

Problem A. Is It A Math Problem?

Input file: standard input
Output file: standard output
Time limit: 1 second
Balloon Color: Light Steel Blue

As the title said, this is a Mathementation problem!

You will be given a number n and you have to print two **integer** numbers a and b satisfying the following conditions:

$$a^b = \prod_{d|n} d$$
$$0 \leq a, b \leq 10^{18}$$

$(\prod_{d|n} d)$ is the product of all divisors of n .

Input

The first and the only line of the input will contain a single integer n ($1 \leq n \leq 10^{12}$).

Output

Print two integers a, b ($0 \leq a, b \leq 10^{18}$). if there are many solutions print any one of them.

Example

standard input	standard output
10	100 1

Note

For the first test case, the divisors of 10 are:

1, 2, 5, 10 .

So, a could be equal to 100 and b could be equal to 1.

and $100^1 = 1 * 2 * 5 * 10$.

Problem B. How Aswad Use Telegram?

Input file: standard input
Output file: standard output
Time limit: 1 second
Balloon Color: Dark Cyan

Aswad is a member of a Telegram group chat with n participants, he has a unique communication style. Although *Aswad* he doesn't like engaging in conversations, he still like to participate in the group by responding to messages.

However, instead of responding immediately, *Aswad* waits for a set amount of time before sending a message. Specifically, *Aswad* waits for k minutes after the previous message is sent before responding.

It's **important** to note that if another message is sent during the waiting period, *Aswad* will ignore the previous message and will wait for a new k minutes. This approach allows *Aswad* to take his time in crafting a response and to avoid getting caught up in the rapid pace of the chat.

Given a list of m timestamps for when messages were sent in the group chat, and the number of minutes k . your task is to determine how many messages *Aswad* has sent in the group chat.

Input

The first line of the input contains a single integer t ($1 \leq t \leq 1000$) — the number of test cases. The description of the test cases follows.

The first line of each test case contains three integers n, k , and m , where n represents the number of members in the group chat, k represents the time in minutes that *Aswad* waits before responding, and m represents the number of messages sent in the group chat ($1 \leq n \leq 1000, 1 \leq k, m \leq 1440$).

The next m lines each contain an ID for the participant who sent the message, and a timestamp in the format $HH:MM$, representing the time when the message was sent in the group chat ($1 \leq ID \leq n, 0 \leq HH \leq 23, 0 \leq MM \leq 59$).

Output

For each test case, output a single integer on a new line representing the number of messages that *Aswad* has sent in the group chat.

Examples

standard input	standard output
3 1 1 1 1 00:00 1 10 1 1 23:59 6 5 5 1 01:00 2 02:00 1 03:00 3 03:06 3 03:07	1 1 4
1 4 15 6 1 03:45 3 04:00 4 04:07 1 04:30 3 04:41 2 06:09	3

Note

in the second example: *Aswad* will respond to the third, fifth and the sixth message because the message after each of them was sent exactly after k minutes, so the answers will be 3.

Problem C. What Happens To Bashar's Laptop?

Input file: standard input
Output file: standard output
Time limit: 1 second
Balloon Color: White

We all know how much it's annoying when a virus comes into your computer and start copy and paste folders until you run out of storage. However, there are antivirus which solve this problem.

But sometimes the virus comes to your computer in a human form. That's what happen to sad *Bashar* when he left his laptop with *Aswad*.

To make you empathize with *Bashar*, I will give you instructions to ruin your computer in *Aswad's* style and let you feel what *Bashar* has felt.

You are given the initial folders labeled from 1 to n . For each folder you will be given the labels of folders inside it (of course there is no folder contains it self). The folder number 1 is the Home folder.

You are given also q queries. Each query consists of k pairs (u_i, v_i) where $(1 \leq i \leq k \leq 3)$. And you have to do the following:

During the i - th iteration $(1 \leq i \leq k)$, You have to copy the folder u_i (with all folders inside it) and paste it inside the folder v_i . (see the example for more clarification).

And since I want you to copy a copied folder, you have to assign labels to the newly pasted folders using the following rule: if the original folder is labeled as u , then the pasted node should be labeled as $u + n \times 2^{i-1}$, where i is the iteration number.

After you finish all the iterations, output the number of folders to see the terrible result.

Note that queries are independent.

Input

The first line contains an integer n $(2 \leq n \leq 10^5)$.

