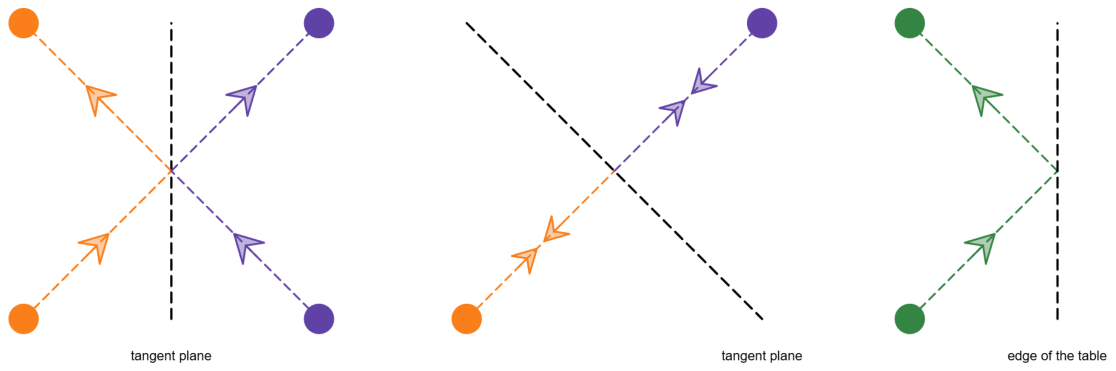
In the fourth test case, it can be proven that the rectangles can't be arranged to form a square with the given constraints.

Problem B. Square Pool

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

Aryan and Harshith are playing pool in universe AX120 on a fixed square pool table of side s with **pockets** at its 4 corners. The corners are situated at $(0, 0)$, $(0, s)$, $(s, 0)$, and (s, s). In this game variation, n identical balls are placed on the table with integral coordinates such that no ball lies on the edge or corner of the table. Then, they are all simultaneously shot at 10^{100} units/sec speed (only at 45 degrees with the axes).

In universe AX120, balls and pockets are almost point-sized, and the collisions are elastic, i.e., the ball, on hitting any surface, bounces off at the same angle and with the same speed.



Harshith shot the balls, and he provided Aryan with the balls' positions and the angles at which he shot them. Help Aryan determine the number of balls potted into the **pockets** by Harshith.

It is guaranteed that multiple collisions do not occur at the same moment and position.

Input

Each test contains multiple test cases. The first line contains the number of test cases t $(1 \le t \le 1000)$. The description of the test cases follows.

The first line of each test case contains two integers n and s $(1 \le n \le 10^3, 2 \le s \le 10^9)$ — the number of balls placed on the table and the side length of the square pool table.

The i-th of the next n lines contains four integers d_x, d_y, x_i, and y_i $(d_x, d_y \in \{-1, 1\}, 0 < x_i, y_i < s)$ — the direction vectors of the launch on the X-axis and Y-axis respectively, and the coordinates of the location where the i-th ball was placed. It is guaranteed that no two balls coincide at the initial moment.

It is also guaranteed that the sum of n over all test cases does not exceed 10^3.

Output

For each test case, print a single integer — the number of balls potted in that game.

