A. Sports Betting

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

The boarding process for various flights can occur in different ways: either by **bus** or through a **telescopic jet bridge**. Every day, exactly one flight is made from St. Petersburg to Minsk, and Vadim decided to demonstrate to the students that he always knows in advance how the boarding will take place.

Vadim made a bet with n students, and with the i-th student, he made a bet on day a_i. Vadim wins the bet if he correctly predicts the boarding process on both day $a_i + 1$ and day $a_i + 2$.

Although Vadim does not know in advance how the boarding will occur, he really wants to win the bet **at least** with one student and convince him of his predictive abilities. Check if there exists a strategy for Vadim that allows him to **guarantee** success.

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 a single integer n ($1 \le n \le 10^5$) — the number of students Vadim made bets with.

The second line of each test case contains n integers $a_1, \ldots, a_n$ ($1 \le a_i \le 10^9$) — the days on which Vadim made bets with the students.

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

Output

For each test case, output "`Yes`" (without quotes) if Vadim can **guarantee** convincing at least one student, and "`No`" otherwise.

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.

Standard Input	Standard Output
5 4 1 1 1 1 3 2 2 2 5 2 4 3 2 4 8 6 3 1 1 5 1 2 6 1 1000000000	Yes No Yes No No

Note

In the first test case, Vadim needs to make at least one correct prediction about the boarding process on the second and third days. There are a total of 4 possible boarding scenarios for these days, so Vadim can give all 4 students different predictions and guarantee that at least one of them will be correct.

In the second test case, Vadim only made bets with three students and cannot guarantee that he will provide at least one of them with a correct prediction.

,

B. Baggage Claim

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

Every airport has a baggage claim area, and Balbesovo Airport is no exception. At some point, one of the administrators at Sheremetyevo came up with an unusual idea: to change the traditional shape of the baggage claim conveyor from a carousel to a more complex form.

Suppose that the baggage claim area is represented as a rectangular grid of size $n \times m$. The administration proposed that the path of the conveyor should pass through the cells $p_1, p_2, \ldots, p_{2k+1}$, where $p_i = (x_i, y_i)$.

For each cell p_i and the next cell p_{i+1} (where $1 \le i \le 2k$), these cells must share a common side. Additionally, the path must be simple, meaning that for no pair of indices $i \ne j$ should the cells p_i and p_j coincide.

Unfortunately, the route plan was accidentally spoiled by spilled coffee, and only the cells with odd indices of the path were preserved: $p_1, p_3, p_5, \ldots, p_{2k+1}$. Your task is to determine the number of ways to restore the original complete path $p_1, p_2, \ldots, p_{2k+1}$ given these $k + 1$ cells.

Since the answer can be very large, output it modulo $10^9 + 7$.

Input

Each test contains multiple test cases. The first line contains the number of test cases t ($1 \le t \le 3 \cdot 10^4$). The description of the test cases follows.

The first line of each test case contains three integers n, m, and k ($1 \le n, m \le 1000$, $n \cdot m \ge 3$, $1 \le k \le \left\lfloor \frac{1}{2}(nm - 1) \right\rfloor$) — the dimensions of the grid and a parameter defining the length of the path.

Next, there are $k + 1$ lines, the i-th of which contains two integers x_{2i-1} and y_{2i-1} ($1 \le x_{2i-1} \le n$, $1 \le y_{2i-1} \le m$) — the coordinates of the cell p_{2i-1} that lies on the path.

It is guaranteed that all pairs (x_{2i-1}, y_{2i-1}) are distinct.

It is guaranteed that the sum $n \cdot m$ over all test cases does not exceed 10^6.

Output

For each test case, output a single integer — the number of ways to restore the original complete path modulo $10^9 + 7$.

Standard Input	Standard Output
5 2 4 2 1 1 2 2 2 4 1 4 1 1 1 1 4 5 5 11	2 0 2 5 1

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

Note

In the first test case, there are two possible paths:

- $(1,1) \to (2,1) \to (2,2) \to (2,3) \to (2,4)$
- $(1,1) \to (1,2) \to (2,2) \to (2,3) \to (2,4)$

In the second test case, there is no suitable path, as the cells $(1,1)$ and $(1,4)$ do not have a common neighboring cell.

C. Bermuda Triangle

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

The Bermuda Triangle — a mysterious area in the Atlantic Ocean where, according to rumors, ships and airplanes disappear without a trace. Some blame magnetic anomalies, others — portals to other worlds, but the truth remains hidden in a fog of mysteries.