The Next n lines describe the contents of the folders:

the i - th line from the n lines start with an integer s_i $(0 \leq s_i \leq 10^5)$ refers to the number of folders inside the i - th folder, then followed by the labels of the folders.

Next line contains an integer q $(1 \leq q \leq 10^5)$.

For the i - th $(1 \leq i \leq q)$ query, first line contains an integer k $(1 \leq k \leq 3)$.

The Next k lines contain the described pairs in the statement. The j - th line contains integers u_j, v_j $(1 \leq u_j, v_j \leq n \times 2^{j-1})$.

It is guaranteed that the input is valid.

Output

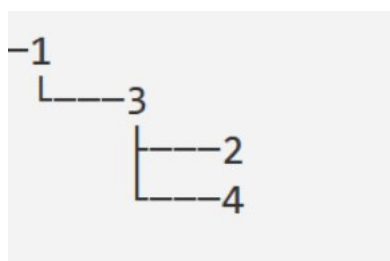
After performing the operations for each query, output a line contains the number of folders.

Example

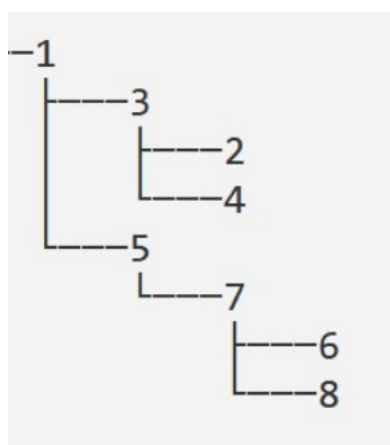
standard input	standard output
4	8
1 3	7
0	
2 2 4	
0	
2	
1	
1 1	
2	
2 2	
2 4	

Note

The initial folder structure is described by the first image.



After applying the first query the final folder will look like:



Note that queries are independent, so for the second query the folder structure will be as described in the first image.

Problem D. Could you help the judges?

Input file: standard input
Output file: standard output
Time limit: 1 second
Balloon Color: Bronze

On June-22-2023, Al-Baath University is supposed to hold a local contest, but the judges are still uncertain whether the problemset is easy or difficult.

The contest consists of n problems, each with a value a_i .

One of the judges, *YouStill-DontKnowMeYet*, suggests determining the difficulty of the contest by taking the *XOR* value of some of the problems values and finding the largest *XOR* value.

However, another judge, *Blade-Master*, argues that determining the contest's difficulty is more challenging than the most challenging problem in the contest. They suggest focusing only on a range of problems from l to r and finding the *XOR* value of those problems.

Later on, the judges realize that the contest is too easy, so they decide to add a problem with k tags to make it more challenging.

The number of tags for a problem refers to the number of ones in its binary representation.

Given the current set of problems and the value of k , your task is to help the judges and determine the *maximum* difficulty that can be achieved by adding a problem that satisfies the given conditions.

In other words, You are given an array a of length n , you have to insert an integer x ($0 \leq x \leq 1023$) to the array a in any position you want so the array becomes in length $n + 1$.

the number x should have exactly k **ones in the binary representation**.

Your task is to find $\max_{1 \leq l \leq r \leq n+1} (a_l \oplus a_{l+1} \oplus \dots \oplus a_r)$

Input

The first line contains an integer t ($1 \leq t \leq 10$) — the number of test cases.

The first line of each test case contains two integers n, k ($1 \leq n \leq 10^5, 0 \leq k \leq 10$).

The second line of each test case contains n integers $a_1, a_2, \dots, a_n$ ($0 \leq a_i \leq 1023$)

Output

For each test case, output one integer: The maximum difficulty that can be achieved by adding a problem that satisfies the given conditions.

Example

standard input	standard output
3	1000
5 2	992
8 5 256 128 37	1019
4 2	
256 512 256 128	
3 2	
3 56 640	

Note

in the first test case we insert the element $x = 576$ in any position and take the *XOR* of all elements in the resulting array so the answer will be 1000. in the third test case we insert

the element $x = 320$ in any position and take the XOR of all elements in the resulting array so the answer will be 1019.

Problem E. What Does Geo Do In His Free Time

Input file: standard input
Output file: standard output
Time limit: 2 seconds
Balloon Color: Red

Suppose you have a collection of n dice, each with k sides, numbered 1 through k .

