Problem A
A VCS Problem

In software engineering, a 'version control system' (VCS) is responsible for managing changes to a computer program, document or other information. In this problem, you need to implement a VCS for an undirected graph.

Initially, your graph has N vertices and 0 edges. The VCS needs to maintain several **snapshots** of the graph. These **snapshots** are indexed in chronological order by integers starting from 0. An initial **snapshot** of the graph is taken, whose index is 0. Then you need to do Q operations on the graph. Each operation is one of the following types:

- `ADD u v`: Add a new edge between two vertices u and v;
- `CHECK u v`: In the current graph, check if two vertices u and v belong to the same connected component;
- `COMMIT` Create a new **snapshot** of the graph;
- `QUERY s u v` In the s-th **snapshot**, check if two vertices u and v belong to the same connected component.

Input

The first line of the input contains two integers N and Q $(1 \leq N, Q \leq 2 \times 10^5)$. In the next Q lines, each describes an operation in any of the following forms:

- `A u v` $(1 \leq u, v \leq N)$ describes an `ADD` operation,
- `? u v` $(1 \leq u, v \leq N)$ describes a `CHECK` operation,
- `C` describes a `COMMIT` operation,
- `Q s u v` $(1 \leq u, v \leq N, 0 \leq s < S)$ where S is the number of snapshots taken so far, describes a QUERY operation.

Output

For each `CHECK` or `QUERY` operation, print a character 'Y' if two vertices u and v belong to the same connected component, and print 'N' otherwise.

Sample Input 1

```
4 9
A 1 2
? 1 2
Q 0 1 2
C
Q 1 1 2
A 1 3
C
Q 1 1 3
Q 2 1 3
```

Sample Output 1

```
YNYNY
```

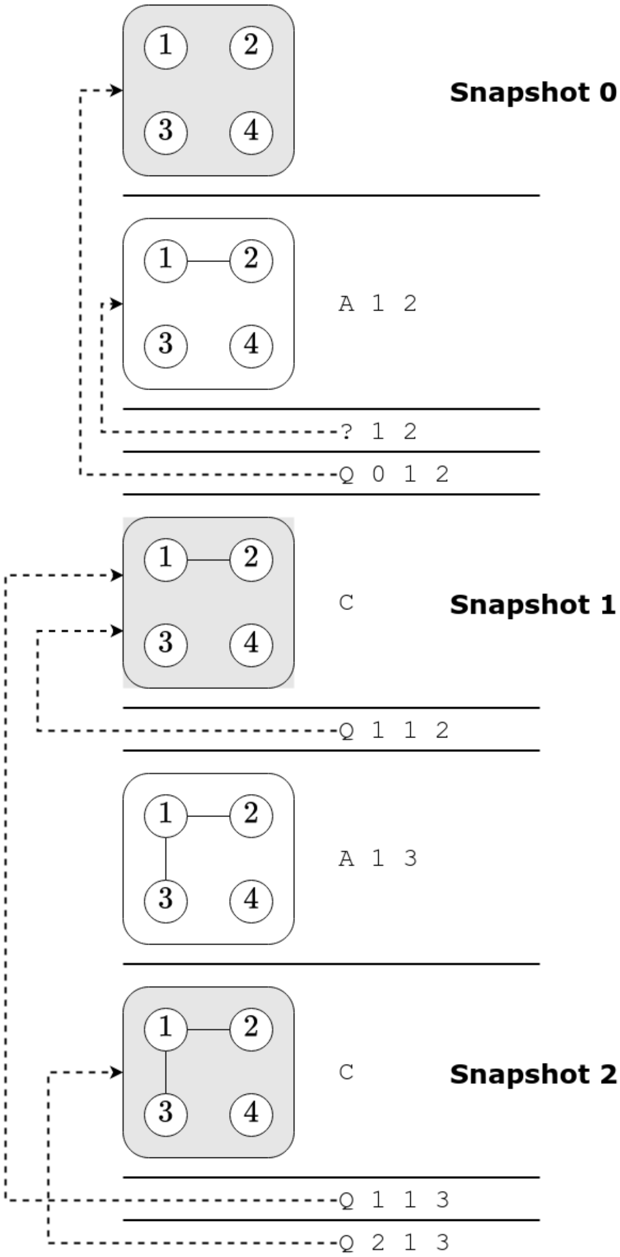


Figure A.1: Explanation of the sample.

