Codeforces Round #694 (Div. 1)

A. Strange Birthday Party

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

Petya organized a strange birthday party. He invited n friends and assigned an integer k_i to the i-th of them. Now Petya would like to give a present to each of them. In the nearby shop there are m unique presents available, the j-th present costs c_j dollars ($1 \le c_1 \le c_2 \le \ldots \le c_m$). It's **not** allowed to buy a single present more than once.

For the i-th friend Petya can either buy them a present $j \le k_i$, which costs c_j dollars, or just give them c_{k_i} dollars directly.

Help Petya determine the minimum total cost of hosting his party.

Input

The first input line contains a single integer t ($1 \le t \le 10^3$) — the number of test cases.

The first line of each test case contains two integers n and m ($1 \le n, m \le 3 \cdot 10^5$) — the number of friends, and the number of unique presents available.

The following line contains n integers $k_1, k_2, \ldots, k_n$ ($1 \le k_i \le m$), assigned by Petya to his friends.

The next line contains m integers $c_1, c_2, \ldots, c_m$ ($1 \le c_1 \le c_2 \le \ldots \le c_m \le 10^9$) — the prices of the presents.

It is guaranteed that sum of values n over all test cases does not exceed $3 \cdot 10^5$, and the sum of values m over all test cases does not exceed $3 \cdot 10^5$.

Output

For each test case output a single integer — the minimum cost of the party.

Examples

input
2 5 4 2 3 4 3 2 3 5 12 20 5 5 5 4 3 2 1 10 40 90 160 250
output
30 190

input
1 1 1 1 1
output
1

Note

In the first example, there are two test cases. In the first one, Petya has 5 friends and 4 available presents. Petya can spend only 30 dollars if he gives

- 5 dollars to the first friend.
- A present that costs 12 dollars to the second friend.
- A present that costs 5 dollars to the third friend.
- A present that costs 3 dollars to the fourth friend.
- 5 dollars to the fifth friend.

In the second one, Petya has 5 and 5 available presents. Petya can spend only 190 dollars if he gives

- A present that costs 10 dollars to the first friend.
- A present that costs 40 dollars to the second friend.
- 90 dollars to the third friend.

- 40 dollars to the fourth friend.
- 10 dollars to the fifth friend.

B. Strange Definition

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Let us call two integers x and y *adjacent* if $\frac{lcm(x,y)}{gcd(x,y)}$ is a perfect square. For example, 3 and 12 are adjacent, but 6 and 9 are not.

Here $gcd(x,y)$ denotes the greatest common divisor (GCD) of integers x and y, and $lcm(x,y)$ denotes the least common multiple (LCM) of integers x and y.

You are given an array a of length n. Each second the following happens: each element a_i of the array is **replaced** by the product of all elements of the array (including itself), that are adjacent to the current value.

Let d_i be the number of adjacent elements to a_i (including a_i itself). The *beauty* of the array is defined as $\max_{1 \le i \le n} d_i$.

You are given q queries: each query is described by an integer w, and you have to output the beauty of the array after w seconds.

Input

The first input line contains a single integer t ($1 \le t \le 10^5$) — the number of test cases.

The first line of each test case contains a single integer n ($1 \le n \le 3 \cdot 10^5$) — the length of the array.

The following line contains n integers $a_1, \ldots, a_n$ ($1 \le a_i \le 10^6$) — array elements.

The next line contain a single integer q ($1 \le q \le 3 \cdot 10^5$) — the number of queries.

The following q lines contain a single integer w each ($0 \le w \le 10^{18}$) — the queries themselves.

It is guaranteed that the sum of values n over all test cases does not exceed $3 \cdot 10^5$, and the sum of values q over all test cases does not exceed $3 \cdot 10^5$

Output

For each query output a single integer — the beauty of the array at the corresponding moment.

Example

input
2 4 6 8 4 2 1 0 6 12 3 20 5 80 1 1 1
output
2 3

Note

In the first test case, the initial array contains elements $[6, 8, 4, 2]$. Element $a_4 = 2$ in this array is adjacent to $a_4 = 2$ (since $\frac{lcm(2,2)}{gcd(2,2)} = \frac{2}{2} = 1 = 1^2$) and $a_2 = 8$ (since $\frac{lcm(8,2)}{gcd(8,2)} = \frac{8}{2} = 4 = 2^2$). Hence, $d_4 = 2$, and this is the maximal possible value d_i in this array.