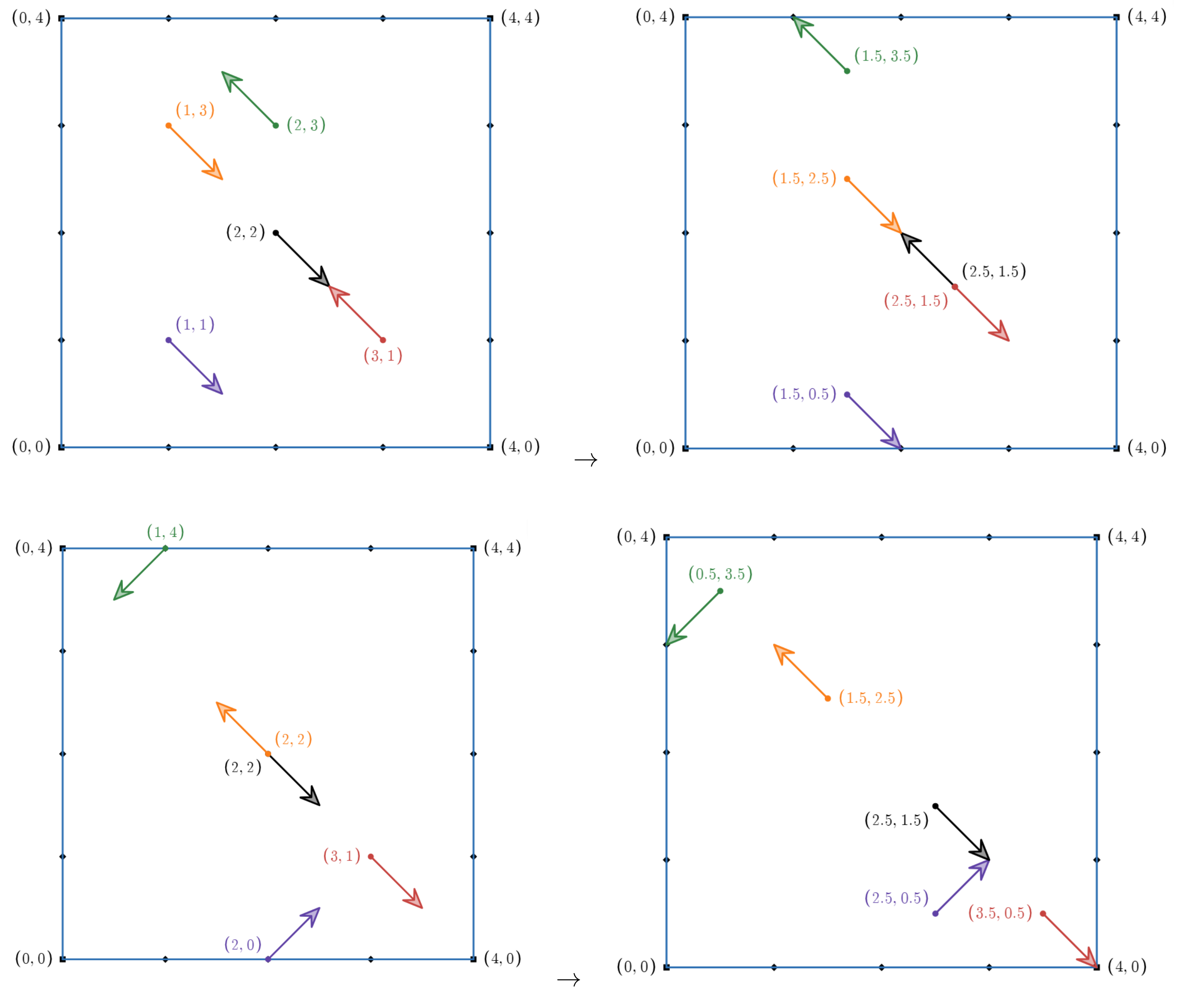
Example

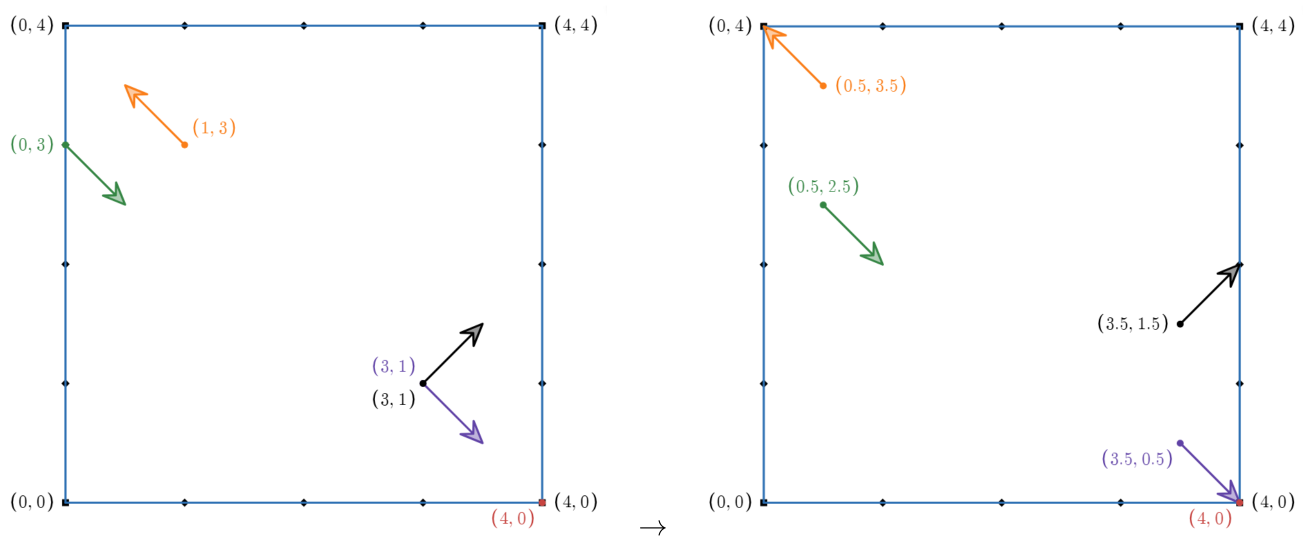
standard input	standard output
2	1
1 2	3
1 1 1 1	
5 4	
1 -1 1 1	
1 -1 2 2	
-1 1 2 3	
1 -1 1 3	
-1 1 3 1	