- The operations are executed from top to bottom.
- For each `?` and `Q` operation, there is an arrow pointing to the corresponding snapshot of the graph.

Problem B
Binary Strings

Let's define a function f over k binary strings of the same length m (k must be odd):

- The result is a binary string of length m;
- Consider the i^{th} bits of k binary strings:
  - Let c_0 be the number of strings whose i-th bit equal to 0,
  - Let c_1 be the number of strings whose i-th bit equal to 1.
  - If $c_0 > c_1$, the i-th bit of the result is 0,
  - Otherwise, the i-th bit of the result is 1.

An example of function f over 3 binary strings: $f(100, 111, 010) = 110$

You are given a set S of n binary strings, all of which have the same length m, and an odd integer k. Your task is to check if S is k-beautiful or not.

A set is considered k-beautiful iff for any selections of k strings $x_1, x_2, \ldots, x_k$ in S, $f(x_1, x_2, \ldots, x_k)$ is also a string in S. Note that a string from S can be selected **multiple times**.

Input

The first line contains 3 integers n, m, k ($1 \leq k \leq n \leq 300, 1 \leq m \leq 300$, k is odd). Then, n lines follow, each consists of a binary string of length m. ~~It is guaranteed that all n strings are pair-wise distinct~~.

Output

Print `YES` if S is k-beautiful, and `NO` otherwise.

Sample Input 1	Sample Output 1
3 3 3 100 111 010	NO

Sample Input 2	Sample Output 2
4 2 3 00 01 10 11	YES

Problem C
Consecutive primes

You may know that prime number is one of the most amazing concepts in number theory! In the life of every competitive programmer, there must be a huge number of problems about prime numbers, and this is another one!

Recall that prime numbers are positive integers which have exactly two positive divisors. 10 smallest prime numbers are 2, 3, 5, 7, 11, 13, 17, 19, 23, and 29.

In this problem, a positive integer n is called "nice" if and only if it can be represented as product of consecutive prime numbers. More formally, a positive integer n is "nice" if and only if there exists a sequence of integers $p_1, p_2, \ldots, p_k$ such that:

- All numbers $p_1, p_2, \ldots, p_k$ are prime numbers,
- $n = p_1 \cdot p_2 \cdot \ldots \cdot p_k$,
- $p_1 < p_2 < \ldots < p_k$,
- No prime number x exists such that $p_i < x < p_{i+1} (1 \le i < k)$.

According to the above definition, $2 = 2$, $6 = 2 \cdot 3$, $30 = 2 \cdot 3 \cdot 5$ and $210 = 2 \cdot 3 \cdot 5 \cdot 7$ are nice, while $4 = 2^2$, $10 = 2 \cdot 5$ and $20 = 2^2 \cdot 5$ are not.

Given several integers, your task is to determine which ones are nice.

Input

The first line of the input contains an integer t $(1 \le t \le 10^5)$. The next t lines contain t integers $n_1, n_2, \ldots, n_t$ $(1 \le n_i \le 10^{19})$, each is printed in a single line.

Output

Print t words. The i-th one should be `NICE` if n_i is a nice number, or `UGLY` otherwise.

Sample Input 1	Sample Output 1
10 1 2 3 4 5 6 7 8 9 10	UGLY NICE NICE UGLY NICE NICE NICE UGLY UGLY UGLY

Problem D
Decorating Tree

This is an interactive problem.

Thang and Trang plan to decorate their house with a Christmas tree. They want the tree to be a connected graph consisting of n nodes indexed from 1 to n with $n-1$ edges.

