Problem A. Magical box

Input file:	`magical.in`
Output file:	`standard output`
Balloon Color:	`White`

Valentina is a big fan of mental games, so for her birthday, her parents gave her a magical box that needs a lot of brain power to be discovered.

The box is made of different layers. In each layer, there is a screen on which two values L and R are shown. To pass from one layer to the next she needs to type the secret number that she has to find after performing bit-wise AND on all the numbers in the range $[L, R]$.

Given the number of layers and the values of L and R on each layer, can you help Valentina finish the game?

Input

The first line contains one integer t ($1 \leq t \leq 10^3$) — the number of test cases.

The only line of each test case contains two numbers L and R ($0 \leq L \leq R \leq 2^{31} - 1$)

Output

For each test case print the answer in a single line.

Example

magical.in	standard output
3 5 6 20 26 11 16	4 16 0

Problem B. Chocolate Lovers

Input file:	`chocolate.in`
Output file:	`standard output`
Balloon Color:	`Purple`

Dina and Donya love chocolate very much. They both always compete for any chocolate they find in their home. Their mom buys them 2 chocolates every day. Dina knows that these 2 pieces of chocolate are not always identical. But Donya doesn't know that. Their mom told them that whoever finishes drinking her milk first will get the chance to open the box and choose the piece of chocolate she prefers.

Dina likes to take the bigger chocolate, obviously. Given the size of each piece of chocolate, help Dina decide if she should drink her milk quickly to be the first to finish or if it doesn't matter.

Input

The first line of the input contains two integers N and M,$(1 \leq N, M \leq 100)$. The size of each chocolate.

Output

If Dina will have to finish her milk first, output "drink drink drink" (without quotations), all in lower case.

Otherwise, print "lazy" (without quotations), all in lower case.

Examples

chocolate.in	standard output
5 10	drink drink drink
7 7	lazy

Problem C. Pattern Farm

Input file:	`farm.in`
Output file:	`standard output`
Balloon Color:	`Gray`

It is almost harvest day and Sara is getting ready to harvest her apple trees. She has exactly n apple trees on her farm arranged in a line, and she knows that the trees will be ready to harvest in the following pattern: The tree at index k will be ready to harvest at minute 1, and any other tree at index $k \pm c$ will be ready to harvest at minute $2 \times c$. So the trees at $k+1$ and $k-1$ will be ready to harvest at minute 2, then the trees at $k+2$ and $k-2$ at minute 4, and so on.

Every minute, Sara can choose a tree and harvest it if it is ready to harvest. However, there is a thief who wants to steal Sara's apples, and he can also choose a tree every minute and harvest it if it is ready to harvest. The thief does not want Sara to see him, so he will never choose to harvest the same tree as Sara at any minute.

Given that both Sara and the thief choose optimally, find the maximum number of trees that Sara will be able to harvest.

Input

The first line contains an integer t ($1 \leq t \leq 10^6$) — the number of test cases. The only line of each test case contains two integers n and k ($1 \leq k \leq n \leq 10^9$) — the number of trees and the index of the first tree which will be ready to harvest.

Output

Print the answer of each test case on a single line.

Example

farm.in	standard output
2 1 1 3 2	1 2

Problem D. Ball on wall

Input file:	`ball.in`
Output file:	`standard output`
Balloon Color:	`Dark green`

Ahlem and Dina are spending their free time by bouncing a ball on walls. In the beginning (time t = 0), the ball is at position A and Ahlem hits it, the ball takes the trajectory shown in the figure and reaches Dina who is at position D. Dina hits it so that it goes in the exact same trajectory but in reverse. It reaches then Ahlem who hits it so that it takes the same trajectory and so on.

The blue trajectory is the trajectory taken by the ball when Ahlem hits it, and red one is the trajectory taken by the ball when Dina hits it.

The ball takes exactly 1 second to travel from any position to its next position, so it takes 1 second to travel from A to B, from B to C, from C to D, from B to A and so on.

Given the number of seconds that have passed since Ahlem first hit the ball at time (t = 0), can you determine the current position of the ball?

Input

The first line of input contains one integer t - the number of test cases ($1 \leq t \leq 1000$)

A single integer n, the time that has passed since Ahlem first struck the ball. $0 \leq n \leq 10^{18}$.