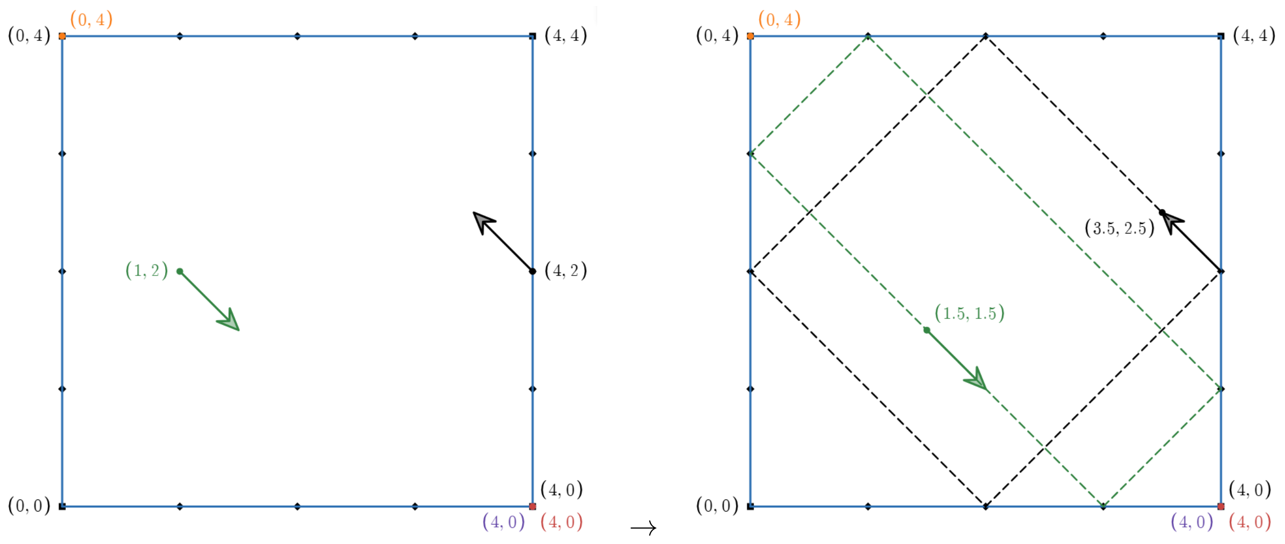
Note

In the first test case, there is a single ball and it's shot directly towards the pocket at $(2, 2)$, thus potted.

In the second test case, the state progresses as




(0, 4)
(4, 4)
(1, 3)
(0, 3)
(3, 1)
(3, 1)
(0, 0)
(4, 0)
(4, 0)
→
(0, 4)
(4, 4)
(0.5, 3.5)
(0.5, 2.5)
(3.5, 1.5)
(3.5, 0.5)
(0, 0)
(4, 0)
(4, 0)


(0, 4)
(0, 4)
(4, 4)
(1, 2)
(4, 2)
(0, 0)
(4, 0)
(4, 0)
(4, 0)
→
(0, 4)
(0, 4)
(4, 4)
(3.5, 2.5)
(1.5, 1.5)
(0, 0)
(4, 0)
(4, 0)
(4, 0)

Problem C. Divine Tree

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

Harshith attained enlightenment in Competitive Programming by training under a Divine Tree. A divine tree is a rooted tree* with n nodes, labelled from 1 to n. The divineness of a node v, denoted $d(v)$, is defined as the smallest node label on the unique simple path from the root to node v.

