

Problem Rapid

Input file `stdin`
Output file `stdout`

Marius has a secret permutation p_1, p_2, \dots, p_n . Although Marius does not want to reveal the values of the elements in the permutation, he is willing to answer a limited number of queries of the following type:

- `? i j` ($1 \leq i, j \leq n, i \neq j$) — What is the minimum value between p_i and p_j ?

Find the positions of the numbers $n - 1$ and n in Marius' permutation.

Note! If $p_i = n - 1$ and $p_j = n$, then both (i, j) and (j, i) are considered correct answers.



Input Data

Each test contains multiple test cases. The first line will contain two numbers t and lim — the number of test cases and the maximum number of queries per test case that Marius is willing to answer.

Interaction Protocol

The hidden permutation is fixed at the beginning of each test case (the interactor is not adaptive).

The interaction for each test case begins with reading n — the size of the permutation p .

If $n = -1$, then the answer for the last test case was incorrect. To obtain a verdict of "Wrong answer" instead of another verdict, we recommend immediately terminating the program.

After this, to ask Marius what the minimum between p_i and p_j is, print a line in the following format:

- `? i j` ($1 \leq i, j \leq n, i \neq j$)

Marius will respond with a number $x = \min(p_i, p_j)$, which you can read from `stdin`.

If the query is invalid or if it is the $lim + 1$ -th query in the test case, x will be equal to -1 . In this case, to obtain a verdict of "Wrong answer" instead of another verdict, we recommend immediately terminating the program.

To display the answer for a test case, print a line in the following format:

- `! i j` ($1 \leq i, j \leq n, i \neq j$)

After displaying the answer, you can read the value of n for the next test case.

If $p_i = n - 1$ and $p_j = n$, then any order of positions i and j is considered correct.

After each printed line, you must output an endline and flush the `stdout`. In C++, this can be done with the instruction `cout<<endl`;

Scoring

Let q be the maximum number of queries used for a test case in a given test. The percentage of the total score awarded for that test will be:

- 1, if $q \leq 10,000$;
- $\frac{\text{lim}+1-q}{\text{lim}+1-10,000}$, if $10,000 < q \leq \text{lim}$;
- 0, if $q > \text{lim}$.

Restrictions

#	Points	Restrictions
1	4	$t = 25, 2 \leq n \leq 140, \text{lim} = 12\,000$
2	8	$t = 25, n = 3\,000, \text{lim} = 12\,000$
3	16	$t = 25, n = 5\,000, \text{lim} = 12\,000$
4	20	$t = 25, n = 7\,500, \text{lim} = 12\,000$
5	12	$t = 25, n = 9\,300, \text{lim} = 11\,000$
6	24	$t = 25, n = 9\,600, \text{lim} = 10\,200$
7	16	$t = 25, n = 9\,900, \text{lim} = 10\,000$

Examples

Input file	Output file
1 12000 5	? 1 2
1	? 1 3
3	? 3 5
4	! 5 3

Explanations

Marius' hidden permutation is $[3, 1, 5, 2, 4]$.

The interaction begins with reading the number of test cases $t = 1$, the maximum number of queries per test case $\text{lim} = 12000$, and the size $n = 5$ of the permutation in the first test case.

After this:

- For the query "What is the minimum value between p_1 and p_2 ?", Marius responds with 1.
- For the query "What is the minimum value between p_1 and p_3 ?", Marius responds with 3.
- For the query "What is the minimum value between p_3 and p_5 ?", Marius responds with 4.

Accepted answers for this test case are $(3, 5)$ and $(5, 3)$, since $a_5 = n - 1$ and $a_3 = n$.

Problem Craiova

Input file `stdin`
Output file `stdout`

In the stands of a match played by Universitatea Craiova, there are N groups of supporters arranged in a linear manner, where each group i ($1 \leq i \leq N$) initially consists of N people.

A brawl between two adjacent groups of sizes a and b respectively will have the following consequences:

- If $a > b$, then the first group will remain with $a - b$ people, the second group will have 0 people left, and the remaining people from the first group will occupy the section of the stands previously occupied by the second group.
- If $a < b$, then the second group will remain with $b - a$ people, the first group will have 0 people left, and the remaining people from the second group will occupy the section of the stands previously occupied by the first group.
- If $a = b$, then both groups will leave (by mutual agreement) from the stadium, and the regions in the stands where these two groups were will be occupied by the neighboring groups (if they exist).



Starting from the initial state, how many groups can remain alone in the stands after a series of brawls?

Input Data

Each test contains multiple test cases. The first line will contain the number of test cases T .

Each test case contains a single number N — the number of groups in the stands.