As always, Trang is very excited for Christmas, she will first create a tree of m nodes indexed from 1 to m with $m-1$ edges.

After Trang is done, Thang will add $n-m$ remaining nodes and $n-m$ edges, without showing Trang the final tree. It is guaranteed that the final tree is connected.

Thang asks Trang to find the diameter of the tree, i.e. the maximum length of the simple path between any two nodes of the tree.

As a Christmas gift, Thang allows Trang to ask up to `666` questions of the following types:

- `distance u v` - Thang will tell the distance between two vertices u and v on the tree,
- `subtree u v` - Considering the tree rooted at vertex u, Thang will tell the maximum distance from u to any vertex in the subtree of v.

Interaction protocol

- In this problem, your program assumes the role of Trang, and the jury program assumes the role of Thang.
- First, your program must read two integers m and n $(1 \leq m \leq 5 \cdot 10^4, m < n \leq 3 \cdot m)$
- Then, your program must print $m-1$ lines, each line containing two integers u, v $(1 \leq u, v \leq m)$, represents an edge in the tree.
- Then, your program must start asking up to `666` questions as mentioned above
  - `distance u v` $(1 \leq u, v \leq n, u \neq v)$
  - `subtree u v` $(1 \leq u, v \leq n, u \neq v)$
- Finally, your program print out an exclamation mark following with a single integer represents the diameter of the tree.

Sample communication

standard input	standard output
`2 5`	
	`1 2` `distance 1 4`
`3`	
	`distance 1 5`
`3`	
	`subtree 3 2`
`2`	
	`! 3`

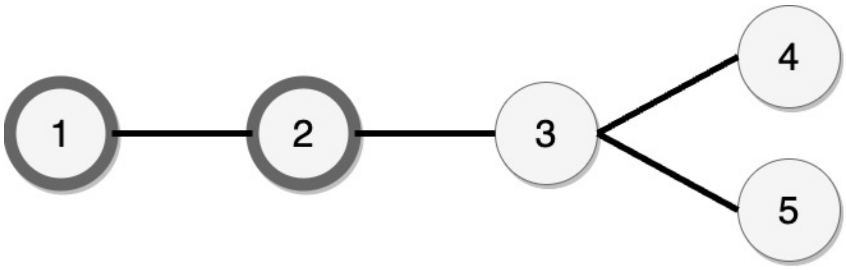


Figure D.1: The final tree of n vertices in the sample. The m initial vertices have thicker border.

Note

When you write the solution for the interactive problem it is important to keep in mind that if you output some data it is possible that this data is first placed to some internal buffer and may be not directly transferred to the interactor. In order to avoid such situation **you have to use special 'flush' operation each time you output some data**. There are these 'flush' operations in standard libraries of almost all languages. For example, in C++ you may use 'fflush(stdout)' or 'cout « flush' (it depends on what do you use for output data — 'scanf/printf' or 'cout'). In Java you can use method 'flush' for output stream, for example, 'System.out.flush()'. In Python you can use 'stdout.flush()'.

Problem E
Estate Construction

Thuc has been playing Age of Empire 2 for several months. In this game, Thuc plays the king's role, dictating his villagers to build estates for the kingdom. The play area can be represented by a board of size $10^9 \times 10^9$, in which the intersection of i-th row and j-th column is denoted by (i, j).

In this game, the resource value of the each cell (i, j) is $(i \times j)^k$, where k is a positive parameter given to the players at the start of the game.

Thuc has already planned to build n estates on the play area to capture those resources. The i-th estate can be represented by a rectangle with coordinates $x_{i,1}, y_{i,1}, x_{i,2}, y_{i,2}$ $(1 \leq x_{i,1} \leq x_{i,2} \leq 10^9, 1 \leq y_{i,1} \leq y_{i,2} \leq 10^9)$, meaning he will capture all cells (x, y) satisfying $x_{i,1} \leq x \leq x_{i,2}$ and $y_{i,1} \leq y \leq y_{i,2}$. There is no restriction of constructing in this game, thus, estates might share some common cells with others.

