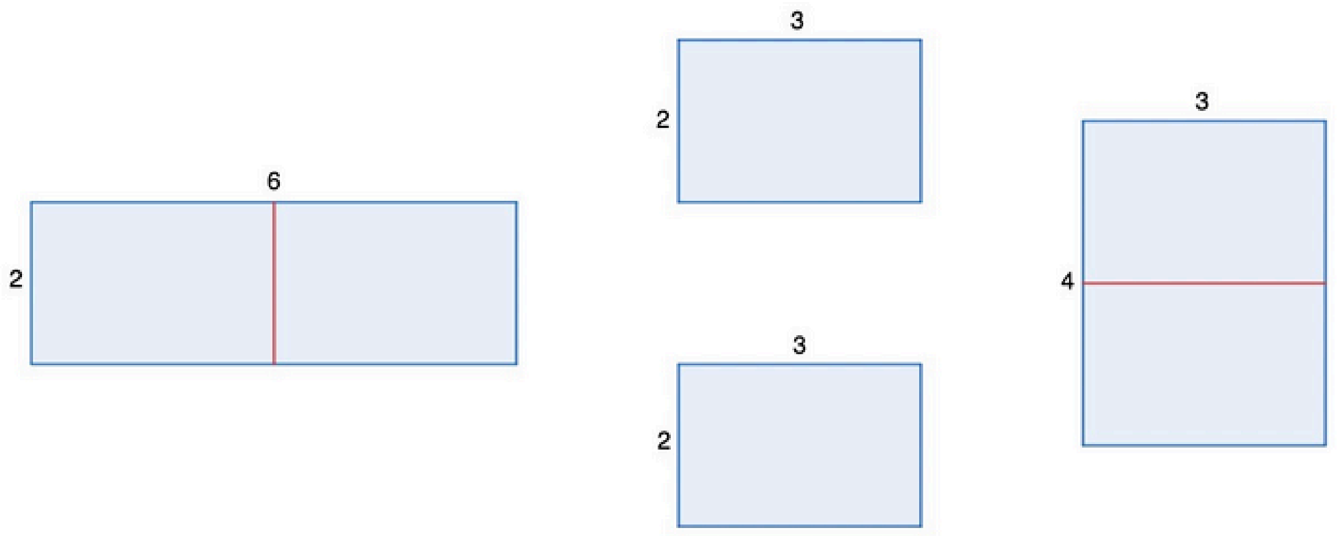
A. Rectangle Cutting

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

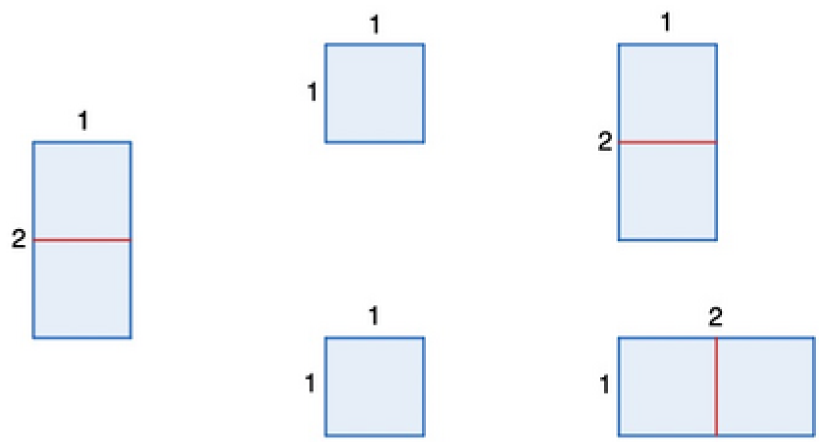
Bob has a rectangle of size $a \times b$. He tries to cut this rectangle into two rectangles with integer sides by making a cut parallel to one of the sides of the original rectangle. Then Bob tries to form some **other** rectangle from the two resulting rectangles, and he can rotate and move these two rectangles as he wishes.

