



Information

Memory limit

- In problem A: 256 megabytes
- In problem B: 256 megabytes
- In problem C: 256 megabytes
- In problem D: 256 megabytes
- In problem E: 1024 megabytes

Source code limit

The size of each solution source code can't exceed 256 KiB.

Submissions limit

You can submit at most 50 solutions for each problem.

You can submit a solution to each task at most once per 30 seconds. This restriction does not apply in the last 15 minutes of the contest round.

Scoring

Each problem consists of several subtasks. The subtask score is awarded if all tests in the subtask are passed.

The number of points scored for the problem is the total number of points scored on each of its subtasks. The score for the subtask is the maximum number of points earned for this subtask among all the solutions submitted.

Feedback

To get feedback for your solution, go to “Runs” tab in PCMS2 Web Client and use “View Feedback” link. In each problem of the contest you will see the score for each subtask, or the verdict for the first failed test.

Scoreboard

The scoreboard will not be available to contestants during the contest round. The official results will be published after the round has concluded, in accordance with the procedures established by the organizers.



Problem A. Romantic date

Time limit: 1 second

Rinat and Laysan are two snow leopards. Rinat has long wanted to ask Laysan out on a date. However, finding a suitable time for them to meet turns out to be not so easy: both of them have very unusual schedules.

Starting at time 0, Rinat is free for the first a seconds, busy for the next a seconds, then free again for a seconds, and so on.

Laysan's schedule is the opposite: she is busy for the first b seconds, free for the next b seconds, then busy again for b seconds, and so on.

To ask Laysan out on a date, Rinat needs to find a time when they are both free for at least one full second.

Rinat is willing to wait until time T .

Find the minimum integer t ($0 \leq t \leq T$) such that Rinat and Laysan are both free throughout the entire interval $[t, t + 1)$, that is, from time t inclusive to time $t + 1$ exclusive.

If no such time exists, output -1 .

Input

The first line contains three integers a , b , and T ($1 \leq a, b \leq 10^{18}$, $0 \leq T \leq 10^{18}$).

Output

Output the minimum integer t such that during the entire time interval $[t, t + 1)$ Rinat and Laysan are both free.

If no such t exists, output -1 .

Scoring

The tests for this problem consist of seven groups. Points for each group are awarded only if all tests of the group, all tests required for it, and the tests from the statement are passed.

Group	Points	Additional constraints	Required groups
1	15	$T \leq 10^6$	–
2	15	$a, b \leq 10^3$	–
3	10	$b < a$	–
4	15	$b < 2a$	3
5	15	$b < 1\,000\,001 \cdot a$	2–4
6	10	b is divisible by a	–
7	20	No additional constraints	1–6



Examples

standard input	standard output
5 3 10	3
3 5 10	6
3 8 5	-1

Note

In the first example, Rinat is free during the first 5 seconds, that is, on the interval $[0, 5)$, then busy on $[5, 10)$. Laysan, on the contrary, is busy on $[0, 3)$ and free on $[3, 6)$. Therefore, the first time they are both free is on the interval $[3, 5)$, and the minimum suitable moment is 3.

In the second example, Laysan is free on $[5, 10)$. At moment 5, Rinat is still busy, since his busy interval is $[3, 6)$. Starting from moment 6, he is free again, so the answer is 6.

In the third example, there is no suitable moment before $T = 5$: Laysan becomes free only at moment 8. Therefore, the answer is -1 .



Problem B. Second Attempt

Time limit: 2 seconds

Despite all his efforts, the Laysan still refused the Rinat. But Rinat is not used to giving up!

To impress Laysan, Rinat decided to give her a bouquet of flowers. He bought n flowers and, before arranging the bouquet, laid them out in a row. Each flower is of some type. For each flower, Rinat wrote down a number a_i — the type of the i -th flower in the row.

However, Rinat learned that Laysan likes variety and wants the bouquet to contain no two flowers of the same type. In other words, the types of all remaining flowers must be pairwise distinct.

Rinat does not want to change the order of the flowers. He can only remove one contiguous fragment of the row: choose two indices l and r ($1 \leq l \leq r \leq n$) and remove the flowers at positions $l, l+1, \dots, r$.