A regular passenger flight 814 was traveling from Miami to Nassau on a clear sunny day. Nothing foreshadowed trouble until the plane entered a zone of strange flickering fog. Radio communication was interrupted, the instruments spun wildly, and flashes of unearthly light flickered outside the windows.

For simplicity, we will assume that the Bermuda Triangle and the airplane are on a plane, and the vertices of the triangle have coordinates $(0,0)$, $(0,n)$, and $(n,0)$. Initially, the airplane is located at the point (x,y) **strictly inside** the Bermuda Triangle and is moving with a velocity vector (v_x, v_y). All instruments have failed, so the crew cannot control the airplane.

The airplane can escape from the triangle if it ever reaches exactly one of the vertices of the triangle. However, if at any moment (possibly non-integer) the airplane hits the boundary of the triangle (but not at a vertex), its velocity vector is immediately reflected relative to that side†, and the airplane continues to move in the new direction.

Determine whether the airplane can ever escape from the Bermuda Triangle (i.e., reach exactly one of its vertices). If this is possible, also calculate how many times before that moment the airplane will hit the boundary of the triangle (each touch of the boundary, even at the same point, counts; crossing a vertex does not count).

† Reflection occurs according to the usual laws of physics. The angle of incidence equals the angle of reflection.

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 only line of each test case contains five integers n, x, y, v_x, and v_y ($3 \le n \le 10^9$, $1 \le x, y$, $x + y < n$, $1 \le v_x, v_y \le 10^9$) — numbers describing the vertices of the triangle, the initial coordinates of the airplane, and the initial velocity vector.

Output

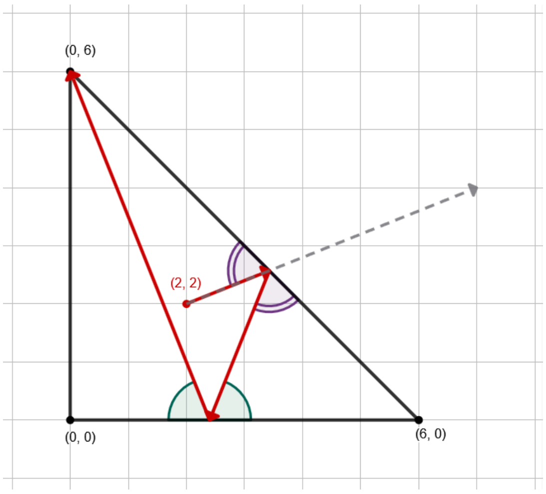
For each test case, output a single integer — the number of times the airplane will hit the boundary of the triangle before it escapes. If the airplane never escapes from the triangle, output -1.

Standard Input	Standard Output
6 6 2 2 5 2 6 2 2 20 8 4 1 2 1 1 4 1 1 1 2	2 2 -1 -1

4 1 1 2 1	-1
6 2 3 2 3	5

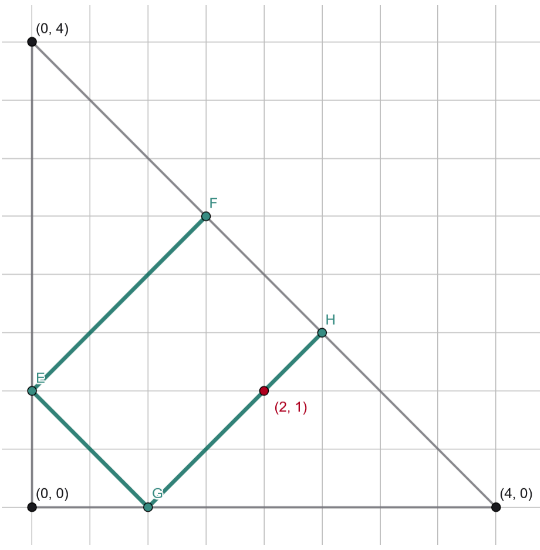
Note

An illustration for the first test case is provided below:



In the second test case, the answer is the same as in the first, as all initial data, except for the speed, are the same, but the airplanes are initially moving in the same direction.

In the third test case, the answer is -1, as the airplane will move exclusively along the segments marked in green. An illustration is provided below:



,

D. Homework

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

Some teachers work at the educational center "Sirius" while simultaneously studying at the university. In this case, the trip does not exempt them from completing their homework, so they do their homework right on the plane. Artem is one of those teachers, and he was assigned the following homework at the university.