Note that if two rectangles differ only by a 90° rotation, they are considered **the same**. For example, the rectangles 6×4 and 4×6 are considered the same.

Thus, from the 2×6 rectangle, another rectangle can be formed, because it can be cut into two 2×3 rectangles, and then these two rectangles can be used to form the 4×3 rectangle, which is different from the 2×6 rectangle.



However, from the 2×1 rectangle, another rectangle cannot be formed, because it can only be cut into two rectangles of 1×1, and from these, only the 1×2 and 2×1 rectangles can be formed, which are considered the same.



Help Bob determine if he can obtain some other rectangle, or if he is just wasting his time.

Input

Each test consists of multiple test cases. The first line contains a single integer t ($1 \le t \le 10^4$) — the number of test cases. This is followed by the description of the test cases.

The single line of each test case contains two integers a and b ($1 \le a, b \le 10^9$) — the size of Bob's rectangle.

Output

For each test case, output "Yes" if Bob can obtain another rectangle from the $a \times b$ rectangle. Otherwise, output "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 answers.

Standard Input	Standard Output
7 1 1 2 1 2 6 3 2 2 2 2 4 6 3	No No Yes Yes Yes Yes No

Note

In the first test case, the 1×1 rectangle cannot be cut into two rectangles, so another rectangle cannot be obtained from it.

In the fourth test case, the 3×2 rectangle can be cut into two 3×1 rectangles, and from these, the 1×6 rectangle can be formed.

In the fifth test case, the 2×2 rectangle can be cut into two 1×2 rectangles, and from these, the 1×4 rectangle can be formed.

,

B. Equalize

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

Vasya has two hobbies — adding permutations† to arrays and finding the most frequently occurring element. Recently, he found an array a and decided to find out the maximum number of elements equal to the same number in the array a that he can obtain after adding some permutation to the array a.

More formally, Vasya must choose exactly one permutation $p_1, p_2, p_3, \ldots, p_n$ of length n, and then change the elements of the array a according to the rule $a_i := a_i + p_i$. After that, Vasya counts how many times each number occurs in the array a and takes the maximum of these values. You need to determine the maximum value he can obtain.

†A permutation of length n is an array consisting of n distinct integers from 1 to n in arbitrary order. For example, $[2, 3, 1, 5, 4]$ is a permutation, but $[1, 2, 2]$ is not a permutation (2 appears twice in the array), and $[1, 3, 4]$ is also not a permutation ($n = 3$ but there is 4 in the array).

Input

Each test consists of multiple test cases. The first line contains a single integer t ($1 \le t \le 2 \cdot 10^4$) — the number of test cases. Then follows the description of the test cases.

The first line of each test case contains a single integer n ($1 \le n \le 2 \cdot 10^5$) — the length of the array a.

The second line of each test case contains n integers $a_1, a_2, \ldots, a_n$ ($1 \le a_i \le 10^9$) — the elements of the array a.

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

Output

For each test case, output a single number — the maximum number of elements equal to the same number after the operation of adding a permutation.

Standard Input	Standard Output

7 2 1 2 4 7 1 4 1 3 103 102 104 5 1 101 1 100 1 5 1 10 100 1000 1 2 3 1 3 1000000000 999999997 999999999	2 2 3 2 1 1 2

Note

In the first test case, it is optimal to choose $p = [2, 1]$. Then after applying the operation, the array a will be $[3, 3]$, in which the number 3 occurs twice, so the answer is 2.

In the second test case, one of the optimal options is $p = [2, 3, 1, 4]$. After applying the operation, the array a will be $[9, 4, 5, 5]$. Since the number 5 occurs twice, the answer is 2.

,

C. Physical Education Lesson

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

In a well-known school, a physical education lesson took place. As usual, everyone was lined up and asked to settle in "the first–k-th" position.

As is known, settling in "the first–k-th" position occurs as follows: the first k people have numbers $1, 2, 3, \ldots, k$, the next $k-2$ people have numbers $k-1, k-2, \ldots, 2$, the next k people have numbers $1, 2, 3, \ldots, k$, and so on. Thus, the settling repeats every $2k-2$ positions. Examples of settling are given in the "Note" section.