Or why would you suppose!!!, I will tell you what *Geo-Ghaffar* does in his free time and you imagine with me.

Geo-Ghaffar has a collection of n dice, each with k sides (numbered 1 through k). When he is bored he plays the following game:

At each turn, he throws all the dice, chooses a number x between 1 and k (inclusive) and discards all dice that show x on their top face. He keeps doing this until He has no dice left.

Now *Geo-Ghaffar* is wondering that if the dice are fair, so each possible **arrangement** of numbers on the dice is **equally likely**, What is the expected number of turns it will take to finish the game if he plays optimally. meaning he always choose the number x that maximizes the number of dice that will be removed on that turn.

Since the answer may be large, print it modulo 998244353.

Can you help him find the answer?

Input

The first line contains an integer t ($1 \leq t \leq 5$) — the number of test cases. The descriptions of the test cases follow.

The only line of each testcase contains two integers n, k ($1 \leq k \leq n \leq 700$).

The sum of n over testcases does not exceed 700.

Output

For each test case, output one integer: The expected number of turns it will take to finish the game if *Geo-Ghaffar* plays optimally — modulo 998244353.

Examples

standard input	standard output
2	332748119
2 2	249561090
3 3	
1	332748119
2 2	

Note

Explanation: *Geo-Ghaffar* has two dice, each will show either 1 or 2. The possible arrangements are:

-[1, 1] and he can exclude 1 and take one turn to finish.

-[1, 2] and he can exclude 1, then there will be one die no matter what it will show it takes one turn to be discarded, so this takes two turns to finish.

-[2, 2] and he can exclude 2 and take one turn to finish.

each has probability $\frac{1}{3}$, so the answer is $\frac{1+2+1}{3} = \frac{4}{3}$

Note that [1, 2] and [2, 1] are equivalent.

Problem F. Isn't it a hard problem?

Input file: standard input
Output file: standard output
Time limit: 1 second
Balloon Color: Blue

People think that *Somar* likes palindrome strings, but actually, palindrome strings mean to *Somar* more than his own life. That's why he always searches for them around the world.

The problem is that a palindrome string is a rare element in nature. Therefore, to obtain a palindrome string, you need to extract it from a normal string (which may or may not be a palindrome) through a difficult process. Of course, *Somar* is busy and needs your help to extract palindrome strings.

You will be given an array w consisting of n strings, each string associated with a score s_i .

You can perform the following operation **at most** k times:

- Change any letter in any string to any other letter.

Your goal is to find the **maximum** possible score of a **beautiful** subarray in w , if you can perform the previous operation no more than k times.

- We call a string array a a subarray of w if it can be obtained by removing any number of strings (possibly zero) from the beginning and end of w . Note that a could be an empty array.
- We call a string array a beautiful if each string in a is a palindrome string.
- The score of a string array a is defined as the sum of the scores of each string in a .
- A palindrome string is a string that remains the same when read from both ends. For example, the strings "z", "aaa", "aba", and "abccba" are palindromes, but the strings "icpc" and "ab" are not.

Input

The first line contains an integer t ($1 \leq t \leq 10^5$) — the number of test cases. The descriptions of the test cases follow.

For each test case, the first line contains two integers n and k ($1 \leq n \leq 10^5, 0 \leq k \leq 10^9$).

The next n lines each contain a string w_i ($1 \leq |w_i| \leq 5 \times 10^5$).

The next line contains n integers $s_1, s_2, \dots, s_n$ ($-10^9 \leq s_i \leq 10^9$).

It is guaranteed that the sum of n over all test cases does not exceed 10^5 , and the sum of $|w_i|$ over all test cases does not exceed 5×10^5 .

Output

For each test case, print one integer: the **maximum** possible score of a **beautiful** subarray in w , if you can perform the allowed operation no more than k times.

Example

standard input	standard output
3	18
6 7	0
you	5
still	
dont	
know	
me	
yet	
3 12 -1 -2 9 2	
3 12	
codeforces	
codeblocks	
wronganswer	
-2 -1 -1	
3 2	
fftee	
depee	
treee	
3 4 5	

Note

in the first test case: change the second string to stits. change the third string to toot. change the fourth string to kook. change the fifth string to ee. so the answer will be: $12-1-2+9 = 18$.

Problem G. What is Kaito's delimma?