Aryan, being a hungry Competitive Programmer, asked Harshith to pass on the knowledge. Harshith agreed on the condition that Aryan would be given two positive integers n and m, and he had to construct a divine tree with n nodes such that the total divineness of the tree is m, i.e., $\sum_{i=1}^{n} d(i) = m$. If no such tree exists, Aryan must report that it is impossible.

Desperate for knowledge, Aryan turned to you for help in completing this task. As a good friend of his, help him solve the task.

Input

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

The first line of each test case contains two integers n and m ($1 \le n \le 10^6$, $1 \le m \le 10^{12}$).

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

Output

For each test case, output a single integer k in one line — the root of the tree.

Then $n-1$ lines follow, each containing a description of an edge of the tree — a pair of positive integers u_i, v_i ($1 \le u_i, v_i \le n$, $u_i \ne v_i$), denoting the i-th edge connects vertices u_i and v_i.

The edges and vertices of the edges can be printed in any order. If there are multiple solutions, print any of them.

If there is no solution, print "-1" instead.

Example

standard input	standard output

2
1 2
4 6
	-1
	3
	3 1
	1 2
	2 4

Note

In the first test case, there is a single node with a value of 1, so getting a sum of 2 is impossible.

In the second test case, getting a sum of 6 is possible with the given tree rooted at 3.

*A tree is a connected graph without cycles. A rooted tree is a tree where one vertex is special and called the root.

Problem D. Matrix game

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

Aryan and Harshith play a game. They both start with three integers a, b, and k. Aryan then gives Harshith two integers n and m. Harshith then gives Aryan a matrix X with n rows and m columns, such that each of the elements of X is between 1 and k(inclusive). After that, Aryan wins if he can find a submatrix[†] Y of X with a rows and b columns such that all elements of Y are equal.

For example, when $a = 2, b = 2, k = 6, n = 3$ and $m = 3$, if Harshith gives Aryan the matrix below, it is a win for Aryan as it has a submatrix of size 2×2 with all elements equal to 1 as shown below.

$$\begin{bmatrix} 1 & 2 & 1 \\ 3 & 4 & 5 \\ 1 & 6 & 1 \end{bmatrix} \xrightarrow{\text{removing 2nd row}} \begin{bmatrix} 1 & 2 & 1 \\ 1 & 6 & 1 \end{bmatrix} \xrightarrow{\text{removing 2nd column}} \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}$$

Example of a matrix where Aryan wins

Aryan gives you the values of a, b, and k. He asks you to find the lexicographically minimum tuple (n, m) that he should give to Harshith such that Aryan always wins. Help Aryan win the game. Assume that Harshith plays optimally. The values of n and m can be large, so output them modulo $10^9 + 7$. A tuple (n_1, m_1) is said to be lexicographically smaller than (n_2, m_2) if either $n_1 < n_2$ or $n_1 = n_2$ and $m_1 < m_2$.

Input

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

Each test case contains a single line with three space-separated integers a, b and k ($1 \le a, b, k \le 10^5$).

It is guaranteed that the sum of $\max(a, b, k)$ over all test cases does not exceed 10^5.

Output

For each test case, output a single line containing two space-separated integers n and m, denoting the answer to the problem. The values of n and m can be large, so output them modulo $10^9 + 7$.

Example

standard input	standard output

`3`
`1 1 5`
`2 2 2`
`90000 80000 70000`
	`1 1`
	`3 7`
	`299929959 603196135`

Note

For the first test case, every $n \times m$ matrix contains a 1×1 submatrix with all elements equal. $(1, 1)$ is the lexicographically minimum tuple among all of them.

For the second test case, it can be verified that whatever 3×7 matrix Harshith gives to Aryan, Aryan can always win by finding a 2×2 submatrix with all elements equal. $(3, 7)$ is also the lexicographically minimum tuple among all possible tuples where Aryan always wins.

[†]A submatrix of a matrix is obtained by removing some rows and/or columns from the original matrix.

Problem E. Lanes of Cars

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

Harshith is the president of TollClub. He tasks his subordinate Aryan to oversee a toll plaza with n lanes. Initially, the i-th lane has a_i cars waiting in a queue. Exactly one car from the front of each lane passes through the toll every second.