Unfortunately, Thuc found out today, that he could not capture a cell's resource more than once. Hence, his calculation for the total resource value was no longer correct. Your task is to help Thuc recalculate the actual total resource value he can capture with his plan.

Input

The first line contains two integers n and k $(1 \leq n \leq 2 \cdot 10^5, 1 \leq k \leq 20)$ — the number of estates in Thuc's plan and the parameter k.

On the next n lines, the i-th line contains four integers $x_{i,1}, y_{i,1}, x_{i,2}, y_{i,2}$ representing the rectangle coordinates of the i-th estate in Thuc's plan.

Output

Print an integer denoting the actual total resource value Thuc can capture with his plan in modulo $10^9 + 7$.

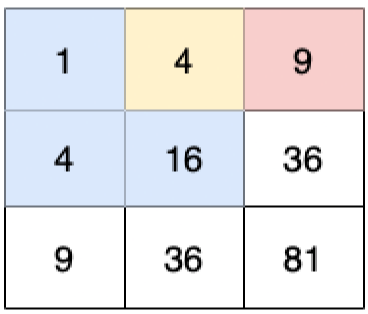


Figure E.1: Occupied cells are highlighted. Two estates in the sample share the cell $(1, 2)$.

Sample Input 1	Sample Output 1
2 2 1 1 2 2 1 2 1 3	34

Problem F
First Name Last Name

Loc is a hard-working security guard at a company. His company only allows Vietnamese people to enter and leave its office, and Loc has to verify everyone's nationality. One day, a man shows up. For safety, Loc politely asks:

- Good day sir. May I see your card?

The man gives out his card. Unfortunately, the card is at a very bad condition. Loc can not see his nationality at all, and can barely see his full name. The card is at a very bad condition, that Loc can only see his name as a string S, containing all of the characters of the name, but there are no spaces. Since Loc is very smart, he quickly comes up with a way to check the man's nationality. As you may know, Vietnamese names always start with the *last name* (the family name), and end with the *first name*. So Loc asked the man:

- What is your first name and your last name, sir?

- My first name is F and my last name is L. I don't have a middle name.

Given the string S, F and L, please help Loc determine if this man has a valid Vietnamese name, so that Loc can let him in.

Input

The first line contains an integer t $(1 \leq t \leq 10\,000)$ — the number of test cases. Then t test cases follow. Each test case is presented in three lines:

- The first line contains the string S — the name of the man without any spaces.
- The second line contains the string F — the first name of the man.
- The third line contains the string L — the last name of the man.

It is guaranteed that each string consists of between 1 and 50 characters, which are lowercase English letter only, and $|F| + |L| = |S|$.

Output

For each test case, output "`YES`" (without quotes) if the man has a valid Vietnamese name, or "`NO`" (without quotes) otherwise.

Explanation of the sample

In the first test case, the man full's name with out spaces is $S =$ "`tranloc`", his *first name* is $F =$ "`loc`", and his *last name* is $L =$ "`tran`". We can see that this is indeed a Vietnamese name, because if we put his *last name* before his *first name*, we will have "`tran loc`", which is the string S when all spaces are removed.

In the second test case, the man's *first name* and *last name* are $F =$ "`albert`" and $L =$ "`einstein`" respectively. If we concatenate his *last name* with his *first name*, we will get "`einstein albert`", which is not the string $S =$ "`alberteinstein`" when all spaces are removed.

Sample Input 1	Sample Output 1
2 tranloc loc tran alberteinstein albert einstein	YES NO

Problem G
GCD query

You are given an array a of n positive integers $a_1, a_2, \ldots, a_n$. There are q queries in the form $\ell\ r\ d$. For each query, you need to find the number of sub-arrays within the range from ℓ to r whose greatest common divisor are less than or equal to d. More formally, you are about to count how many pairs of indices (u, v) such that:

- $\ell \leq u \leq v \leq r$;
- $gcd(a_u, a_{u+1}, \ldots, a_{v-1}, a_v) \leq d$

Input

The first line of input contains 2 integers n and q ($1 \leq n \leq 2 \times 10^5, 1 \leq q \leq 2 \times 10^5$).