Output Data

For each test case, print the number of groups that can remain alone in the stands after a series of brawls, starting from the initial state.

Restrictions

- $1 \leq T \leq 2 \cdot 10^5$.
- $1 \leq N \leq 10^{18}$.

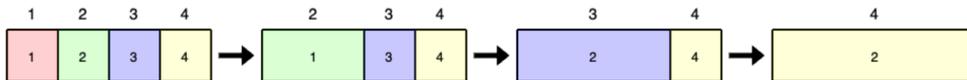
#	Points	Restrictions
1	17	$1 \leq T, N \leq 10$
2	18	$1 \leq T, N \leq 50$
3	26	$1 \leq T, N \leq 5000$
4	28	$1 \leq N \leq 10^9$
5	11	No additional restrictions

Examples

Input file	Output file
5	1
4	3
5	4
8	7
11	8
12	

Explanations

- In the first test case, there are 4 groups, with sizes 1, 2, 3, and 4 people respectively. The only group that can remain alone in the stadium is group 4, after the following series of brawls:

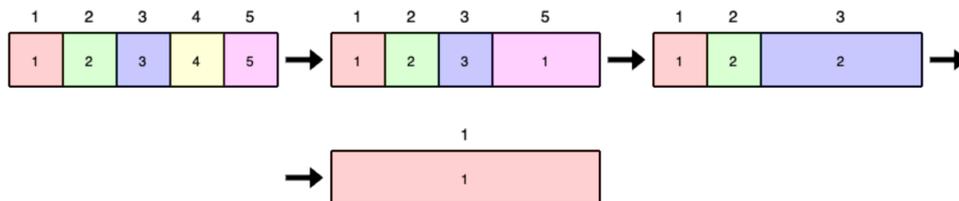


A series of brawls after which group 4 remains alone in the stadium.

The numbers inside the cells represent the sizes of the groups, and the numbers above the cells represent the group controlling the respective region in the stands.

- In the second test case, the groups that can remain alone in the stadium are 1, 3, and 5.

A series of brawls after which group 1 is the only one left is described below:



A series of brawls after which group 1 remains alone in the stadium. In the last brawl, groups 2 and 3 have the same number of people, so they leave the stands by mutual agreement. The regions in the stands where groups 2 and 3 were are occupied by group 1.

- In the third test case, the groups that can remain alone in the stadium are 3, 4, 6, and 8.
- In the fourth test case, the groups that can remain alone in the stadium are 3, 4, 5, 6, 7, 9, and 11.
- In the fifth test case, the groups that can remain alone in the stadium are 3, 4, 5, 6, 7, 8, 10, and 12.

Problem București, Steaua București

Input file `stdin`
Output file `stdout`

Steaua București is the team with the greatest achievements in the history of Romanian football.

It has had incredible seasons. For example, in 1986, it managed to win the **UEFA Champions League**.

In 2024, the team won the **Romanian Championship**, and the staff organized a celebration.

After a few hours of fun, the coaches Elias and Mihai started thinking about interesting statistics from the team's history.

The two were analyzing the list of the team's points at the end of each season over the past N years. Then, Mihai posed the following problem to Elias:

We define a **change of maximum** as the appearance of a value greater than all previously encountered values. If we are at the first value, then this property is automatically fulfilled.

The question is: Given the list of the team's points at the end of each season over the past N years, what is the longest continuous sequence of values among these N with exactly K_1 **changes of maximum** when traversing the values from **left to right** and exactly K_2 **changes of maximum** when traversing the values from **right to left**?

Since the celebration was a great success, coaches Mihai and Elias cannot solve the problem on their own, so they are asking for your help.

Input

The first line contains the numbers N , K_1 , and K_2 , separated by a space.

The second line contains an array A of length N , representing the scores at the end of each season over the past N years, in chronological order, separated by a space.

Output

The output contains the **length** of the longest continuous sequence with the property described in the statement if there is at least one, or -1 otherwise.

Restrictions

- $1 \leq N \leq 1\,000\,000$
- $1 \leq K_1, K_2 \leq N$
- $1 \leq A_i \leq N, (1 \leq i \leq N)$



#	Points	Restrictions
1	20	$1 \leq N \leq 500$
2	25	$501 \leq N \leq 2\,000$
3	30	$A_i \neq A_j, (1 \leq i < j \leq N)$
4	25	No additional restrictions.

Examples

Input file	Output file
12 3 5 1 3 1 2 1 3 4 5 4 3 2 1	11

Explanations

There are 2 sequences where there are exactly 3 **changes of maximum** from left to right and exactly 5 **changes of maximum** from right to left: (2, 12), (6, 12). The longest among them is (2, 12), having a length of 11.