If all flowers already have pairwise distinct types, Rinat may remove nothing. In this case, the number of removed flowers is considered to be 0.

Help Rinat determine the minimum number of consecutive flowers that must be removed so that all remaining flowers have pairwise distinct types.

Input

Each test consists of several test cases. The first line contains one integer t ($1 \leq t \leq 10^5$) — the number of test cases. The description of the test cases follows.

The first line of each test case contains one integer n ($1 \leq n \leq 10^6$) — the number of flowers in the row.

The second line of each test case contains n integers $a_1, a_2, \dots, a_n$ ($0 \leq a_i \leq 10^9$) — the types of the flowers.

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

Output

For each test case, output one integer — the minimum number of consecutive flowers that need to be removed so that the species numbers of all remaining flowers are pairwise distinct.



Scoring

The tests for this problem consist of nine groups. Points for each group are awarded only if all tests in the group, all tests required for it, and the tests from the statement are passed.

Group	Points	Additional constraints	Required groups
1	10	$n \leq 100$	–
2	5	$n \leq 10^3$	1
3	10	$n \leq 5 \cdot 10^3$	1, 2
4	10	$n \leq 10^4$	1–3
5	15	Each number appears at most twice, $n \leq 10^5$	–
6	15	$n \leq 5 \cdot 10^4$	1–4
7	10	$n \leq 10^5$	1–6
8	10	$n \leq 5 \cdot 10^5$	1–7
9	15	No additional constraints	1–8

Example

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

Note

In the first test case, one can remove the fragment $[2, 3]$.

In the second test case, nothing needs to be removed.

In the third test case, one can remove the fragment $[1, 6]$.

In the fourth test case, one can remove the fragment $[3, 4]$.

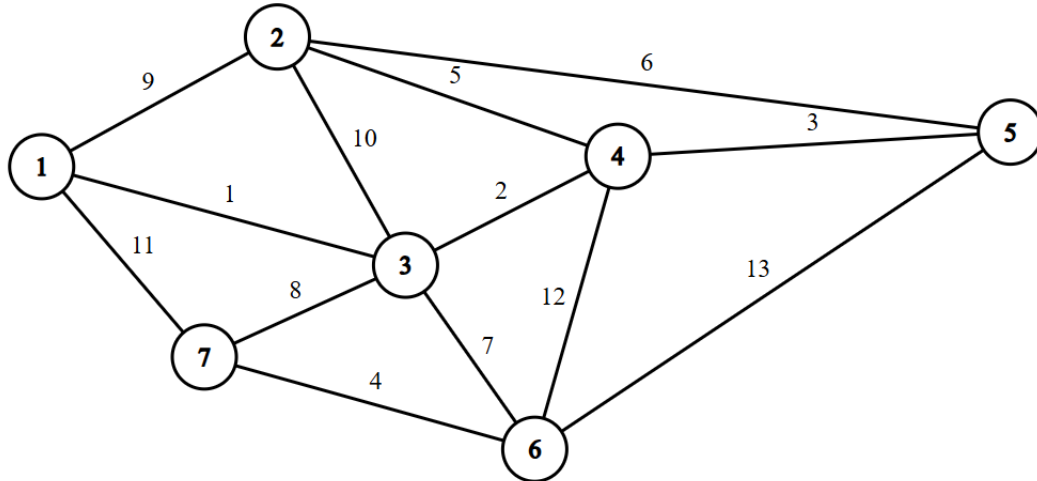
In the fifth test case, one can remove the fragment $[2, 2]$.



Problem C. Forest trails

Time limit: 1 second

You are visiting a local forester and notice a map of the nearby forest.



The map shows intersections numbered from 1 to 7 and forest trails numbered from 1 to 13.