With an arbitrary string a of **even** length m, he can perform the following operation. Artem splits the string a into two halves x and y of equal length, after which he performs **exactly one** of three actions:

- For each $i \in \left\{1, 2, \ldots, \frac{m}{2}\right\}$ assign $x_i = (x_i + y_i) \bmod 2$;
- For each $i \in \left\{1, 2, \ldots, \frac{m}{2}\right\}$ assign $y_i = (x_i + y_i) \bmod 2$;
- Perform an arbitrary number of operations (the same operations defined above, applied recursively) on the strings x and y, independently of each other. Note that in this case, the strings x and y must be of even length.

After that, the string a is replaced by the strings x and y, concatenated in the same order.
Unfortunately, Artem fell asleep on the plane, so you will have to complete his homework. Artem has two binary strings s and t of length n, each consisting of n characters `0` or `1`. Determine whether it is possible to make string s equal to string t with **an arbitrary** number of operations.

Input

Each test contains multiple test cases. The first line contains the number of test cases t ($1 \le t \le 10^5$). The description of the test cases follows.

The first line of each test case contains a single integer n ($1 \le n \le 10^6$) — the length of the strings s and t.

The second line of each test case contains the string s of length n, consisting only of characters `0` and `1`.

The third line of each test case contains the string t of length n, consisting only of characters `0` and `1`.

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

Output

For each test case, output "`Yes`" (without quotes) if it is possible to make string s equal to string t, and "`No`" otherwise.

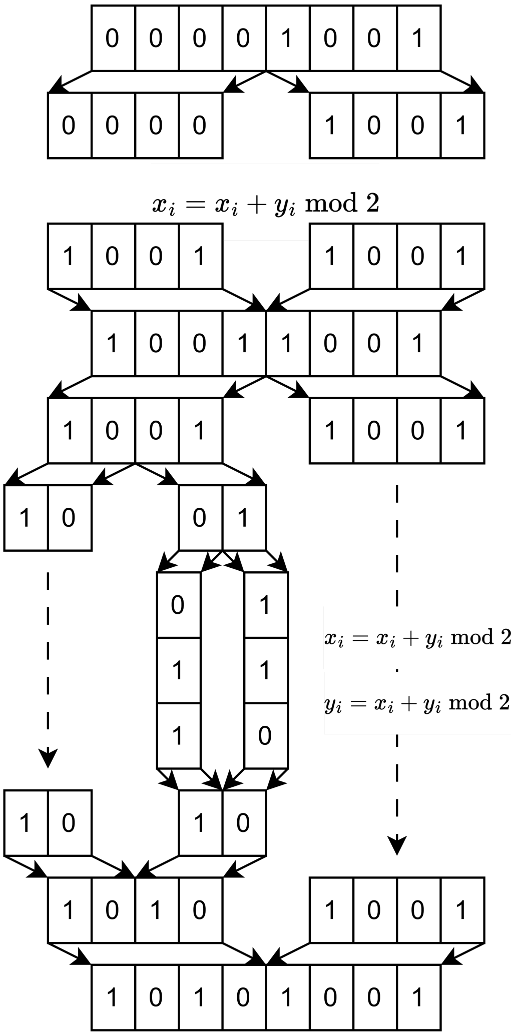
You can output each letter in any case (lowercase or uppercase). For example, the strings "`yEs`", "`yes`", "`Yes`", and "YES" will be accepted as a positive answer.

Standard Input	Standard Output
3 8 00001001 10101001 8 00000000 00001001	Yes No Yes

6 010110 100010	

Note

In the first test case, the string `00001001` can be transformed into the string `10101001` in two operations. The sequence of actions is illustrated in the figure below:



In the second test case, the string `00000000` cannot be transformed into any string other than `00000000`, as no non-zero elements can be formed during any operation.

,

E. Clearing the Snowdrift

```
Input file:     standard input
Output file:    standard output
Time limit:     2 seconds
Memory limit:   1024 megabytes
```

Boy Vasya loves to travel very much. In particular, flying in airplanes brings him extraordinary pleasure. He was about to fly to another city, but the runway was heavily covered with snow and needed to be cleared.

The runway can be represented as n consecutive sections numbered from 1 to n. The snowstorm was quite strong, but it has already stopped, so Vasya managed to calculate that the i-th section is covered with a_i meters of snow. For such situations, the airport has a snowplow that works in a rather unusual way. In one minute, the snowplow can do the following:

- Choose a consecutive segment of length no more than d and remove one meter of snow from the most snow-covered sections.
  Formally, one can choose $1 \le l \le r \le n$ ($r - l + 1 \le d$). After that, $c = \max\{a_l, a_{l+1}, \ldots, a_r\}$ is calculated, and if $c > 0$, then for all $i\colon l \le i \le r$ such that $a_i = c$, the value of a_i is decreased by one.