In the second test case, the initial array contains elements $[12, 3, 20, 5, 80, 1]$. The elements adjacent to 12 are $\{12, 3\}$, the elements adjacent to 3 are $\{12, 3\}$, the elements adjacent to 20 are $\{20, 5, 80\}$, the elements adjacent to 5 are $\{20, 5, 80\}$, the elements adjacent to 80 are $\{20, 5, 80\}$, the elements adjacent to 1 are $\{1\}$. After one second, the array is transformed into $[36, 36, 8000, 8000, 8000, 1]$.

C. Strange Shuffle

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

This is an interactive problem.

n people sitting in a circle are trying to shuffle a deck of cards. The players are numbered from 1 to n, so that players i and $i + 1$

are neighbours (as well as players 1 and n). Each of them has exactly k cards, where k is **even**. The left neighbour of a player i is player $i-1$, and their right neighbour is player $i+1$ (except for players 1 and n, who are respective neighbours of each other).

Each turn the following happens: if a player has x cards, they give $\lfloor x/2 \rfloor$ to their neighbour on the left and $\lceil x/2 \rceil$ cards to their neighbour on the right. This happens for all players simultaneously.

However, one player p is the impostor and they just give all their cards to their neighbour on the right. You know the number of players n and the number of cards k each player has initially, but p is unknown to you. Your task is to determine the value of p, by asking questions like "how many cards does player q have?" for an index q of your choice. After each question all players will make exactly one move and give their cards to their neighbours. You need to find the impostor by asking no more than 1000 questions.

Input

The first line contains two integers n and k ($4 \le n \le 10^5$, $2 \le k \le 10^9$, k is even) — the number of players and the number of cards.

Interaction

You can ask questions by printing "? q". The answer to this question is the number of cards player q has now ($1 \le q \le n$). The shuffling process starts immediately after your first question, so the answer to the first one is always equal to k.

Once you have identified the impostor, you can output the answer by printing "! p", where p is the player who is the impostor ($1 \le p \le n$). Then you have to terminate your program.

You have to find the impostor by asking no more than 1000 questions.

After printing a query do not forget to output end of line and flush the output. Otherwise, you will get `Idleness limit exceeded`. To do this, use:

- `fflush(stdout)` or `cout.flush()` in C++;
- `System.out.flush()` in Java;
- `flush(output)` in Pascal;
- `stdout.flush()` in Python;
- see documentation for other languages.

Hacks

To make a hack, use the following test format.

The only line of input should contain three integers n, k and p ($4 \le n \le 10^5$, $2 \le k \le 10^9$, k is even, $1 \le p \le n$) — the number of people, the number of cards each person has initially, and the position of the impostor.

Example

input
4 2 2 1 2 3 2
output
? 1 ? 1 ? 2 ? 3 ? 4 ! 2

Note

In the example the cards are transferred in the following way:

- $2\ 2\ 2\ 2$ — player 1 has 2 cards.
- $1\ 2\ 3\ 2$ — player 1 has 1 card.

After this turn the number of cards remains unchanged for each player.

D. Strange Housing

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input

output: standard output

Students of Winter Informatics School are going to live in a set of houses connected by underground passages. Teachers are also going to live in some of these houses, but they can not be accommodated randomly. For safety reasons, the following must hold:

- All passages between two houses will be closed, if there are no teachers in both of them. All other passages will stay open.
- It should be possible to travel between any two houses using the underground passages that are **open**.
- Teachers should not live in houses, directly connected by a passage.

Please help the organizers to choose the houses where teachers will live to satisfy the safety requirements or determine that it is impossible.

Input

The first input line contains a single integer t — the number of test cases ($1 \le t \le 10^5$).

Each test case starts with two integers n and m ($2 \le n \le 3 \cdot 10^5$, $0 \le m \le 3 \cdot 10^5$) — the number of houses and the number of passages.

Then m lines follow, each of them contains two integers u and v ($1 \le u, v \le n$, $u \ne v$), describing a passage between the houses u and v. It is guaranteed that there are no two passages connecting the same pair of houses.

The sum of values n over all test cases does not exceed $3 \cdot 10^5$, and the sum of values m over all test cases does not exceed $3 \cdot 10^5$.

Output

For each test case, if there is no way to choose the desired set of houses, output "NO". Otherwise, output "YES", then the total number of houses chosen, and then the indices of the chosen houses in arbitrary order.