The boy Vasya constantly forgets everything. For example, he forgot the number k described above. But he remembers the position he occupied in the line, as well as the number he received during the settling. Help Vasya understand how many natural numbers k fit under the given constraints.

Note that the settling exists if and only if $k > 1$. In particular, this means that the settling **does not exist** for $k = 1$.

Input

Each test consists of multiple test cases. The first line contains a single integer t ($1 \le t \le 100$) — the number of test cases. This is followed by the description of the test cases.

The only line of each test case contains two integers n and x ($1 \le x < n \le 10^9$) — Vasya's position in the line and the number Vasya received during the settling.

Output

For each test case, output a single integer — the number of different k that fit under the given constraints.

It can be proven that under the given constraints, the answer is finite.

Standard Input	Standard Output
5 10 2 3 1 76 4 100 99 1000000000 500000000	4 1 9 0 1

Note

In the first test case, k equals $2, 3, 5, 6$ are suitable.

An example of settling for these k:

k / №	1	2	3	4	5	6	7	8	9	10
2	1	2	1	2	1	2	1	2	1	2
3	1	2	3	2	1	2	3	2	1	2

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

In the second test case, $k = 2$ is suitable.

,

D. Lonely Mountain Dungeons

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

Once, the people, elves, dwarves, and other inhabitants of Middle-earth gathered to reclaim the treasures stolen from them by Smaug. In the name of this great goal, they rallied around the powerful elf Timothy and began to plan the overthrow of the ruler of the Lonely Mountain.

The army of Middle-earth inhabitants will consist of several squads. It is known that each pair of creatures of **the same race**, which are in different squads, adds b units to the total strength of the army. But since it will be difficult for Timothy to lead an army consisting of a large number of squads, the total strength of an army consisting of k squads is reduced by $(k-1) \cdot x$ units. Note that the army always consists **of at least one squad**.

It is known that there are n races in Middle-earth, and the number of creatures of the i-th race is equal to c_i. Help the inhabitants of Middle-earth determine the maximum strength of the army they can assemble.

Input

Each test consists of multiple test cases. The first line contains a single integer t ($1 \le t \le 2 \cdot 10^4$) — the number of test cases. The description of the test cases follows.

The first line of each test case contains three integers n, b, and x ($1 \le n \le 2 \cdot 10^5$, $1 \le b \le 10^6, 0 \le x \le 10^9$) — the number of races and the constants b and x described above.

The second line of each test case contains n integers $c_1, c_2, \ldots, c_n$ ($1 \le c_i \le 2 \cdot 10^5$) — the number of creatures of each of the n races.

It is guaranteed that the sum of the values $c_1 + c_2 + \ldots + c_n$ over all test cases does not exceed $2 \cdot 10^5$.

Output

For each test case, output a single integer — the maximum strength of the army that the inhabitants of Middle-earth can assemble.

Standard Input	Standard Output
5 3 1 0 1 2 3 3 5 10 2 5 3 4 3 3 3 2 1 2 4 1 0 4 1 4 2 4 1 10 4 1 4 2	4 40 9 13 0

Note

In the first test case, the inhabitants of Middle-earth can form 3 squads. Since $x = 0$, the army's strength will not decrease due to the number of squads. The inhabitants can be distributed among the squads as follows:

- The single representative of the first species can be sent to the first squad.
- The first representative of the second species can be sent to the first squad, the second representative of the second species can be sent to the second squad. Then the total strength of the army will increase by $b = 1$.
- The first representative of the third species can be sent to the first squad, the second representative of the third species can be sent to the second squad, the third representative of the third species can be sent to the third squad. Then the total strength of the army will increase by $3 \cdot b = 3$, as they form three pairs in different squads.

Thus, the total strength of the army is 4.