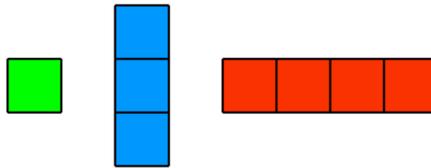
Problem Dinamo

Input file `stdin`
Output file `stdout`

Due to the positive reviews received by the committee regarding a certain problem involving tiling on matrices, we present the following problem:

A tile is called a *ketris tile* if:

1. It has a rectangular shape, and the lengths of its sides (in meters) are natural numbers.
2. At least one of its sides has a length of 1 meter.



Examples of *ketris tiles*

A complete coverage of a surface with *ketris tiles* is called a *ketris tiling*.

How many *ketris tilings* exist for a rectangular surface with side lengths of n and m meters? Since the answer can be very large, print the result modulo $10^9 + 7$.

Two *ketris tilings* are considered different if there exists a *ketris tile* that appears in the first *tiling* in a certain location but not in the second.

Input Data

The first line of the input file contains two natural numbers n and m — the height and width of the surface, respectively.

Output Data

Print the number of possible *ketris tilings*. Since the answer can be very large, print it modulo $10^9 + 7$.

Restrictions

- $1 \leq n \leq 10$
- $1 \leq m \leq 10^{18}$

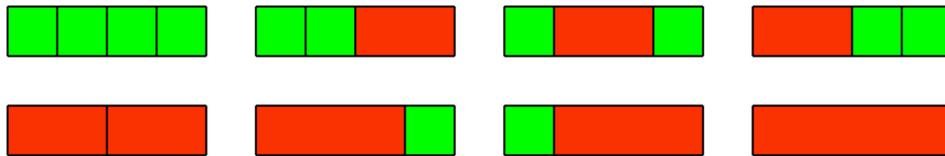
#	Points	Restrictions
1	7	$n = 1$
2	4	$n = 2, m \leq 1000.$
3	5	$n = 2, m \leq 10^6.$
4	8	$n = 2, m \leq 10^{18}.$
5	13	$n, m \leq 5.$
6	16	$n \leq 7, m \leq 1000.$
7	26	$n \leq 7.$
8	21	No additional restrictions

Examples

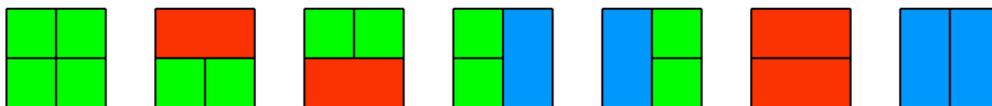
Input file	Output file
1 4	8
2 2	7
2 3	29
2 11	3381689
6 5	441987135
7 420	496377800

Explanations

In the first example, there are 8 possible *ketris tilings*:



In the second example, there are 7 possible *ketris tilings*:



Problem CFR Cluj

Input file `stdin`
Output file `stdout`

Because time means money, and students don't have either, much like CFR Cluj's trophy cabinet, this problem will not have a story:

You have to process q queries of the following types:

- 1 x y : Activate point (x, y) . It is guaranteed that an already active point with the same x or y coordinate does not exist.
- 2 x y : Deactivate point (x, y) . It is guaranteed that point (x, y) is active at the moment.
- 3 x y : Draw open line segments (segments which do not contain their endpoints) from point (x, y) , which is active at the moment, to all the other active points, and then print the maximum total number of intersections of a "plus" with these drawn segments (if more than one intersection occurs at the same point, all of them are counted separately). A "plus" centered in (x_0, y_0) , where $x_0, y_0 \in \mathbb{R}$, is a shape formed from the two lines with equations $x = x_0$ and $y = y_0$.



Observation: The open line segments drawn in queries of type 3 are deleted after printing the answer (i.e. they do not persist through queries).

Input

The first line contains an integer, the number of queries q .

The next q lines will each contain three integers t, x and y — the parameters of the queries.

Output

For each query of type 3, print the maximum number of intersections of a "plus" with the drawn open line segments.

Restrictions

- $2 \leq q \leq 5 \cdot 10^5$
- $1 \leq t \leq 3$
- $-10^9 \leq x, y \leq 10^9$
- It is guaranteed that there is at least one query of type 3.
- It is guaranteed that for queries of type 1, there does not exist any active point with the same x or y coordinate.
- It is guaranteed that for queries of type 2 and 3, the point (x, y) is active.