The angriness of a car is defined as the number of seconds it had to wait before passing through the toll. Consider it takes 1 sec for each car to pass the toll, i.e., the first car in a lane has angriness 1, the second car has angriness 2, and so on.

To reduce congestion and frustration, cars are allowed to switch lanes. A car can instantly move to the back of any other lane at any time. However, changing lanes increases its angriness by an additional k units due to the confusion caused by the lane change.

Harshith, being the awesome person he is, wants to help the drivers by minimising the total angriness of all cars. He asks Aryan to do so or get fired. Aryan is allowed to change lanes of any car anytime (possibly zero), but his goal is to find the minimum possible total angriness if the lane changes are done optimally. Help Aryan retain his job by determining the minimum angriness he can achieve.

Input

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

The first line of each test case contains two integers n and k ($1 \le n \le 2 \cdot 10^5$, $1 \le k \le 10^6$) — the number of lanes and the increment in angriness on a lane change.

The second line of each test case contains n space-separated integers, denoting array a — the i-th number representing the number of cars in the i-th lane ($1 \le a_i \le 10^6$).

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

Note that the sum of $\max a_i$ over all test cases is **not** bounded.

Output

For each test case, output a single integer in a new line, denoting the minimum total angriness.

Example

`standard input`

```
6
3 4
13 7 4
4 9
6 12 14 5
5 3
2 4 13 5 10
1 7
6
13 4
10 26 34 39 9 43 48 41 1 38 13 4 46
16 3
176342 171863 70145 80835 160257 136105
 78541 100795 114461 45482 68210 51656
  29593 8750 173743 156063
```

`standard output`

```
123
219
156
21
5315
82302351405
```

Note

In the first test case, Aryan shifts two cars from lane 1 to lane 3, after which the array becomes $[11, 7, 6]$. The total angriness is $\frac{11\cdot 12}{2} + \frac{7\cdot 8}{2} + \frac{6\cdot 7}{2} + 2 \cdot 4 = 123$. It can be proven that this is the minimum possible angriness.

In the fourth test case, there is only one lane, so cars can't shift lanes. Total angriness is $\frac{6\cdot 7}{2} = 21$.

Problem F. Superb Graphs

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

As we all know, Aryan is a funny guy. He decides to create fun graphs. For a graph G, he defines fun graph G' of G as follows:

- Every vertex v' of G' maps to a non-empty independent set[‡] or clique[§] in G.
- The sets of vertices of G that the vertices of G' map to are pairwise disjoint and combined cover all the vertices of G, i.e., the sets of vertices of G mapped by vertices of G' form a partition of the vertex set of G.
- If an edge connects two vertices v'_1 and v'_2 in G', then there is an edge between every vertex of G in the set mapped to v'_1 and every vertex of G in the set mapped to v'_2.
- If an edge does not connect two vertices v'_1 and v'_2 in G', then there is not an edge between any vertex of G in the set mapped to v'_1 and any vertex of G in the set mapped to v'_2.

As we all know again, Harshith is a superb guy. He decides to use fun graphs to create his own superb graphs. For a graph G, a fun graph G'' is called a superb graph of G if G'' has the minimum number of vertices among all possible fun graphs of G.

Aryan gives Harshith k simple undirected graphs[¶] $G_1, G_2, \ldots, G_k$, all on the same vertex set V. Harshith then wonders if there exist k other graphs $H_1, H_2, \ldots, H_k$, all on some other vertex set V' such that:

- G_i is a superb graph of H_i for all $i \in \{1, 2, \ldots, k\}$.
- If a vertex $v \in V$ maps to an independent set of size greater than 1 in one G_i, H_i $(1 \le i \le k)$ pair, then there exists no pair G_j, H_j $(1 \le j \le k, j \ne i)$ where v maps to a clique of size greater than 1.

Help Harshith solve his wonder.

Input

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

The first line of each test case contains two integers n and k $(1 \le n \le 300, 1 \le k \le 10)$.

Then, there are k graphs described. The first line of each graph description contains a single integer m $(0 \le m \le \frac{n\cdot(n-1)}{2})$.

Next m lines each contain two space-separated integers u and v $(1 \le u, v \le n, u \ne v)$, denoting that an edge connects vertices u and v.

It is guaranteed that the sum of m over all graphs over all test cases does not exceed $2 \cdot 10^5$, and the sum of n over all test cases does not exceed 300.

Output

For each testcase, print "`Yes`" if there exists k graphs satisfying the conditions; otherwise, "`No`".