Input file: standard input
Output file: standard output
Time limit: 1 second
Balloon Color: Gold

KaitoKid is a skilled problem solver, but he's still upset about losing the last ACPC contest to *UnitedBugac*. Now, he's planning to break their alliance with a secret plan that he needs your help with.

KaitoKid has n friends, and he assigned a mysterious value a_i to each one. He wants to take some of his friends with him, but only those who are *close* to each other. specifically, he wants the people who will go with him to have a degree of *closeness* equals to x .

The degree of *closeness* between friends is calculated as the bitwise **AND** of their values. In other words, the degree of *closeness* between friends with values $b_1, b_2, \dots, b_k$ is: $b_1 \& b_2 \& \dots \& b_k$ where $\&$ represents the bitwise **AND** operator.

KaitoKid is in a dilemma: he doesn't know how many friends he should take with him, and he's afraid to go alone. Could you help him figure out the maximum number of friends he can take with him? If no one can go with him, output -1 .

Join *KaitoKid* on his mission and use your problem solving skills to save the day!

Input

The first line contains an integer t ($1 \leq t \leq 5 \times 10^5$) — the number of test cases. The descriptions of the test cases follow.

The first line contains two integers n and x ($1 \leq n \leq 10^5, 0 \leq x \leq 10^8$) — the number of *KaitoKid*'s friends and the desired degree of closeness between the friends who will go with him to complete his mission.

The second line contains n integers $a_1, a_2, \dots, a_n$ ($0 \leq a_i \leq 10^8$) — the mysterious value of *KaitoKid*'s friends.

It is guaranteed that sum of n over all test cases doesn't exceed 5×10^5 .

Output

For each test case, If no one can go with *KaitoKid*, output -1 . Otherwise, output the maximum number of friends he can take with him.