The local foresters have come up with their own algorithm for exploring the forest and written it as a function called **FOREST**:

```
FOREST(C, D)
  dist[C] = D
  for all trails of the form (C, X) or (X, C)
    if dist[X] not initialized
      FOREST(X, dist[C] + 1)
    else
      dist[C] = min(dist[C], dist[X] + 1)
```

In this function:

- **dist** — an array of distances whose values are initially undefined;
- **min** — the minimum function;
- **for all trails** — a loop through all trails connected to the current intersection;
- trails are **always processed in increasing order** of their numbers.

A detailed example of how this function works on the given map can be found in the **Note** section.

The foresters often use this algorithm to explore the forest without paying attention to the calculated distances.

However, some foresters also use it to find the shortest distance from the main entrance (intersection 1) to every other intersection.

Here, the distance is the minimum number of trails needed to reach an intersection, since every trail has length 1.

To calculate these distances, they call the function **FOREST(1, 0)**.



After looking at this function, you and the local forester realize that it does not always calculate the shortest distances correctly.

To demonstrate this, you decide to construct t example maps.

The i -th map must contain n_i intersections, and the function `FOREST(1, 0)` must calculate the correct shortest distance for **exactly** k_i of them.

For some maps, the number of trails is also fixed and must be equal to m_i .

More details about the requirements for these maps can be found in the **Output** section.

Input

The first line contains one positive integer t ($1 \leq t \leq 500$) — the number of test cases.

Each of the next t lines contains three integers n_i , m_i , and k_i — the requirements for the i -th map ($1 \leq n_i \leq 300\,000$; $0 \leq k_i \leq n_i$; $-1 \leq m_i \leq \min\left(\frac{n_i(n_i - 1)}{2}, 300\,000\right)$).

If $m_i = -1$, the number of trails is not fixed.

It is guaranteed that the sum of all w_i does not exceed 300 000, where $w_i = m_i$ if $m_i \neq -1$, and otherwise $w_i = \min\left(\frac{n_i(n_i - 1)}{2}, 300\,000\right)$.

Output

For each test case, output your answer in the following format:

- If a valid map exists, print `Ok` m'_i on the first line, followed by the descriptions of m'_i trails.
If $m_i \neq -1$, then $m'_i = m_i$. Otherwise, $0 \leq m'_i \leq \min\left(\frac{n_i(n_i - 1)}{2}, 300\,000\right)$.
- If no valid map exists, print `Impossible`.

If a valid map exists, each of the next m'_i lines must contain two integers u_j and v_j — the numbers of the intersections connected by the j -th trail.

It must be possible to travel between any two intersections using the trails.

Each pair of intersections can be connected by at most one trail. Each trail must connect two different intersections and can be used in both directions.

The maps do not have to be realistic in any other way.

For example, it is allowed to construct a map that cannot be drawn on a plane without trails crossing each other.

Scoring

The tests for this problem are divided into thirteen groups.

Points for a group are awarded only if all tests in that group and all tests in its required groups are passed.



Group	Points	Additional constraints	Required groups
0	0	Sample tests	–
1	8	$n_i \leq 8$ and $m_i = -1$	–
2	12	$n_i \leq 5000$ and $m_i = -1$	1
3	8	$0 \leq m_i < n_i - 1$	–
4	13	$k_i \leq 3$	–
5	6	$k_i = n_i$	–
6	10	$n_i \leq 8$	0, 1
7	10	$n_i \leq 100$	0, 1, 6
8	10	$n_i \leq 5000$ and $m_i = n_i$	–
9	4	$n_i \leq 5000$ and $m_i \geq n_i + k_i$	0
10	4	$n_i \leq 5000$	0–2, 6–9

Groups 11–13 are evaluated using **offline testing**. This means that the results of your solution on these groups will only be available after the contest ends.