Vasya has been preparing for the flight for a long time and wants to understand how much time he has left to wait until all sections are completely cleared of snow. In other words, it is required to calculate the minimum number of minutes that the snowplow will need to achieve $a_i = 0$ for all i from 1 to n.

Input

Each test contains multiple test cases. The first line contains the number of test cases t ($1 \le t \le 2 \cdot 10^5$). The description of the test cases follows.

The first line of each test case contains two integers n and d ($1 \le n \le 5 \cdot 10^5, 1 \le d \le n$) — the number of sections on the runway and the maximum length of the segment that the snowplow can choose.

The second line of each test case contains n integers $a_1, a_2, \ldots, a_n$ ($1 \le a_i \le 10^9$), where a_i is the number of meters of snow on the i-th section.

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

Output

For each test case, output a single integer — the minimum number of minutes required for the snowplow to achieve $a_i = 0$ for all i from 1 to n.

Standard Input	Standard Output
2 5 2 1 5 2 1 2 3 1 1000000000 1000000000 1000000000	8 3000000000

Note

In the first test case, there is an optimal sequence of operations. First, select the segment $[2, 3]$ four times. After three operations, a_2 will turn into 2, and the array a will look like $[1, 2, 2, 1, 2]$. After the fourth operation, the array a will become $[1, 1, 1, 1, 2]$. Next, the array can be transformed into zeros by selecting the segments $[1, 2]$, $[3, 3]$, $[5, 5]$, and $[4, 5]$ (in that exact order).

In the second test case, $d = 1$, which means that each section is cleared independently of the others, and the answer is equal to the sum of all a_i.

,

F. Lost Luggage

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

As is known, the airline "Trouble" often loses luggage, and concerned journalists decided to calculate the maximum number of luggage pieces that may not return to travelers.

The airline "Trouble" operates flights between n airports, numbered from 1 to n. The journalists' experiment will last for m days. It is known that at midnight before the first day of the experiment, there were s_j lost pieces of luggage in the j-th airport. On the i-th day, the following occurs:

- **In the morning**, $2n$ flights take off simultaneously, including n flights of the *first type* and n flights of the *second type*.
  - The j-th flight of the first type flies from airport j to airport $(((j-2) \bmod n)+1)$ (the previous airport, with the first airport being the last), and it can carry no more than $a_{i,j}$ lost pieces of luggage.
  - The j-th flight of the second type flies from airport j to airport $((j \bmod n)+1)$ (the next airport, with the last airport being the first), and it can carry no more than $c_{i,j}$ lost pieces of luggage.
- **In the afternoon**, a check of lost luggage is conducted at the airports. If after the flights have departed on that day, there are x pieces of luggage remaining in the j-th airport and $x \geq b_{i,j}$, then at least $x - b_{i,j}$ pieces of luggage are found, and they **cease to be lost**.
- **In the evening**, all $2n$ flights conclude, and the lost luggage transported that day arrives at the corresponding airports.

For each k from 1 to m, the journalists want to know the maximum number of lost pieces of luggage that may be **unfound** during the checks over the first k days. Note that for each k, these values are calculated independently.

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.

The first line of each test case contains two integers n and m ($3 \leq n \leq 12, 1 \leq m \leq 2000$) — the number of airports and the number of days of the experiment.

The second line of each test case contains n integers $s_1, s_2, \ldots, s_n$ ($0 \leq s_i \leq 10^8$) — the initial number of lost pieces of luggage in each airport.

Next, the descriptions for each of the m days follow in order.

The first line of the description of the i-th day contains n integers $a_{i,1}, a_{i,2}, \ldots, a_{i,n}$ ($0 \leq a_{i,j} \leq 10^8$) — the maximum number of lost pieces of luggage that can be transported to the previous airport for each airport.

The second line of the description of the i-th day contains n integers $b_{i,1}, \ldots, b_{i,n}$ ($0 \leq b_{i,j} \leq 10^8$) — the minimum number of lost pieces of luggage that will be found on the i-th day in each airport.

The third line of the description of the i-th day contains n integers $c_{i,1}, \ldots, c_{i,n}$ ($0 \leq c_{i,j} \leq 10^8$) — the maximum number of lost pieces of luggage that can be transported to the next airport for each airport.

It is guaranteed that the sum of m over all test cases does not exceed 2000.