[‡]For a graph G, a subset S of vertices is called an independent set if no two vertices of S are connected with an edge.
[§]For a graph G, a subset S of vertices is called a clique if every vertex of S is connected to every other vertex of S with an edge.
[¶]A graph is a simple undirected graph if its edges are undirected and there are no self-loops or multiple edges between the same pair of vertices.

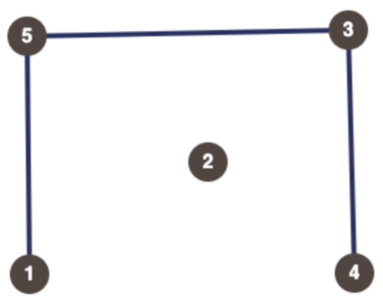
You can output the answer in any case (upper or lower). For example, the strings "`yEs`", "`yes`", "`Yes`", and "`YES`" will be recognized as positive responses.

Example

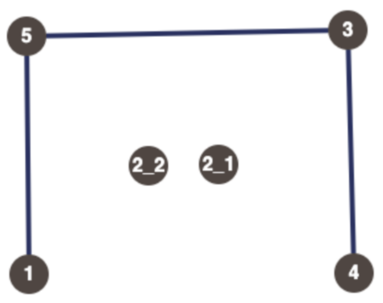
standard input	standard output
3	Yes
5 2	Yes
3	No
3 4	
5 3	
5 1	
6	
3 5	
3 4	
1 4	
1 2	
2 3	
4 2	
4 3	
0	
3	
3 1	
1 4	
1 2	
4	
4 2	
4 3	
1 2	
2 3	
3 2	
0	
3	
3 1	
3 2	
1 2	

Note

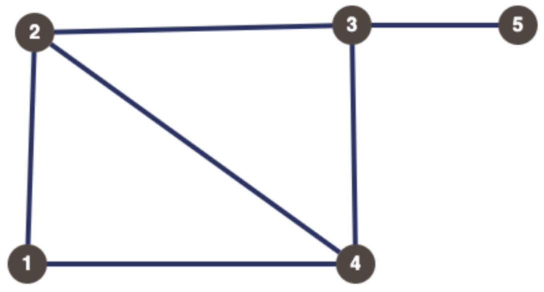
For the first test case, the following are the graphs of G_1, H_1 and G_2, H_2 such that G_1 is superb graph of H_1 and G_2 is superb graph of H_2.



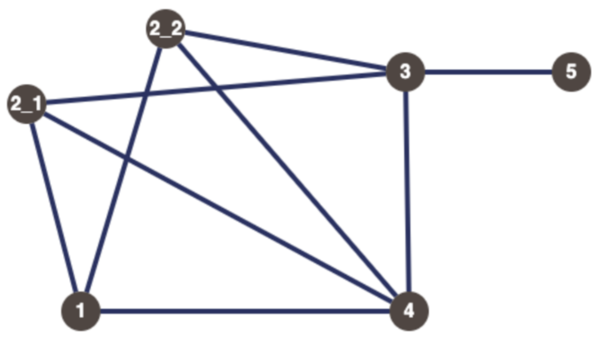
Graph G_1



Graph H_1



Graph G_2



Graph H_2

In each graph, vertex 2 of G_i corresonds to independent set $\{2\_1, 2\_2\}$ of corresponding H_i and remaining vertices $v \in \{1, 3, 4, 5\}$ of G_i correspond to independent set/clique $\{v\}$ in corresponding H_i(a single vertex set can be considered both an independent set and a clique).

In the third test case, it can be proven that the answer is "`No`".

Problem G. Eulerian Line Graph

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

Aryan loves graph theory more than anything. Well, no, he likes to flex his research paper on line graphs to everyone more. To start a conversation with you, he decides to give you a problem on line graphs. In the mathematical discipline of graph theory, the line graph of a simple undirected graph G is another simple undirected graph $L(G)$ that represents the adjacency between every two edges in G.

Precisely speaking, for an undirected graph G without self-loops or multiple edges, its line graph $L(G)$ is a graph such that

- Each vertex of $L(G)$ represents an edge of G.
- Two vertices of $L(G)$ are adjacent if and only if their corresponding edges share a common endpoint in G.