Example

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

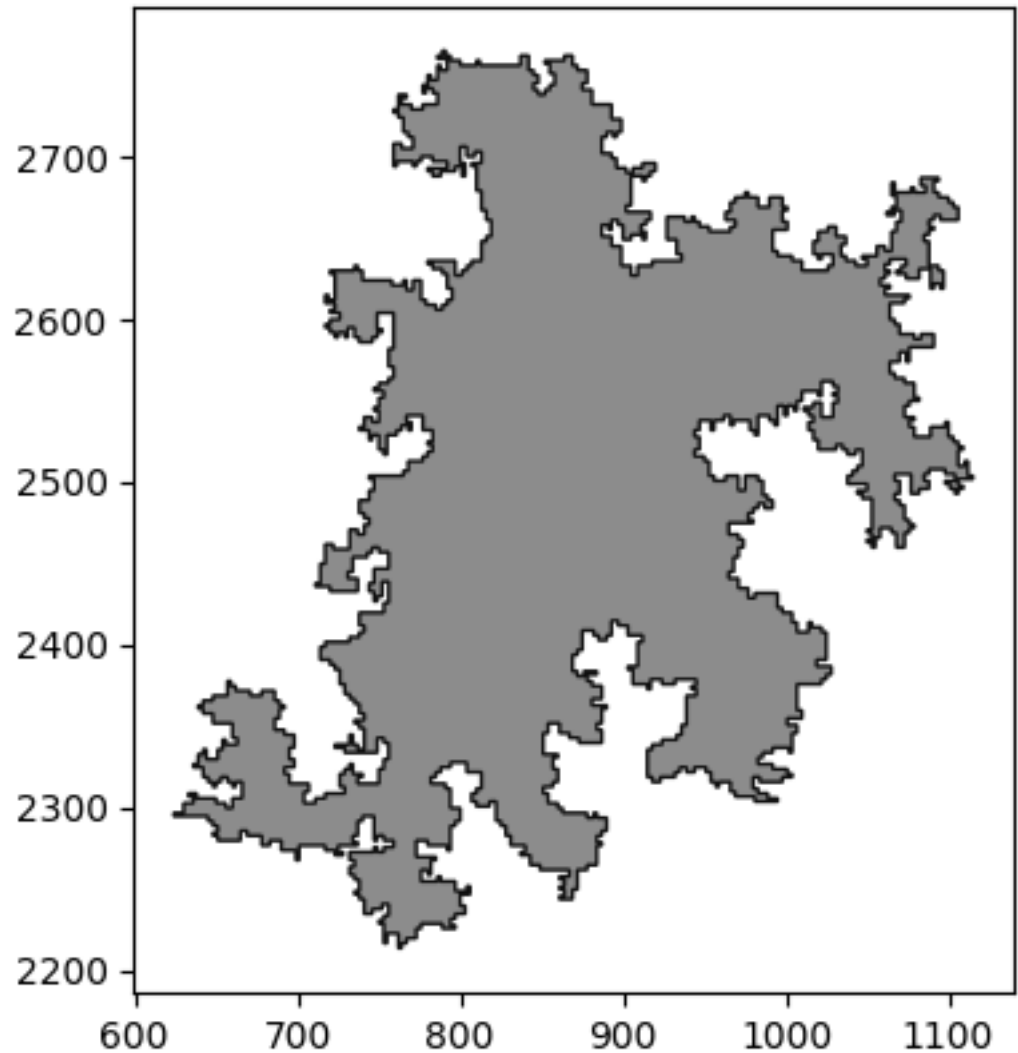
Note

In the first test case, all 6 friends can go because their bitwise AND value is 0, which satisfies the degree of *closeness* requirement.

In the second case, only the first, third and fourth friends can go because $a_1 \& a_3 \& a_4 = 2 \& 7 \& 6 = 2$, which satisfies the degree of *closeness* requirement.

Problem H. Do you love HIAST?

Input file: standard input
Output file: standard output
Time limit: 2 seconds
Balloon Color: Orange



ALBaath University is hosting a competitive programming contest under the ICPC rules, and things were going great until the competitors hit some hard problems. Suddenly, they were in desperate need of some reinforcements, and some teams from other universities agreed to help them out.

But wait! Uh-oh, it turns out that outside help is strictly forbidden in this contest. *HIAST* (*Hilarious Institute for the Advanced Study of Trickery*) heard about this and sent a special team to surround the university. Its mission is to catch any team from outside that might have sneaked in.

HIAST is tricky. It gave you the problem to solve it so It can catch your allies. Now, it's up to you if you want to sacrifice your allies just to get the problem accepted.

You are given a polygon of n vertices and q queries.

Each query consists of a point (x, y) and you have to check if it is inside (including the borders) or outside the polygon.

The points of the polygon are given in clockwise order with the property that either $x_i = x_{i-1}$ or $y_i = y_{i-1}$ (but not both), indicating that the edges are parallel to either the x - or y -axis, Also no two edges intersect (endpoints are not considered into the intersections).

Input

The first line of the input contains an integer n ($4 \leq n \leq 10^5$), representing the number of vertices in the polygon.

The next n lines each contain two integers x_i and y_i ($0 \leq x_i, y_i \leq 10^6$), representing the x - and y -coordinates, respectively, of the i -th vertex of the polygon in clockwise order.

The next line contains an integer q ($1 \leq q \leq 10^5$), representing the number of queries.

The next q lines each contain two integers x and y ($0 \leq x, y \leq 10^6$), representing the x - and y -coordinates, respectively, of a point to be checked whether it is inside or outside the polygon.

Output

Output q lines, where each line contains either "YES" or "NO" (without quotes), indicating whether the corresponding query point is inside or outside the polygon, respectively.