Output

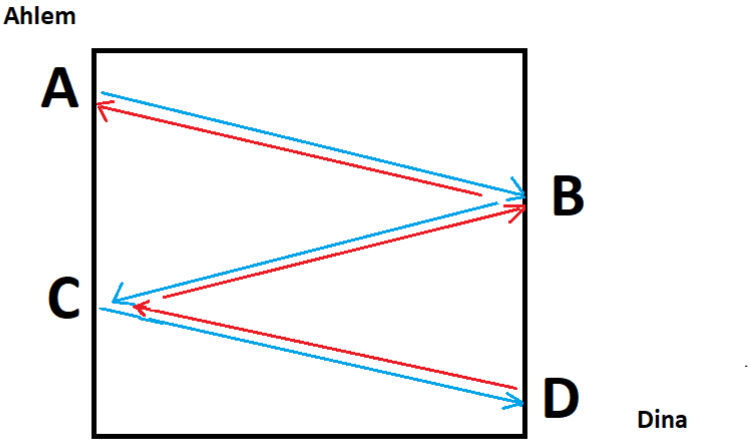
For each of the test cases:

A single letter 'A', 'B', 'C' , or 'D'. The position of the ball at the given time.

Example

ball.in	standard output
1 5	B

Note



For the first test case: at t=0 the ball was in A, at t=1 it hits B, at t=2 it hits C, at t=3 it hits D, at t=4 it goes back to C, at t=5 it hits B!

Problem E. Flowery Country

Input file:	`flowers.in`
Output file:	`standard output`
Balloon Color:	`Orange`

The queen of flowers, Artha, is named that because every city in her country is filled with flower gardens! The cities in Artha's country are connected between each other by edges making the country a Tree rooted in the capital, of course, the capital number is 1! Each city i has A_i gardens in it.

Artha, wants to keep her cities beautiful, so every time she picks up a subtree of the city S, and wants to pull out the weeds from all the gardens in all the cities in this subtree! To do that, she will hire 'beautiful garden' company!

Anaghem is the main director in 'beautiful garden' company, and the number of available workers in her company is X. Anaghem wants to keep her company's rating high, so she only accepts orders, if and only if, X was the gcd for all A_i in the cities she supposed to weed!

Queen Artha knows Anaghem's strategy, every time Artha wants to form an order to weed cities gardens in subtree S, she knows the number of available workers at 'beatiful gardens' company at that time X, She wants her country to stay beautiful, so she changes numbers of gardens A_i in the cities in subtree S, so Anaghem would accept the order to weed these gardens in all cities in subtree S.

Can you help Artha, to tell her what is the minimum amount of cities in which she'll need to change the number of gardens, so Anaghem will accept the order!

Input

The first line of the input contains a single integer number T ($1 \leq T \leq 10^5$). The number of test cases. Each test case is described as follows :

The first line contains one integer n ($1 \leq n \leq 10^5$) The number of the cities in Artha's country.

The second line contains n integers, $A_1, A_2, ..., A_n$ ($1 \leq A_i \leq 10^5$), A_i is the numbers of gardens in city i.

The following $n-1$ lines, each contain two integers u, v representing an edge between city u and city v.

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

Each query will consist of a single line with two integers S, X ($1 \leq S \leq n, 1 \leq X \leq 10^5$). The city of the subtree, and the number of available workers in 'beautiful garden' company.

It's guaranteed that the sum of n over all test cases doesn't exceed 10^5, and the sum of q over all test cases doesn't exceed 10^5

Output

For each test case, print a single integer, the minimum number of cities in which queen Artha should change the number of gardens, so Anaghem would accept the order.

Example

flowers.in	standard output
1 4 10 5 3 6 1 2 2 3 2 4 2 1 2 3 3	2 0

Problem F. Build a Cube

Input file:	`cube.in`
Output file:	`standard output`
Balloon Color:	`Blue`

You want to build a cube with side length equal to x using black and white mini cubes. The volume of the mini cubes is 1. You need x^3 mini cubes to build the desired cube. You can use at most $\lceil \frac{x^3}{2} \rceil$ white mini cubes and the rest should be black. Imagine you've built the cube and you're looking at it from all directions. You want the visible white mini cubes to be as maximum as possible. In another way, you want the white area on the surface of the final cube to be maximized. What is the maximum white area that you can achieve?