Output

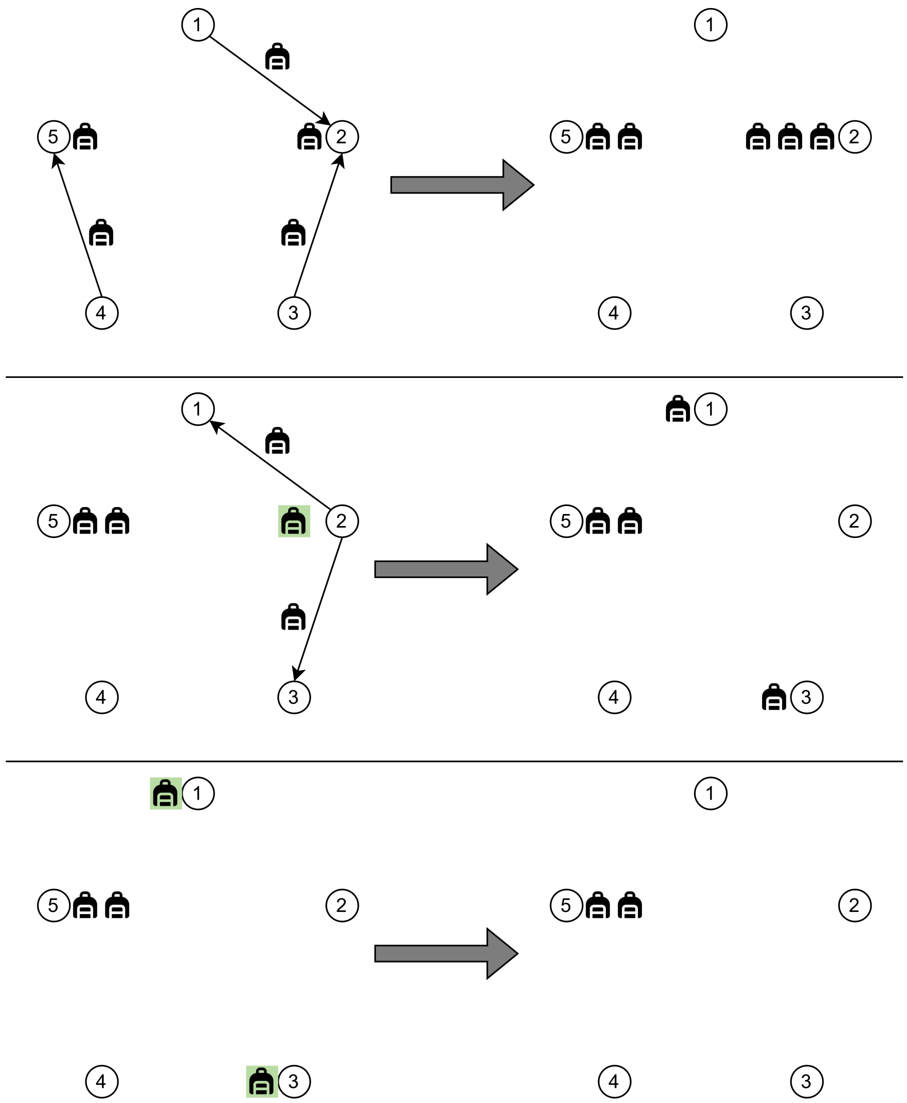
For each test case, output m integers — the maximum number of unfound pieces of luggage for each number of days from 1 to m.

Standard Input	Standard Output
2 5 3 1 1 1 1 1 0 0 1 0 0 0 1 0 0 1 1 0 0 1 0 0 1 0 0 0 9 0 9 9 9 0 1 0 0 0 0 0 0 0 0 9 0 9 0 0 0 0 0 0 0 3 1 0 100000000 5 0 100000000 5 0 100000000 5 0 100000000 5	5 4 2 100000005

Note

In the first test case:

- On the first day, all 5 pieces of luggage may not be found, as the lost luggage can be sent on flights from each airport.
- In the morning of the second day, there may be no more than 3 pieces of luggage in the 2-nd airport, no more than 2 pieces in the 5-th airport, and no luggage in the other airports. All luggage from the 5-th airport may remain there. In the 2-nd airport, no more than 2 pieces of luggage can be sent on flights to neighboring airports. Thus, at least 1 piece of luggage will be found.
- By the end of the third day, lost luggage may only be in the 1-st and 2-nd airports. There can be no more than one piece in each, meaning that at most 2 pieces of luggage will remain unfound in total.



The found luggage is marked in green.

In the second test case, all pieces of luggage may remain in their original airports, and the inspection won't find any lost suitcases. Therefore, the answer is $10^9 + 5$.