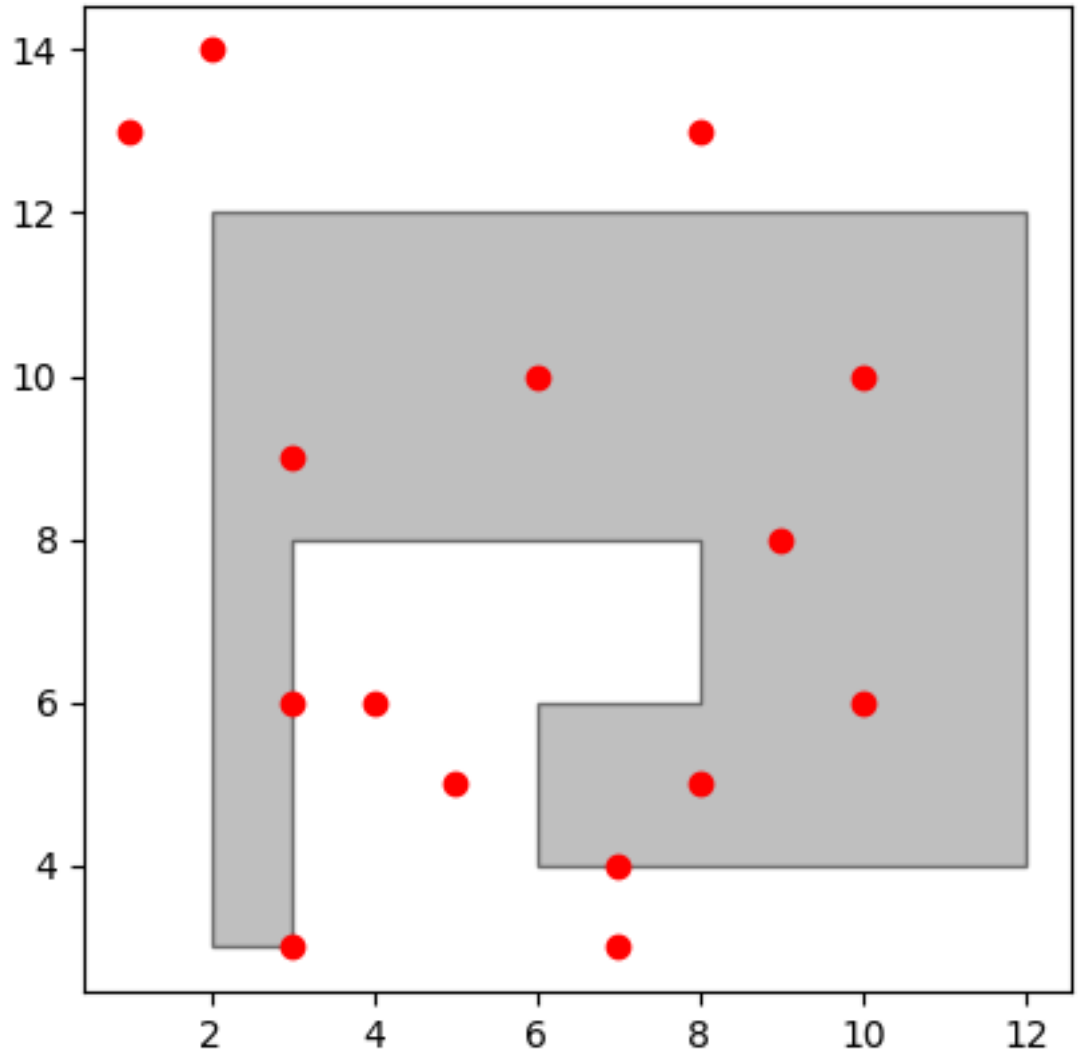
Please Note that the answer is **CaSe SeNsITiVe** so print the answer (YES/NO) with capital letters.

Example

standard input	standard output
10	NO
3 3	NO
3 8	NO
8 8	NO
8 6	YES
6 6	NO
6 4	YES
12 4	YES
12 12	YES
2 12	YES
2 3	YES
15	YES
4 6	NO
5 5	YES
7 3	YES
2 14	
3 3	
8 13	
6 10	
10 10	
3 6	
7 4	
10 6	
9 8	
1 13	
8 5	
3 9	

Note

The polygon and the queries for the first test:



Problem I. Will you accept Basharo challenge?

Input file: standard input
Output file: standard output
Time limit: 3 seconds
Balloon Color: Light Pink

Basharo is too much into counting problems and he challenge you to solve this one.

You'll be given a **connected undirected graph** of n vertices and $n - 1$ edges. The vertices are numbered from 1 to n and the vertex with the number i is colored with the color c_i . The edges are also numbered from 1 to $n - 1$.

The path from a vertex u to a vertex v is called **beautiful** if it satisfies these conditions:

- $u < v$
- $\gcd(c_u, c_v) = 1$. Where $\gcd(x, y)$ is the greatest common divisor of the numbers x and y .

Your task is to count for each edge i the number of beautiful paths that contain this edge.

Input

The first line of the input contains a single integer n ($2 \leq n \leq 5 \times 10^4$) – the number of vertices in the graph.

The second line contains n space separated integers $c_1, c_2, \dots, c_n$ ($1 \leq c_i \leq 3 \times 10^4$). c_i is the color of the vertex number i .

The next $n - 1$ lines contain the edges of the graph. The i^{th} line describes the edge number i . Each line will contain two space-separated integers x and y denoting an edge between vertex number x and vertex number y . ($1 \leq x, y \leq n$)

Output

Print exactly $n - 1$ space-separated integers where i^{th} of them is the answer for the i^{th} edge (The number of beautiful paths that contain this edge).