The second line contains n positive integers $a_1, a_2, \ldots, a_n$ $(1 \leq a_i \leq 10^9)$ describing the given array.

On the next q lines, each contains 3 integers $\ell\ r\ d$ $(1 \leq \ell \leq r \leq n, 1 \leq d \leq 10^9)$ describing a query.

Output

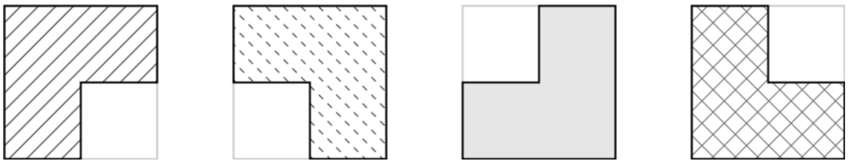
For each query, you need to print the answer in one line.

Sample Input 1	Sample Output 1
6 5 3 9 6 2 8 4 1 5 3 2 4 3 1 5 4 2 6 2 1 6 1	12 4 12 9 6

Problem H
Hallway Tilling

In today's *Indoor Creative Pattern Competition* (ICPC), participating teams need to decorate a long hallway. The hallway floor has a rectangular shape, which can be seen as a $2 \times n$ grid of cells.

Normally, we can use 1×1 or 2×2 tiles to tile the floor. However, to ensure the creative nature of the competition, the ICPC jury requires all teams to use L-shape tiles instead. L-shape tiles are tiles that fit into a 2×2 grid and have one of the following shapes:



Each team can use a number of L-shape tiles to tile the hallway. A team can choose not to use any tiles if they think that the hallway is already beautiful and no further decoration is required. All teams must place the tiles on the floor so that:

- Each L-shape tile covers exactly three cells on the floor.
- No cell on the floor is covered by more than one tiles.

Let's define the *coverage* of a *tiled hallway* as the number of covered cells in the hallway. The ICPC jury wants to make some predictions before the competition, so they would like to know the sum of *coverage* over all *tiled hallways*.

Two *tiled hallways* are considered different, if and only if at least one of the below statements are true:

- There exists a cell which is covered in one *tiled hallways* but is not covered in the other.
- There exists a pair of cells which are covered by the same L-shape tile in one *tiled hallways*, but are covered by two different L-shape tiles in the other.

Given n — the number of columns of the hallway floor. Find the *sum of coverage* over all *tiled hallways* modulo $998\,244\,353$.

Input

The first and only line contains an integer n ($2 \le n \le 10^{18}$) — the number of columns of the hallway floor.

Output

Output the *sum of coverage* over all *tiled hallways* modulo $998\,244\,353$.

Explanation of the samples

- In the first sample, $n = 2$. The hallway floor has size 2×2. There are:
  - 1 *tiled hallway* with 0 coverage (no L-shapes tile are used).
  - 4 *tiled hallways* with 3 coverage (one L-shape tile is used, with 4 choices).

  Therefore, the *sum of coverage* over all *tiled hallways* is $1 \cdot 0 + 4 \cdot 3 = 12$
- In the second sample, $n = 3$. The hallway floor has size 2×3. There are:
  - 1 *tiled hallway* with 0 coverage.
  - 8 *tiled hallways* with 3 coverage.
  - 2 *tiled hallways* with 6 coverage.