Note: $\lceil x \rceil$ means rounding x upward, returning the smallest integral value that is not less than x.

Input

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

Each test case consists of a single line contains an integer $x(1 \leq x \leq 10^5)$denotes the side length of the desired cube.

Output

The maximum white area that you can achieve.

Example

cube.in	standard output
3 2 10 988	12 600 5856864

Note

In the first testcase we need 8 mini cubes and we can use 4 white cubes, whereever you put the cubes you will obtain white area equal to 12.

Problem G. Fill The Buckets

Input file:	`buckets.in`
Output file:	`standard output`
Balloon Color:	`Red`

There are N **Empty** buckets, each bucket i $(1 \leq i \leq N)$ has a_i capacity.

Also, you are given M **full** of water buckets, each bucket j $(1 \leq j \leq M)$ has b_j capacity.

In one operation you can choose any not full bucket i $(1 \leq i \leq N)$ and a full of water bucket j $(1 \leq j \leq M)$, then start filling the bucket i using bucket j until there is no water left in bucket j or the bucket i is full of water then if the bucket j still not empty you move to fill another bucket, and so on until bucket j becomes empty or we have filled all the water bucket.

You are asked to answer Q queries, each query is either:

- 1 j val and it means to change the capacity of b_j $(1 \leq j \leq M)$ to val $(1 \leq val \leq 10^9)$
- 2 and it means to print the **minimum** number of operations needed to make all the empty buckets from 1 to N to be full of water or -1 if it's impossible.

Input

The first line contains a single integer t $(1 \leq t \leq 1000)$ — the number of test cases in the test. Then t test cases follow.

For each test case: In the first line You are given N, M $(1 \leq N, M \leq 3 * 10^5)$ — the number of empty buckets, and full of water buckets.

In the second line you are given N integers $(1 \leq a_i \leq 10^9)$—representing the capacity of i_{th} empty bucket.

In the third line you are given M integers $(1 \leq b_j \leq 10^9)$—representing the intial capacity of j_{th} full of watrer bucket.

In the fourth line You are given Q $(1 \leq Q \leq 3 * 10^5)$ — the number of queries.

Then Q lines follow. The i-th line contains the i-th query in format as in the problem statement.

It's guaranteed that The sum of N, the sum of M, the sum of Q over all test cases won't exceed $3 * 10^5$

Output

For each query of type 2 print the answer in a separate line.

Example

buckets.in	standard output
1 3 4 5 10 7 20 3 4 2 3 2 1 1 8 2	2 -1

Note

For the first Test case, initially the following array a represents how much we need to fill for every bucket $[5, 10, 7]$.

For the first query we can use 2 operations to fill all buckets:

1. we can use the b_0 to fill a_0 and a_1 and part of a_2 so a becomes $[0, 0, 2]$
2. we can use the b_1 to fill the remaining of a_2.

For the second query, b becomes $[8, 3, 4, 2]$.

For the third query, since the total capacity of the empty buckets a $[5 + 10 + 7] = 22 >$ the total capacity of the water buckets b $[8 + 3 + 4 + 2] = 17$. It's impossible to fill all buckets in a using buckets in b.

Problem H. Beautiful Permutations

Input file:	`permutations.in`
Output file:	`standard output`
Balloon Color:	`Cyan`

Amal, Marwa and Abeer were bored, so they decided to write an array A of N numbers (not necessarily distinct) and do some manipulations with it. Amal loves permutations, so she wrote down all the possible **distinct** permutations of the array. Marwa loves prefix sum, so for every permutation P, she wrote under each number P_i the value B_i = the sum of all P_j of the current permutation for all j where $j < i$. However, Abeer loves Beautiful Permutations, so she deleted all the permutations that are not Beautiful.

A Beautiful Permutation is a permutation such that $GCD(P_i, B_i) = 1$ for all $2 \leq i \leq N$.

What is the number of the remaining Permutations (Beautiful Permutations)?

Note: you have to print the answer module 1000000007

Input

The first line contains one integer T ($1 \leq T \leq 20$) number of test cases, each test case starts with one integer N ($2 \leq N \leq 20$) represents the size of the array The second line contains N integers represents the elements of the array ($1 \leq A_i \leq 10^6$).