Examples

standard input	standard output
4 3 2 2 2 1 2 1 3 2 4	2 1 1
7 35 210 14 6 21 10 15 1 2 2 3 2 4 2 5 4 6 4 7	1 1 3 1 1 1

Note

in the second example beautiful paths are: (1,4) , (5,6) , (3,7). we can see that all paths contains the third edge. otherwise all edges will be included only in one beautiful path.

Problem J. How Kifah sees Extreme?

Input file: standard input
Output file: standard output
Time limit: 1.5 seconds
Balloon Color: Brown

In a parallel universe, a contest has been organized by advanced problem setters, and *Kifah* is one of them. The chief judge asked *Kifah* to set an extreme problem. Unfortunately, *Kifah* didn't understand the meaning of "extreme problem" and was too shy to ask for clarification. It's possible that he understood "extreme" in the opposite sense. As the problem is being held in a far universe, no one can catch *Kifah*, and the contestants cannot seek revenge.

Unfortunately for you, there is a bug in the universe's system which directed the problem to your contest. I'm sorry for that, and I hope you enjoy your hard luck.

In this Problem you will be given an array $a_1, a_2, \dots, a_n$.

You can do the following operation any number of times. During each operation you can choose **one** element a_i from the array and do **one** of the following two actions:

- Choose **one** bit from the binary representation of a_i and turn it off.
- Choose **two** bits from the binary representation of a_i and turn them both on (the two chosen bits have to be **both** zeros before turning them on)

What is the minimum number of operations you have to do to make all elements of array a equal.

Input

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

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

The second line contains n integers $a_1, a_2, \dots, a_n$ ($0 \leq a_i \leq 10^5$).

It is guaranteed that the sum of n over all test cases doesn't exceed 10^5 .

Output

For each test case, print one integer — the minimum number of operations you have to do to make all elements of the array equal.

Example

standard input	standard output
1 4 1 2 3 2	4

Note

In the example, the solution could be done in the following order:

$[1, 2, 3, 2] \rightarrow [7, 2, 3, 2] \rightarrow [6, 2, 3, 2] \rightarrow [6, 2, 2, 2] \rightarrow [2, 2, 2, 2]$

Problem K. Do you believe that this is a real story?

Input file: standard input
Output file: standard output
Time limit: 1.5 seconds
Balloon Color: Light Green

On a boring Friday night, *Ramez* and his son *Molham* were sitting together. As usual every Friday, each Dad in the world should ask his son to do something. So, *Ramez* asked his son to solve the following problem:

He gave *Molham* n cards, numbered from 1 to n and arranged in a clockwise circle shape.

Initially, all the cards are **black**. And *Ramez* asked *Molham* to paint $n - 1$ cards with **red** (if it is possible).

Molham can make the painting operation as follows:

He can paint a **black** card (i) with red if and only if there is another **black** card (j) and there is a way to get from i to j passing through **exactly two cards differ than i and j** .

Molham thought the problem is so easy, but *Ramez* did not stop here; if there are many possible orders to paint the $n - 1$ cards red, he asked for the **minimum** lexicographically order of painting.

The coloring order is defined as follows:

If at the k -th operation *Molham* colored the card with number i with red (if it is possible), then the k -th number of the coloring order should be i .

After this extra condition, *Molham* found the problem so hard so he asked for your help.

Input

The first line of the input will contain a single integer ($1 \leq t \leq 10^5$) the number of test cases.

Each of the following t lines will contain a single integer ($4 \leq n \leq 10^5$) the number of cards .

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

Output

For each test case, if it is possible to paint $n - 1$ cards red print "YES" (without quotes) and if not print "NO" (without quotes).

If the first line was "YES" print another line containing $n - 1$ numbers, the minimum lexicographically order of coloring.

Please Note that the answer is **CaSe SeNsiTiVe** so print the answer (YES/NO) with capital letters.