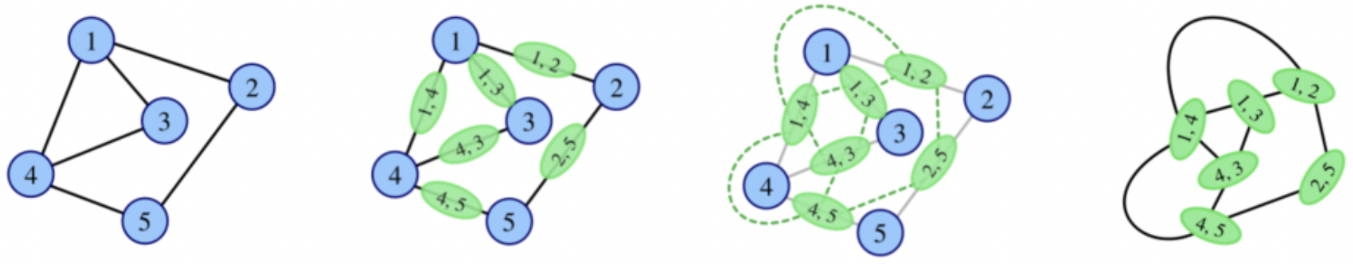


Figure: Generation of the Line Graph

Also, $L^0(G) = G$ and $L^k(G) = L(L^{k-1}(G))$ for $k \geq 1$.

An Euler trail is a sequence of edges that visits every edge of the graph exactly once. This trail can be either a path (starting and ending at different vertices) or a cycle (starting and ending at the same vertex). Vertices may be revisited during the trail, but each edge must be used exactly once.

Aryan gives you a simple connected graph G with n vertices and m edges and an integer k, and it is guaranteed that G has an Euler trail and it is not a path graph[‖]. He asks you to determine if $L^k(G)$ has an Euler trail.

Input

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

The first line of each test case contains three space-separated integers n, m, and k ($5 \leq n \leq 2 \cdot 10^5$, $n - 1 \leq m \leq \min(\frac{n \cdot (n-1)}{2}, 2 \cdot 10^5)$, $1 \leq k \leq 2 \cdot 10^5$).

The next m lines of each test case contain two space-separated integers u and v ($1 \leq u, v \leq n$, $u \neq v$), denoting that an edge connects vertices u and v.

It is guaranteed that the sum of n and m over all test cases does not exceed $2 \cdot 10^5$.

Output

For each testcase, print "`YES`" if $L^k(G)$ has an Euler trail; otherwise, "`NO`".

You can output the answer in any case (upper or lower). For example, the strings "`yEs`", "`yes`", "`Yes`", and "`YES`" will be recognized as positive responses.

[‖]A path graph is a tree where every vertex is connected to atmost two other vertices.

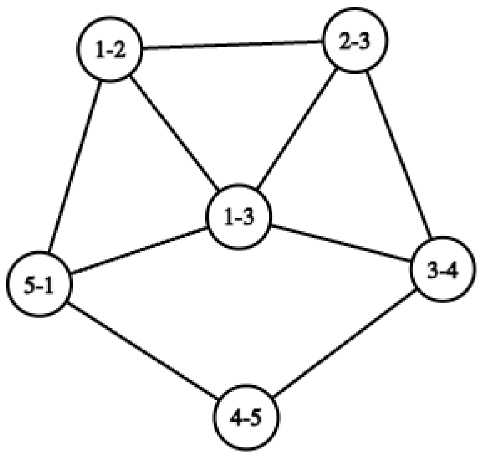
Example

standard input	standard output
4	YES
5 5 2	NO
1 2	YES
2 3	NO
3 4	
4 5	
5 1	
5 6 1	
1 2	
2 3	
3 4	
4 5	
5 1	
1 3	
10 11 3	
1 2	
2 3	
3 4	
4 5	
4 6	
4 7	
5 7	
6 7	
7 8	
8 9	
9 10	
7 8 2	
1 3	
2 3	
1 4	
4 5	
2 5	
1 6	
6 7	
2 7	

Note

For the first test case, $L^2(G)$ is isomorphic to G itself. So, since G has an Euler trail, $L^2(G)$ also has an Euler trail.

For the second test case, $L(G)$ looks as follows(Vertex $i - j$ of $L(G)$ in figure corresponds to edge between vertices i and j of G). It can be proven that this doesn't have an Euler trail.


1-2
2-3
1-3
5-1
3-4
4-5