  Therefore, the *sum of coverage* among all *tiled hallways* is $1 \cdot 0 + 8 \cdot 3 + 2 \cdot 6 = 36$

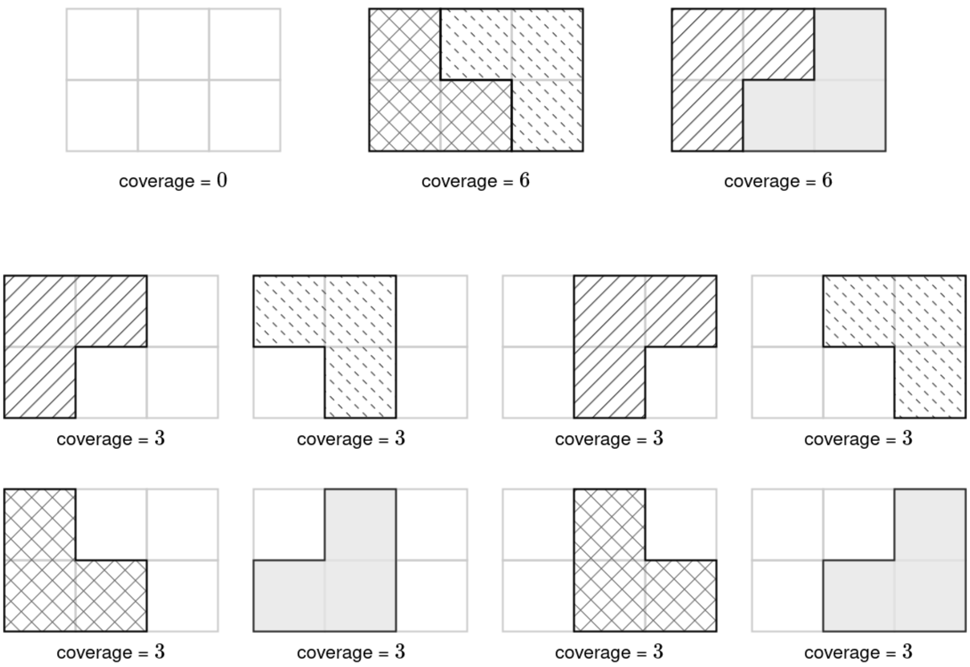


Figure H.1: All *tiled hallways* for $n = 3$ with their respective coverages.

Sample Input 1	Sample Output 1
2	12

Sample Input 2	Sample Output 2
3	36

Sample Input 3	Sample Output 3
177013	912417795

Problem I
Inversion

Permutation is an important and interesting topic in mathematics. One of the most well-known concept relating to permutation is inversion.

To recap, a sequence of integers $p_1, p_2, \ldots, p_n$ is called a **permutation** of integers $1, 2, \ldots, n$ if and only if:

- For every $1 \le i \le n$, $1 \le p_i \le n$.
- For every $1 \le i < j \le n$, $p_i \ne p_j$.

An inversion of a permutation $p_1, p_2, \ldots, p_n$ is a pair (i, j) such that $i < j$ and $p_i > p_j$.

In this problem, there is a secret permutation $p_1, p_2, \ldots, p_n$ of integers $1, 2, \ldots, n$. You are given a two-dimensional array c where $c_{u,v}$ equals the number of inversions of p if we swap p_u and p_v. Your task is to guess this secret permutation.

Input

The first line contains an integer n $(1 \le n \le 1\,000)$ — the length of the permutation.

In the next n lines, the u-th one contains n integers $c_{u,1}, c_{u,2}, \ldots, c_{u,n}$ $\left(0 \le c_{u,v} \le \frac{n \cdot (n-1)}{2}\right)$.

It is guaranteed that there is at least one valid secret permutation.

Output

You should print a single line consists of n integers $p_1, p_2, \ldots, p_n$ representing the secret permutation.

If there are multiple correct permutations, you can output any of them.

Sample Input 1	Sample Output 1
2 0 1 1 0	1 2

Sample Input 2	Sample Output 2
3 2 3 1 3 2 1 1 1 2	2 3 1

Problem J
Jungle Joint

The apes just moved to a rectangle jungle, which can be modelled as a board consisting of n rows and m columns. Its rows are numbered from 1 to n from North to South, and its columns are numbered from 1 to m from West to East. The cell on the r-th row and the c-th column is denoted as (r, c). On this cell, there is a tree of height $h_{r,c}$.