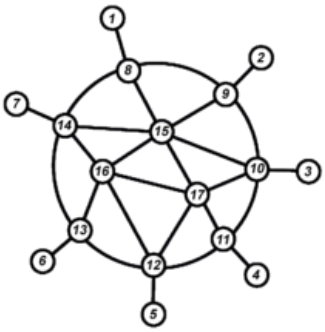
Examples

input
2 3 2 3 2 2 1 4 2 1 4 2 3
output
YES 2 1 3 NO

input
1 17 27 1 8 2 9 3 10 4 11 5 12 6 13 7 14 8 9 8 14 8 15 9 10 9 15 10 11 10 15 10 17 11 12 11 17 12 13 12 16 12 17 13 14 13 16 14 16 14 15 15 16 15 17 16 17
output
YES 8 1 3 4 5 6 9 14 17

Note

The picture below shows the second example test.



E. Strange Permutation

time limit per test: 3 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Alice had a permutation $p_1, p_2, \ldots, p_n$. Unfortunately, the permutation looked very boring, so she decided to change it and choose some **non-overlapping** subranges of this permutation and reverse them. The cost of reversing a single subrange $[l, r]$ (elements from position l to position r, inclusive) is equal to $r - l$, and the cost of the operation is the sum of costs of reversing individual subranges. Alice had an integer c in mind, so she only considered operations that cost no more than c.

Then she got *really* bored, and decided to write down all the permutations that she could possibly obtain by performing exactly one operation on the initial permutation. Of course, Alice is very smart, so she wrote down each obtainable permutation exactly once (no matter in how many ways it can be obtained), and of course the list was sorted lexicographically.

Now Bob would like to ask Alice some questions about her list. Each question is in the following form: what is the i-th number in the j-th permutation that Alice wrote down? Since Alice is too bored to answer these questions, she asked you to help her out.

Input

The first line contains a single integer t ($1 \le t \le 30$) — the number of test cases.

The first line of each test case contains three integers n, c, q ($1 \le n \le 3 \cdot 10^4$, $1 \le c \le 4$, $1 \le q \le 3 \cdot 10^5$) — the length of the permutation, the maximum cost of the operation, and the number of queries.

The next line of each test case contains n integers $p_1, p_2, \ldots, p_n$ ($1 \le p_i \le n$, $p_i \ne p_j$ if $i \ne j$), describing the initial permutation.

The following q lines describe the queries. Each of them contains two integers i and j ($1 \le i \le n$, $1 \le j \le 10^{18}$), denoting parameters of this query.

It is guaranteed that the sum of values n over all test cases does not exceed $3 \cdot 10^5$, and the sum of values q over all test cases does not exceed $3 \cdot 10^5$.

Output

For each query output the answer for this query, or -1 if j-th permutation does not exist in her list.

Examples

input
2 3 1 9 1 2 3 1 1 2 1 3 1 1 2 2 2 3 2 1 3 2 3 3 3 6 4 4 6 5 4 3 1 2 1 1 3 14 1 59 2 6

output
1 2 3 1 3 2 2 1 3

```
1
4
-1
5
```

input
1 12 4 2 1 2 3 4 5 6 7 8 9 10 11 12 2 20 2 21
output
2 2

Note

In the first test case, Alice wrote down the following permutations: $[1, 2, 3]$, $[1, 3, 2]$, $[2, 1, 3]$.

Note that, for a permutation $[3, 2, 1]$ Alice would have to reverse the whole array, and it would cost her 2, which is greater than the specified value $c = 1$. The other two permutations can not be obtained by performing exactly one operation described in the problem statement.

F. Strange Covering

time limit per test: 6 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

You are given n points on a plane.

Please find the minimum sum of areas of two axis-aligned rectangles, such that each point is contained in at least one of these rectangles.

Note that the chosen rectangles can be degenerate. Rectangle contains all the points that lie inside it or on its boundary.

Input

The first line contains one integer t ($1 \le t \le 2 \cdot 10^5$) — the number of test cases.

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

The following n lines contain the coordinates of the points x_i and y_i ($0 \le x_i, y_i \le 10^9$). It is guaranteed that the points are distinct.

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

Output

For each test case print one integer — the minimum sum of areas.

Example

input
3 2 9 1 1 6 2 8 10 0 7 4 0 0 1 1 9 9 10 10
output
0 0 2

Note

In the first two test cases the answer consists of 2 degenerate rectangles. In the third test case one of the possible answers consists of two rectangles 1×1 with bottom left corners $(0, 0)$ and $(9, 9)$.