Output

For each test case print one integer, the number of beautiful permetutaions.

Example

permutations.in	standard output
3 3 3 3 6 4 1 6 2 3 5 9 1 6 3 4	0 2 40

Note

In the first test case there is 3 distinct permutations: $P1$ $[3, 3, 6]$ $P2$ $[3, 6, 3]$ $P3$ $[6, 3, 3]$, wrote by Amal.

The prefix sum of each of this permutations is $B1$ $[0, 3, 6]$ $B2$ $[0, 3, 9]$ $B3$ $[0, 6, 9]$, wrote by Marwa.

The 1st permutation is not beautiful because $GCD(P1_2, B1_2) = GCD(3, 3) = 3 \neq 1$

The 2nd permutation is not beautiful because $GCD(P2_2, B2_2) = GCD(6, 3) = 3 \neq 1$

The 3rd permutation is not beautiful because $GCD(P3_2, B3_2) = GCD(3, 6) = 3 \neq 1$

So Abeer deleted all of 3 permutations because they are all not beautiful. This make the number of remaining permutations = 0.

Problem I. Hanomorphism

Input file:	`hano.in`
Output file:	`standard output`
Balloon Color:	`Gold`

Hano was watching a competitive programming contest. There were N teams participating in the contest. Hano wrote on a piece of paper the standings of the teams at the beginning of the blind hour. When the results were out, Hano took the final standings and wrote them on another piece of paper. He spent some time observing the results. Then he came up with the following definition:

- Hanomorphism: a set of teams with continuous places in both of the standings such that the set of their places before the blind hour is the same as the set of their places in the final standings.

For example: let's say that the standings before the blind hour were like this:

1.A, 2.B, 3.C, 4.D, 5.E

and the final standings were like this:

1.B, 2.C, 3.A, 4.E, 5.D

Then we could say that we have two Hanomorphisms here, the first consists of teams A, B, and C, the set of their places before the blind hour is $\{1, 2, 3\}$ which is the same as the set of their places in the final standings, plus they are in continuous places in both of the standings, from 1 to 3, also teams D and E form another Hanomorphism. So for this example, we have two Hanomorphisms.
We can also say that all 5 teams form one big Hanomorphism because their set of places is $\{1, 2, 3, 4, 5\}$, and they are in continuous places in both of the standings, from 1 to 5.

After understanding what a Hanomorphism is, Hano asked the following question:
Given the blind hour standings and the final standings, what is the maximum number of Hanomorphisms that we can form from them such that each team will be in **one and only one** Hanomorphism?

Note that no two teams will take the same place at any moment of the contest. In other words, there are no ties in the standings. Also, note that team names are numbers from 1 to N.

Input

The first line of the input contains one integer T, the number of test cases.
The first line of each test case contains one integer N,$(1 \leq N \leq 10^5)$ the number of teams who participated in the contest.
The second line of each test case contains N integers $a_1, a_2...a_n$ $(1 \leq a_i \leq N)$, where a_i is the number of the team who is in the $i-th$ place in the blind hour, it's guaranteed that all a_i are distinct.
The third line of each test case contains N integers $b_1, b_2...b_n$ $(1 \leq b_i \leq N)$, where b_i is the number of the team who is in the $i-th$ place in the final standings, it's guaranteed that all b_i are distinct.
Note that the sum of N over all test cases won't exceed 10^6.

Output

For each test case, print one integer M, the maximum number of Hanomorphisms that you can form in the given standings.

Example

hano.in	standard output
4 5 1 2 3 4 5 2 3 1 5 4 3 3 2 1 3 2 1 4 1 2 3 4 4 2 3 1 1 1 1	2 3 1 1

Note

In the third example, we can only form one Hanomorphism with all teams. We couldn't just use the second team as one Hanomorphism and the third team as one Hanomorphism because we can't take first and fourth teams if we did that, and Hano wants each team to be in exactly one Hanomorphism.

Problem J. Playing with portals

Input file:	`portals.in`
Output file:	`standard output`
Balloon Color:	`Yellow`

Given a grid of $N * N$ cells, each cell is either a free cell or a blocked cell. You can move from one cell to any neighbour cell in one step. Two cells are considered neighbours if they share a border.

But there is a feature in that grid that, there are M portals that can make a shortcut between two free cells.