In the second test case, the inhabitants of Middle-earth can form 3 squads. Since $x = 10$, the army's strength will decrease by 20. The inhabitants can be distributed among the squads as follows:

- The first representative of the first species can be sent to the first squad, the second representative of the first species can be sent to the second squad. Then the total strength of the army will increase by $b = 5$.
- The first and second representatives of the second species can be sent to the first squad, the third and fourth representatives of the second species can be sent to the second squad, the fifth representative of the second species can be sent to the third squad. Then the total strength of the army will increase by $8 \cdot b = 40$.
- The first representative of the third species can be sent to the first squad, the second representative of the third species can be sent to the second squad, the third representative of the third species can be sent to the third squad. Then the total strength of the army will increase by $3 \cdot b = 15$, as they form three pairs in different squads.

Thus, the total strength of the army is $5 + 40 + 15 - 20 = 40$.

E. Modular Sequence

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

You are given two integers x and y. A sequence a of length n is called *modular* if $a_1 = x$, and for all $1 < i \leq n$ the value of a_i is either $a_{i-1} + y$ or $a_{i-1} \bmod y$. Here $x \bmod y$ denotes the remainder from dividing x by y.

Determine if there exists a modular sequence of length n with the sum of its elements equal to S, and if it exists, find any such sequence.

Input

Each test contains multiple test cases. The first line contains the number of test cases t ($1 \leq t \leq 2 \cdot 10^4$). The description of the test cases follows.

The first and only line of each test case contains four integers n, x, y, and s ($1 \leq n \leq 2 \cdot 10^5$, $0 \leq x \leq 2 \cdot 10^5$, $1 \leq y \leq 2 \cdot 10^5$, $0 \leq s \leq 2 \cdot 10^5$) — the length of the sequence, the parameters x and y, and the required sum of the sequence elements.

The sum of n over all test cases does not exceed $2 \cdot 10^5$, and also the sum of s over all test cases does not exceed $2 \cdot 10^5$.

Output

For each test case, if the desired sequence exists, output "Yes" on the first line (without quotes). Then, on the second line, output n integers $a_1, a_2, \ldots, a_n$ separated by a space — the elements of the sequence a. If there are multiple suitable sequences, output any of them.

If the sequence does not exist, output "No" on a single line.

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 5 8 3 28 3 5 3 6 9 1 5 79	YES 8 11 2 2 5 NO NO

Note

In the first example, the sequence $[8, 11, 2, 5, 2]$ satisfies the conditions. Thus, $a_1 = 8 = x$, $a_2 = 11 = a_1 + 3$, $a_3 = 2 = a_2 \bmod 3$, $a_4 = 5 = a_3 + 3$, $a_5 = 2 = a_4 \bmod 3$.

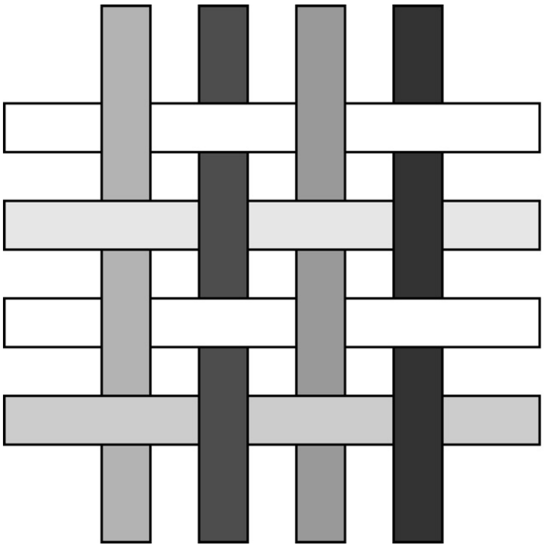
In the second example, the first element of the sequence should be equal to 5, so the sequence $[2, 2, 2]$ is not suitable.

F. Digital Patterns

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

Anya is engaged in needlework. Today she decided to knit a scarf from semi-transparent threads. Each thread is characterized by a single integer — the transparency coefficient.