Group	Points	Additional constraints	Required groups
11	5	$m_i = -1$	1, 2
12	5	$m_i = n_i$	8
13	5	–	0–12

Example

standard input	standard output
1 7 13 6	Ok 13 1 3 4 3 4 5 6 7 2 4 2 5 3 6 7 3 1 2 2 3 1 7 4 6 5 6



Note

```
FOREST(1, 0)
  dist[1] = 0
  trail 1 (1, 3): call FOREST(3, 1)
  FOREST(3, 1)
    dist[3] = 1
    trail 1 (1, 3): dist[3] is not changed
    trail 2 (4, 3): call FOREST(4, 2)
    FOREST(4, 2)
      dist[4] = 2
      trail 2 (4, 3): dist[4] is not changed
      trail 3 (4, 5): call FOREST(5, 3)
      FOREST(5, 3)
        dist[5] = 3
        trail 3 (4, 5): dist[5] is not changed
        trail 6 (2, 5): call FOREST(2, 4)
        FOREST(2, 4)
          dist[2] = 4
          trail 5 (2, 4): dist[2] = min(dist[2], dist[4] + 1) = min(4, 2 + 1) = 3
          trail 6 (2, 5): dist[2] is not changed
          trail 9 (1, 2): dist[2] = min(dist[2], dist[1] + 1) = min(3, 0 + 1) = 1
          trail 10 (2, 3): dist[2] is not changed
          trail 13 (5, 6): call FOREST(6, 4)
          FOREST(6, 4)
            dist[6] = 4
            trail 4 (6, 7): call FOREST(7, 5)
            FOREST(7, 5)
              dist[7] = 5
              trail 4 (6, 7): dist[7] is not changed
              trail 8 (7, 3): dist[7] = min(dist[7], dist[3] + 1) = min(5, 1 + 1) = 2
              trail 11 (1, 7): dist[7] = min(dist[7], dist[1] + 1) = min(2, 0 + 1) = 1
              trail 7 (3, 6): dist[6] = min(dist[6], dist[3] + 1) = min(4, 1 + 1) = 2
              trail 12 (4, 6): dist[6] is not changed
              trail 13 (5, 6): dist[6] is not changed
            trail 5 (2, 4): dist[4] is not changed
            trail 12 (4, 6): dist[4] is not changed
            trail 7 (3, 6): dist[3] is not changed
            trail 8 (7, 3): dist[3] is not changed
            trail 10 (2, 3): dist[3] is not changed
            trail 9 (1, 2): dist[1] is not changed
            trail 11 (1, 7): dist[1] is not changed
```

So in summary:

Crossroad	1	2	3	4	5	6	7
Distance	0	1	1	2	3	2	1

The only incorrect distance is the distance to the crossroad 5.



Problem D. F1

Time limit: 2 seconds

The cat Miston found a fragment of an old Soviet newspaper about an F1 race in the attic. In the margins, there were written $2m$ numbers, two numbers per line.

From other notes, Miston understood what these numbers meant: someone had professionally, in chronological order, transcribed all overtakes. That is, the numbers f_i and s_i in the i -th line meant that driver number f_i overtook driver number s_i . Moreover, only those overtakes that changed the results table were recorded (in other words, overtakes by one or more laps were not recorded).

However, there is no description of the starting grid anywhere — that is, the order in which the drivers line up before the race starts. Only the total number of drivers was found — there were n of them.

After leaving the dusty attic, the kittens bombarded the tired cat with questions of the form “could driver x_i have ended up in place p_i ?”. He will have to answer them.

Input

The first line contains three integers n , m , and q ($0 \leq n, m, q \leq 3 \cdot 10^5$; $n \neq 0$; $q \neq 0$) — the number of drivers, overtakes, and questions.

The i -th of the next m lines contains two integers f_i and s_i ($1 \leq f_i, s_i \leq n$; $f_i \neq s_i$) — the numbers from the i -th line in the margins.

It is guaranteed that the overtaking data is correct, i.e. there exists at least one starting grid such that all recorded overtakes are possible in the specified order.

The i -th of the next q lines contains two integers x_i and p_i ($1 \leq x_i, p_i \leq n$) — the driver number and the place from the i -th question.

Output

Print q lines, where the i -th line should contain **Yes** if the answer to the i -th question is positive, and **No** otherwise.

Assume that all drivers finished the race, nobody was disqualified, and nobody received any delayed time penalties.



Scoring

The tests for this problem consist of nine groups. Points for each group are awarded only if all tests in the group and all tests required for it are passed.