The apes want to choose some cell as their house. To navigate between trees conveniently, they want all trees on the same row or on the same column as their house to have the same height as the tree on their house.

More formally, if the apes choose the cell (i, j) to be their house, the following condition must hold: For every cell (x, y), if $(i - x) \cdot (j - y) = 0$, then $h_{i,j} = h_{x,y}$.

Given the size of the jungle and the heights of all trees, find the number of cells the apes can choose as their house.

Input

The first line contains an integer t — the number of test cases. Then t test cases follow.

The first line of each test case contains two integers n and m ($1 \le n, m \le 1000$) — the size of the rectangle jungle.

On the next n lines, the i-th one contains m integers $h_{i,1}, h_{i,2}, \ldots, h_{i,m}$ ($1 \le h_{i,j} \le 10^9$) — the heights of the trees.

It is guaranteed that the sum of n over all test cases does not exceed $1\,000$, and the sum of m over all test cases does not exceed $1\,000$.

Output

For each test case, print an integer — the number of cells that the apes can choose as their house.

Explanation of the samples

In the first test case, there is only one tree in the jungle. The apes can use that tree, as there is not any other tree in the jungle, so their requirement is still fulfilled.

In the second test case, the apes can choose cells $(2, 1)$ and $(2, 3)$.

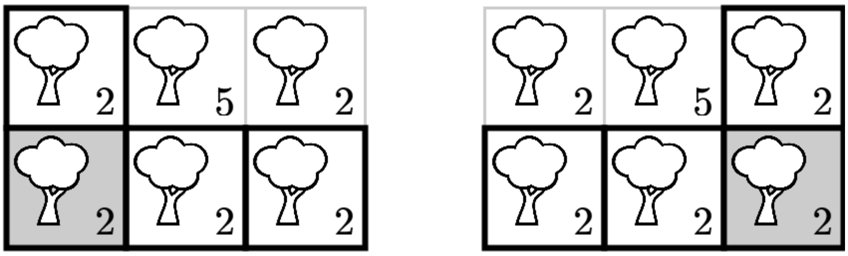


Figure J.1: The chosen cells are colored gray. The cells that has the same row or column with the chosen cell has thicker border.

In the third test case, every tree in the jungle has the same height, so the apes can choose whichever they want.

Sample Input 1	Sample Output 1
3 1 1 10 2 3 2 5 2 2 2 2 3 3 1 1 1 1 1 1 1 1 1	1 2 9

Problem K
Killer Testcase

Thuan is helping his teacher generate testcases for a contest. The hardest problem of this contest is a geometry one where the input is a **convex** polygon. Thuan wants to generate a **convex polygon** such that:

- It has exactly n vertices.
- No three consecutive vertices are collinear.
- All vertices' coordinates are **integers** between -10^6 and 10^6, inclusive.
- The length of all n sides are **positive integers**.

Your task is to help Thuan generate such a killer testcase.

Input

The input consists of only one integer n $(1 \le n \le 1000)$ — the number of vertices of the polygon.

Output

In the first line, you should print `YES` or `NO` indicating whether or not such a polygon exists. If it does, you should print n more lines, each line consists of two integers x_i and y_i $(0 \le |x_i|, |y_i| \le 10^6)$ — the coordinates of the i^{th} vertex. Vertices should be listed in clock-wise order.

If there are multiple valid polygons, you can output any of them.

Sample Input 1	Sample Output 1
3	YES 0 0 0 3 4 0

Sample Input 2	Sample Output 2
4	YES 0 0 0 4 4 4 4 0

Problem L
Land Occupation

In the year 2030, Elon Sumk and Jeff Sozeb landed in the Mars successfully. They both chose the same flat area where they could land their spaceship. The shape of this area can be modelled as a rectangular board, divided into r rows and c columns. The rows are numbered from 1 to r from North to South, the columns are numbered from 1 to c from West to East. The cell in the u^{th} row and v^{th} column is denoted by (u, v).