The scarf is made according to the following scheme: horizontal threads with transparency coefficients $a_1, a_2, \ldots, a_n$ and vertical threads with transparency coefficients $b_1, b_2, \ldots, b_m$ are selected. Then they are interwoven as shown in the picture below, forming a piece of fabric of size $n \times m$, consisting of exactly nm nodes:



Example of a piece of fabric for $n = m = 4$.

After the interweaving tightens and there are no gaps between the threads, each node formed by a horizontal thread with number i and a vertical thread with number j will turn into a cell, which we will denote as (i, j). Cell (i, j) will have a transparency coefficient of $a_i + b_j$.

The *interestingness* of the resulting scarf will be the number of its sub-squares† in which there are no pairs of neighboring†† cells with the same transparency coefficients.

Anya has not yet decided which threads to use for the scarf, so you will also be given q queries to increase/decrease the coefficients for the threads on some ranges. After each query of which you need to output the interestingness of the resulting scarf.

†A sub-square of a piece of fabric is defined as the set of all its cells (i, j), such that $x_0 \le i \le x_0 + d$ and $y_0 \le j \le y_0 + d$ for some integers x_0, y_0, and d ($1 \le x_0 \le n - d$, $1 \le y_0 \le m - d$, $d \ge 0$).

††. Cells (i_1, j_1) and (i_2, j_2) are neighboring if and only if $|i_1 - i_2| + |j_1 - j_2| = 1$.

Input

The first line contains three integers n, m, and q ($1 \le n, m \le 3 \cdot 10^5$, $0 \le q \le 3 \cdot 10^5$) — the number of horizontal threads, the number of vertical threads, and the number of change requests.

The second line contains n integers $a_1, a_2, \ldots, a_n$ ($-10^9 \le a_i \le 10^9$) — the transparency coefficients for the horizontal threads, with the threads numbered from top to bottom.

The third line contains m integers $b_1, b_2, \ldots, b_m$ ($-10^9 \le b_i \le 10^9$) — the transparency coefficients for the vertical threads, with the threads numbered from left to right.

The next q lines specify the change requests. Each request is described by a quadruple of integers t, l, r, and x ($1 \le t \le 2, l \le r$, $-10^9 \le x \le 10^9$). Depending on the parameter t in the request, the following actions are required:

- $t = 1$. The transparency coefficients for the horizontal threads in the range $[l, r]$ are increased by x (in other words, for all integers $l \le i \le r$, the value of a_i is increased by x);
- $t = 2$. The transparency coefficients for the vertical threads in the range $[l, r]$ are increased by x (in other words, for all integers $l \le i \le r$, the value of b_i is increased by x).

Output

Output $(q+1)$ lines. In the $(i+1)$-th line ($0 \le i \le q$), output a single integer — the interestingness of the scarf after applying the first i requests.

Standard Input	Standard Output
4 4 0 1 1 2 3 1 2 2 3	20
3 3 2 1 1 1 2 2 8 1 2 3 1 2 2 3 -6	9 10 11
3 2 2 -1000000000 0 1000000000 -1000000000 1000000000 1 1 1 1000000000 2 2 2 -1000000000	8 7 7

Note

In the first example, the transparency coefficients of the cells in the resulting plate are as follows:

2	3	3	4
2	3	3	4
3	4	4	5
4	5	5	6

Then there are the following sub-squares that do not contain two neighboring cells with the same transparency coefficient:

- Each of the 16 cells separately;
- A sub-square with the upper left corner at cell $(3, 1)$ and the lower right corner at cell $(4, 2)$;
- A sub-square with the upper left corner at cell $(2, 3)$ and the lower right corner at cell $(3, 4)$;
- A sub-square with the upper left corner at cell $(2, 1)$ and the lower right corner at cell $(3, 2)$;
- A sub-square with the upper left corner at cell $(3, 3)$ and the lower right corner at cell $(4, 4)$.

In the second example, after the first query, the transparency coefficients of the horizontal threads are $[1, 2, 2]$. After the second query, the transparency coefficients of the vertical threads are $[2, -4, 2]$.