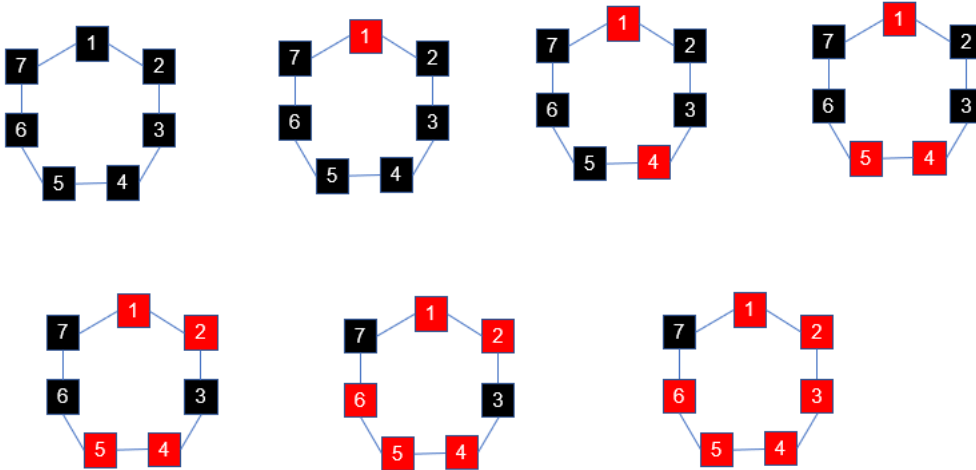
Example

standard input	standard output
2	NO
6	YES
7	1 4 5 2 6 3

Note

when $n = 6$ there is no way to color 5 cards red

when $n = 7$ here is the explanation of the test



first Molham will color the card with number 1 (it is possible because we can get from the black card with number 5 to card with number 1 passing through cards with numbers 6 and 7).

then he will color card with number 4 (using card with number 7).

then he will color card with number 5 (using card with number 2).

then he will color card with number 2 (using card with number 6).

then he will color card with number 6 (using card with number 3).

finally he will color card with number 3 (using card with number 7).

So, the coloring order will be 1, 4, 5, 2, 6, 3.

And the previous order of coloring is the minimum lexicographically between all the possible orders of coloring.

Problem L. Amazing Teacher

Input file: standard input
Output file: standard output
Time limit: 1 second
Balloon Color: Silver

We all dreamed about that teacher who gives us the mark we want. Lately we started dreaming about the teacher who gives us what we deserve at least.

Today is your day, we are giving you the chance to write a code which take the mark of the student and change it to 10 directly then print it.

But wait, we can't give a student with an empty paper a mark of 10. so at least their origin mark should be greater than zero to get 10.

Input

The first line contains a single integer t ($1 \leq t \leq 11$) — the number of test cases.

each test case consists of a single line contains a single integer n ($0 \leq n \leq 10$) — the origin mark of the student.

Output

For each test case, print one integer — print 10 if the origin mark is greater than zero, and print zero otherwise.

Example

standard input	standard output
3	0
0	10
5	10
9	

Problem M. Is it possible?

Input file: standard input
Output file: standard output
Time limit: 1 second
Balloon Color: Pink

You are in front of a hard grid challenge.

There are a grid, a clock, and of course a coin on the grid. The clock starts counting from the moment number 1, then number 2, and so on. At Each moment you can move the coin according to the following rule:

At each moment i you can choose an integer number x_i . Let's assume that the coin is at the coordinate (a, b) at the i th moment. Now if i is **odd**, the coin will jump to the coordinate $(a + x_i, b + x_i)$. But if i is **even**, the coin will jump to the coordinate $(a + x_i, b - x_i)$.

The challenge is to find the minimum number of moments needed to move the coin from the coordinate $(0, 0)$ to the coordinate (n, m) and what is the sequence x_i you have chosen to solve the challenge.

The challenge may be **Impossible**, so in this case print **-1**.

Input

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

At each test case there are two integers n, m ($-10^9 \leq n, m \leq 10^9$), the target coordinate (n, m) .

Output

For each test case, Print **-1** if it's **impossible** to reach (n, m) starting from $(0, 0)$. **Otherwise**, print the minimum number of moments needed to complete the challenge followed by the chosen numbers during the porocess.

Example

standard input	standard output
3	1 3
3 3	-1
2 7	2 -1 -9
-10 8	

Note

in the third test case we can reach the position $(-10, 8)$ in 2 moves. in the first move we choose $x_1 = -1$ so the new coordinate will be $(0-1, 0-1) = (-1, -1)$. in the second move we choose $x_2 = -9$ so the new coordinate will be $((-1)+(-9), (-1)-(-9)) = (-10, 8)$.