- One portal can start and end in the same cell.
- Portal start at cell (u, v) and end at cell (x, y), can be used only to go from cell (u, v) to cell (x, y) and **not** the opposite.
- The same cell can be the end of many portals but **not** be the start of more than one portal.

There is a special rule you must follow, you can't use more than K portals in your path.

Given that cell $(1, 1)$ and cell (N, N) are always free, can you reach the cell (N, N) from the cell $(1, 1)$ or not? if so, can you tell the shortest path you took!

Input

In the first line, you are given two integers N ($1 \leq N \leq 300$), K ($1 \leq K \leq 30$) the grid size and the maximum number of portals you can use.

The next N lines you will get a String of N character each, the cell either a free cell '.' or a blocked cell '#'.

Then given an integer M ($0 \leq M \leq \min(N * N, 300)$) the number of portals in the grid followed by M lines, each of which will be 4 integers u, v, x, y ($1 \leq u, v, x, y \leq N$) meaning that there is a portal that can move you from cell (u, v) to the cell (x, y) in one move.

Output

If there is a path from cell $(1, 1)$ to (N, N) with using at most K portals print "YES"(without quotations, all in upper case) in a line followed by one integer in the next line which is the shortest path you took (You don't have to minimize the number of portals you use), otherwise print "NO"(without quotations, all in upper case).

Examples

portals.in	standard output
5 3 .##.# ##### ##### ##### ####. 3 1 1 1 4 5 5 5 5 1 4 5 5	YES 2
5 5 ....##. .... #.#.. 19 2 2 2 1 1 2 1 2 1 1 3 1 2 1 3 1 4 1 5 4 5 4 4 5 5 2 5 2 4 3 4 3 3 4 5 4 4 5 3 2 2 5 4 5 3 2 4 5 1 4 2 1 5 5 5 5 3 3 3 1 3 5 3 4 1 3 5 4 3 1 4 5 4 2 5 4	YES 3

Problem K. Lego Contest

Input file:	`lego.in`
Output file:	`standard output`
Balloon Color:	`Light green`

A block building girls contest is taking place in a rectangular hall of dimensions $n \times m$ which consists of 1×1 cells. There is a competitor in each cell, who is trying to establish the tallest building using toy blocks.

The contest is about to end, so the competitors are trying their best. Each competitor, will repeat this operation every minute. She looks at the buildings of her neighboring competitors and increases the height of her own building so that it becomes equal to the maximum of the heights of the buildings of her neighboring competitors. If her building is already taller than the tallest building of his neighboring competitors, then she doesn't do anything in this minute.

Given the current heights of the buildings, can you find how many minutes it take for all the buildings to be equal in height?

Note: two competitors are considered neighboring if their cells share a side or a corner.

Input

The first line contains two integers n and m ($1 \le n, m \le 600$) — the dimensions of the hall.

Then follows n lines containing the current heights of the buildings. ($1 \le h_{ij} \le 10^9$)

Output

Print the answer on a single line.

Examples

lego.in	standard output
2 4 1 4 4 6 8 1 1 3	3
2 2 2 1 1 1	1

Problem L. The Grand Great Wall

Input file:	greatwall.in
Output file:	standard output
Balloon Color:	Pink

You are given a $2d$ grid of width W and height H.

The grid is initially empty and your goal is to fill the entire grid with blocks in such a way that no two blocks can overlap and no block goes out of the grid.

You have three sizes for blocks each available in infinite amount:

1. blocks with dimensions 1 x 1 available in a colors.
2. blocks with dimensions 1 x 2 available in b colors (height 1, width 2).
3. blocks with dimensions 1 x 3 available in c colors (height 1, width 3).

You are **not** allowed to rotate any block.

Count the number of ways to fill the entire grid.

Two ways are considered different if there exists some cell of the grid that is covered by two blocks of different sizes or different colors.

Input

The first line contain T ($1 \leq T \leq 1000$) which is the number of testcases

The first line of each testcase 2 integers W , H ($1 \leq W, H \leq 10^{18}$).

the second line contains 3 integers a , b , c ($1 \leq a, b, c \leq 10^9$).

Output

Print the answar modulo $10^9 + 7$

Example

greatwall.in	standard output
2 3 4 1 2 3 1200 213 4 5 7	4096 227541656