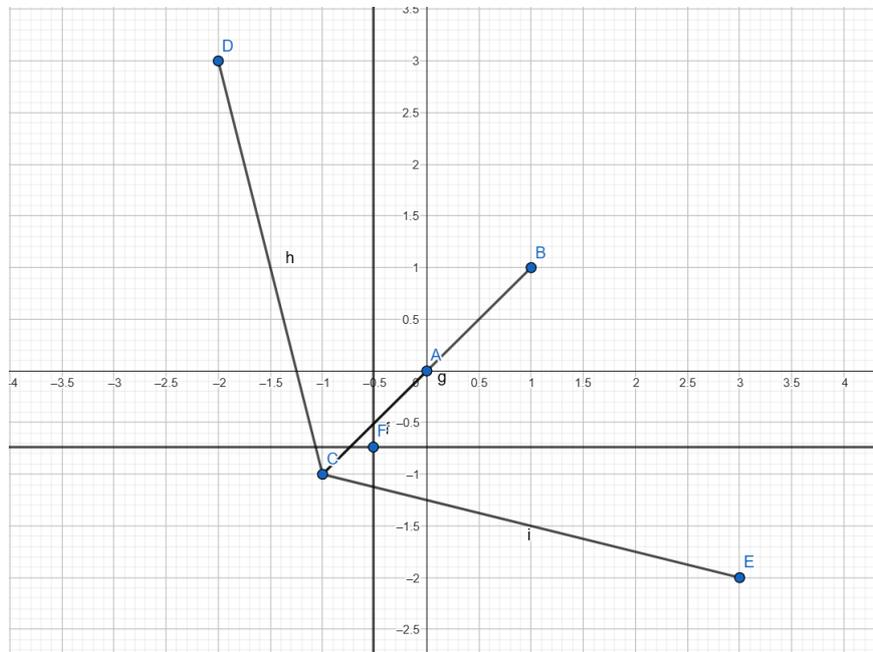
#	Points	Restrictions
1	24	$q \leq 3000$
2	40	Queries of type 2 do not exist, all the queries with type 1 appear before all the queries of type 3
3	36	No additional restrictions

Examples

Input file	Output file
17 1 0 0 1 1 1 1 -1 -1 3 0 0 1 -2 3 1 3 -2 3 0 0 3 -1 -1 2 0 0 1 4 4 3 1 1 2 1 1 1 2 2 3 4 4 3 2 2 3 -2 3 3 3 -2	2 4 6 4 8 4 7 7
4 1 905580697 285643736 2 905580697 285643736 1 687880848 -231766091 3 687880848 -231766091	0
13 1 0 0 1 1 1 1 -1 -1 1 -2 3 1 3 -2 1 4 4 1 2 2 3 -1 -1 3 1 1 3 4 4 3 2 2 3 -2 3 3 3 -2	10 6 12 8 11 11

Explanation

Down below there is a graphic for the first example and query $(3, -1, -1)$, illustrating the 6 intersections:



The active points are : $C(-1, -1)$, $A(0, 0)$, $B(1, 1)$, $D(-2, 3)$, $E(3, -2)$.

A "plus" centered in point $(-0.5, -0.75)$ has 6 intersections, since:

- Its vertical line intersects the segments CA, CB and CE .
- Its horizontal line intersects the segments CA, CB and CD .

Problem Halftime Changes

Input file `stdin`
Output file `stdout`

You are given a tree with N nodes, with root in node 1, and two arrays A and B , having N elements, indexed from 1 to N .

It is guaranteed that each level of the tree contains at most 100 nodes.

You have to select some nodes Y for which it is possible to choose:

- a subset $S(Y)$ of nodes on the path from the root to Y ; $S(Y)$ can be empty
- a node $X(Y)$ that is a **proper ancestor** of both Y and all the nodes from $S(Y)$.

Such as the following condition holds:

$$A[X(Y)] - \sum_{v \in S(Y)} A[v] = B[Y]$$

A **proper ancestor** of V is any node U such that node U is an ancestor of node V and U is not the same node as V .

Before solving the problem, just like Nea' Gigi, the owner of FCSB, often makes miraculous changes at halftime, you are allowed to make changes. You can change at most K elements of the array A . The changed values are at least 1 and at most 10^7 .

Take note that K is not given in the input. In the description of the subtasks it is specified the value of K depending on N .

After the changes are made and the nodes are selected as specified in the above statement, let M be the number of selected nodes Y .

You will be scored based on how large the number M is. **Note:** You are not required to find the maximum possible value of M . You will get the points for a subtask if M meets the condition specified in the subtask description.

Input

The first line contains the number N . The second line contains $N - 1$ values, where the i -th value represents the parent of node $i + 1$. The third line contains N integers, representing the array A . The last line contains N integers, representing the array B .