Group	Points	Additional constraints	Required groups
0	0	Tests from the statement	–
1	5	$m = 0$	–
2	17	$n, m \leq 10$	0
3	8	The leader finished last*	–
4	7	$p_i = 1$	–
5	11	$p_i \leq 3$	4
6	12	$n, m \leq 1000$	0, 2
7	11	$n \leq 20\,000$	0, 2, 6
8	7	$q \leq 10$	–
9	22	–	0–8

* The driver in pole position — that is, the one starting first — finished last.

Example

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

Note

Consider all starting grids suitable for the sample and the corresponding race results and answers to the questions.

Starting grid	Finish order	1	2	3	4	5	6
1, 5, 3, 2, 4, 6	5, 1, 2, 3, 4, 6	No	Yes	No	No	No	No
1, 5, 6, 3, 2, 4	5, 1, 6, 2, 3, 4	No	Yes	Yes	No	No	No
3, 2, 4, 1, 5, 6	2, 3, 4, 5, 1, 6	No	No	No	No	No	No
3, 2, 4, 6, 1, 5	2, 3, 4, 6, 5, 1	No	No	No	No	Yes	No
6, 1, 5, 3, 2, 4	6, 5, 1, 2, 3, 4	No	No	Yes	No	No	Yes
6, 3, 2, 4, 1, 5	6, 2, 3, 4, 5, 1	No	No	No	No	Yes	Yes



Problem E. Being a Cheater

Time limit: 5 seconds

Consider a quiz show with the following rules: there are n participants and k question topics numbered from 1 to k . Each player has a strong topic a_i and a weak topic b_i . Of course, $a_i \neq b_i$. A player always answers correctly on a question from their strong topic, always answers incorrectly on a question from their weak topic, and on a question from any of the middle topics they answer correctly on the first $L - 1$ questions, and starting from the L -th question they answer incorrectly.

However, things are not so simple. Among the players, there is one cheater! The cheater always answers all questions correctly, ignoring the previous rules.

As the host, you want to identify the cheater using the minimum number of questions. For each question, you choose a topic, and this question is asked to all participants. After that, you learn who answered the question correctly and who answered incorrectly. Then you may ask the next question. You know that exactly one of the players is the cheater.

Let the minimum number of questions that must be asked to guarantee identifying the cheater for the game with participants numbered $l, l + 1, \dots, r$ be $f(l, r)$.

Compute $\sum_{l=1}^n \sum_{r=l}^n f(l, r)$. In other words, consider all possible contiguous segments of players from l to r , where $1 \leq l \leq r \leq n$.

For each such segment, determine the minimum number of questions needed to identify the cheater with certainty, assuming that only players numbered from l to r participate in the game.

Your task is to find the sum of these values over all such segments.

Input

The first line contains three integers n , k , and L ($2 \leq n \leq 200\,000$, $2 \leq k \leq n$, $2 \leq L \leq n$).

The second line contains n integers $a_1, a_2, \dots, a_n$ ($1 \leq a_i \leq k$).

The third line contains n integers $b_1, b_2, \dots, b_n$ ($1 \leq b_i \leq k$).

For each i , $a_i \neq b_i$.

Output

Print a single integer — the sum of $f(l, r)$ over all pairs of indices l and r satisfying $1 \leq l \leq r \leq n$.

Scoring

The tests of the problem are divided into 16 subgroups. Points for a subgroup are awarded only if all its tests and all tests of the required subgroups listed in the table are passed. The table gives additional constraints; the remaining constraints are the same as the general ones.



Subgroup	Points	Additional constraints			Required subgroups
		n	k	Other conditions	
0	0	Examples from the statement			–
1	2	–	–	$b_i = 1$ for all i	–
2	7	$n \leq 7$	–	–	–
3	9	$n \leq 70$	–	–	0, 2
4	5	$n \leq 3000$	$k \leq 8$	–	0, 2
5	6	$n \leq 3000$	$k \leq 30$	–	0, 2, 4
6	7	$n \leq 3000$	–	–	0, 2–5
7	4	–	$k = 2$	–	–
8	5	–	$k = 3$	$L = 2$	–
9	11	–	–	$L \leq 20$	0, 2, 8
10	5	–	–	$L \leq 200$	0, 2, 3, 8, 9
11	6	–	–	$L = n$	–
12	16	–	$k = n$	–	–
13	7	–	–	$L = k - 1$	8
14	5	$n \leq 100\,000$	–	–	0, 2–6
15	4	$n \leq 150\,000$	–	–	0, 2–6, 14
16	1	–	–	–	0–15



Examples

standard input	standard output
3 3 3 2 3 2 1 1 1	3
4 2 3 1 2 1 2 2 1 2 1	7
6 3 2 1 2 1 2 1 2 2 1 3 3 2 1	23
4 4 2 2 3 4 1 1 2 3 4	9
5 3 5 2 3 1 3 1 1 1 2 2 3	13
20 5 2 1 2 3 4 5 1 2 3 4 5 1 2 3 4 5 1 2 3 4 5 2 3 4 5 1 3 4 5 1 2 4 5 1 2 3 5 1 2 3 4	448

Note

If there is only one person in the game, that person is the cheater, so no questions are needed.

In the first example, the weak topic of each player is topic 1. If there are at least two participants, the host asks one question on this topic. All honest players answer incorrectly, and the cheater answers correctly, so the host immediately identifies them. Three games with one participant, two games with two participants, and one game with all three participants are considered. Therefore, the answer is $3 \cdot 0 + 2 \cdot 1 + 1 \cdot 1 = 3$.

In the second example, players 1 and 3 have strong topic 1 and weak topic 2, while players 2 and 4 have the opposite. For the game with four participants, the host asks one question on topic 1, and then one on topic 2. Each honest player will make a mistake on the question from their weak topic. Only the cheater will answer both questions correctly. One question is not enough: if the cheater is one of the two players for whom the chosen topic is strong, both of these players will answer correctly and it will be impossible to distinguish between them. Thus, $f(1, 4) = 2$.

In the game with two participants, it is enough to ask the weak topic of one of them. If they answer correctly, they are the cheater; if incorrectly, the cheater is the second player. In the game with three participants, we choose a topic that is weak for two of them. If one of those two answers correctly, they are the cheater; if both answer incorrectly, the cheater is the third player. Therefore, the three games with two participants and the two games with three participants have complexity 1. The total sum is $3 \cdot 1 + 2 \cdot 1 + 2 = 7$.

In the sixth example, consider the game with only players 1, 2, ..., 10. The host first asks two questions on topic 1. For players 1 and 6, this topic is strong, so both of them answer correctly. For each of the other eight players, it is weak or middle: if such a player is honest, they will make at least one mistake. If one of these eight players answers both questions correctly, then they are the cheater. Otherwise, the cheater is one of players 1 and 6.

To distinguish between these two players, the host asks one more question on topic 2. For player 1, this topic is weak. If they answer correctly, they are the cheater. If they answer incorrectly, then the cheater is player 6, found by elimination. For player 6, topic 2 is middle, and this is the first question on it, so



they will answer correctly even if they are honest. In the worst case, 3 questions are needed: first $L = 2$ questions on one topic, then one more on another topic.

Two questions are not enough. If the same topic is asked twice, both players for whom it is strong will answer correctly. If the cheater is one of them, it will be impossible to distinguish them from the other. If two different topics are asked once each, honest players can make mistakes only on their weak topics. Each topic is weak for exactly two of the first ten players, so neither of the chosen topics will be weak for at least six players. All of them will answer both questions correctly. If the cheater is one of them, it will be impossible to determine which one. Therefore, $f(1, 10) = 3$.

In the same example, consider the game with all 20 participants. The host asks two questions on topic 1 and two questions on topic 2, for a total of 4 questions. Each player has only one strong topic, so at least one of the chosen topics is weak or middle for them. If the player is honest, on a weak topic they will make a mistake immediately, and on a middle topic they will make a mistake on the second question, since $L = 2$. Thus, every honest player will make at least one mistake, while the cheater will answer all four questions correctly.

The minimum number of questions for this game is 4, so $f(1, 20) = 4$.