Output

The first line contains the elements of the array A after the changes, in order from 1 to N . The second line contains an integer M , representing the number of nodes Y that satisfy the given condition. The following M lines contains data about chosen nodes Y . Each of these M lines has the following format:

- The node Y which satisfies the condition.
- The chosen **proper ancestor** node $X(Y)$.
- The size of the selected subset $S(Y)$ of nodes (can be 0).
- The nodes in the subset $S(Y)$ (if any).

The values on the same line in the output are separated by a single space.

Restrictions

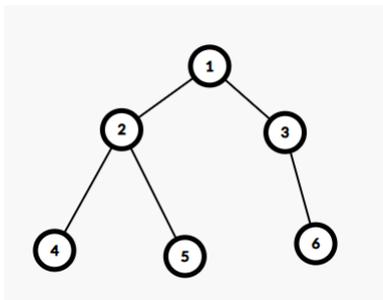
- $1 \leq N \leq 400\,000$
- $1 \leq A_i, B_i \leq N, (1 \leq i \leq N)$
- The solution is not unique and you will be scored according to the subtasks description.

#	Points	Restrictions
1	12.5	$N = 5000, K = 1500, 10 \leq M$
2	4	$N = 5000, K = 1500, 70 \leq M$
3	8.5	$N = 5000, K = 1500, 500 \leq M$
4	25	$N = 5000, K = 1500, 3500 \leq M$
5	12.5	$N = 400\,000, K = 3000, 20 \leq M$
6	4	$N = 400\,000, K = 3000, 1000 \leq M$
7	8.5	$N = 400\,000, K = 3000, 80\,000 \leq M$
8	25	$N = 400\,000, K = 3000, 397\,000 \leq M$

Examples

Input file	Output file
6 1 1 2 2 3 15 7 20 2 3 2 10 5 9 3 2 7	10 5 1 2 3 2 5 2 1 1 2 3 1 1 3 4 2 1 4 5 1 2 2 5 6 1 2 3 6

Explanations



In the array A , 3 changes were made for positions 1, 2, and 3.

The total number of nodes Y that satisfy the condition after the changes is $M = 5$.

For example, one of the selected nodes is $Y = 2$, $X(2) = 1$ and the subset $S(Y)$ has 1 node, $S(2) = \{2\}$. It can be observed that $A[1] - A[2] = 10 - 5 = 5 = B[2]$.

Another selected node is $Y = 5$, $X(5) = 1$ and the subset $S(Y)$ has 2 nodes, $S(5) = \{2, 5\}$. It can be observed that $A[1] - (A[2] + A[5]) = 10 - (5 + 3) = 2 = B[5]$.

Problem U Cluj

Input file `stdin`
Output file `stdout`

After RANDy sold U Cluj for their terrible performance last season, he is now wondering what to invest the money into. After hearing about the prestigious fireman olympiad, he had a marvelous idea: a fireman department business.

You are very eager to become a fireman, so you go to an interview and got the following question: "You are given an integer N and an integer P . For every K from 1 to P , compute the number of ways you can write N as a sum of K numbers which can be written as $2^x - 1$ for some integer $x \geq 1$, modulo $10^9 + 7$ ".

RANDy promises you that if you solve this problem, he will give you his best hose and put it on your shoulder, thus making you the most prolific fireman the world has ever seen, and you can be sure you'll get the gold medal at the fireman olympics!



Input

The first line of the input contains two integers, N and P .

Input

The first line of the output contains P integers, separated by spaces. The K^{th} integer represents the number of ways you can write N as a sum of K numbers which can be written as $2^x - 1$ for some integer $x \geq 1$, modulo $10^9 + 7$.

Restrictions

- $1 \leq N \leq 10^{18}$.
- $1 \leq P \leq 500$.

#	Points	Restrictions
1	7	$1 \leq N \leq 500, 1 \leq P \leq 15$
2	14	$1 \leq N \leq 50000, 1 \leq P \leq 50$
3	28	$1 \leq P \leq 50$
4	22	$1 \leq P \leq 150$
5	29	No additional constraints.

Examples

Input file	Output file	Explanations
4 2	0 1	For $K = 1$, there exists no solution. For $K = 2$, $4 = 3 + 1$.
14 4	0 1 0 1	For $K = 1$, there exists no solution. For $K = 2$, $14 = 7 + 7$. For $K = 3$, there exists no solution. For $K = 4$, $14 = 7 + 3 + 3 + 1$.
9 3	0 0 2	For $K = 1$, there exists no solution. For $K = 2$, there exists no solution. For $K = 3$, $9 = 3 + 3 + 3$ and $9 = 7 + 1 + 1$.