Elon Sumk landed in (r_E, c_E), Jeff Sozeb landed in (r_J, c_J). After the successful landings, they started to conquer more land. Due to the limited technology, they could only conquer the land inside this rectangular flat area. On Mars, Manhattan distance is used to measure the distance between cells.

Elon Sumk and Jeff Sozeb both agreed to conquer cells of land according to the following rules:

- If a cell is strictly closer to Elon Sumk's landing location than to Jeff Sozeb's, it would belong to Elon Sumk.
- If a cell is strictly closer to Jeff Sozeb's landing location than to Elon Sumk's, it would belong to Jeff Sozeb.
- If the two distances between a cell and two landing locations are equal, no one would conquer it.

In this problem, you are given the size of the area r and c, the number of cells belonging to Elon Sumk E and the number of cells belonging to Jeff Sozeb J, your task is to identify their landing locations.

Recall that the Manhattan distance between two cells (x_1, y_1) and (x_2, y_2) is $|x_1 - x_2| + |y_1 - y_2|$.

Input

The first line contains an integer t — the number of test cases. Then t test cases follow.

Each test case is presented in one line, with four positive integers r, c, E and J ($1 \leq r, c \leq 250$, $2 \leq E + J \leq r \cdot c$).

It is guaranteed that the sum of $r \cdot c$ over all test cases does not exceed $625\,000$.

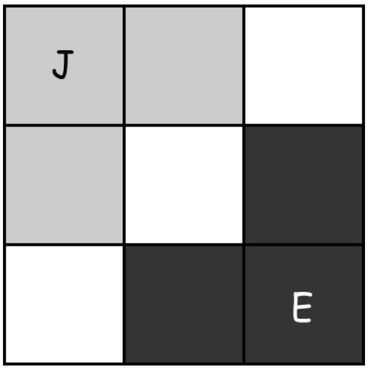
Output

For each test case, you should print on one line four integers r_E, c_E, r_J and c_J representing the landing locations of Elon Sumk and Jeff Sozeb, or `-1 -1 -1 -1` if there is no solution.

If there are multiple solutions, you can output any of them.

Explanation of the sample

The picture below shows the first sample. Cells occupied by Elon Sumk are colored gray, cells occupied by Jeff Sozeb are colored black. White cells are unoccupied.



Sample Input 1

```
1
3 3 3 3
```

Sample Output 1

```
3 3 1 1
```

Problem M
Median of Xor Sequence

Given four non-negative integers a, b, c and d; let S be the "multiset" containing all values $z = x \oplus y$ of all pairs of integers (x, y) such that $a \le x \le b$ and $c \le y \le d$. Your task is to find the median value of S.

Please note that S is a "multiset". In other words, if there are several pairs (x, y) with the same value of $x \oplus y$, this value appears multiple times in S.

For example, consider $a = 3$, $b = 5$, $c = 6$ and $d = 9$. We have:

- $3 \oplus 6 = 5$, $3 \oplus 7 = 4$, $3 \oplus 8 = 11$, $3 \oplus 9 = 10$
- $4 \oplus 6 = 2$, $4 \oplus 7 = 3$, $4 \oplus 8 = 12$, $4 \oplus 9 = 13$
- $5 \oplus 6 = 3$, $5 \oplus 7 = 2$, $5 \oplus 8 = 13$, $5 \oplus 9 = 12$

Hence, 12 elements of S, in increasing order, are 2, 2, 3, 3, 4, 5, 10, 11, 12, 12, 13, 13; meaning that the median of S, the sixth element, is 5.

A bitwise XOR (denoted as $\oplus$) is a binary operation that takes two bit patterns of equal length and performs the logical exclusive OR operation on each pair of corresponding bits. The result in each position is 1 if and only if two bits are different, and is 0 if two bits are equal. For example:

- $3 \oplus 6 = 011_2 \oplus 110_2 = 101_2 = 5$
- $4 \oplus 7 = 100_2 \oplus 111_2 = 011_2 = 3$
- $5 \oplus 8 = 0101_2 \oplus 1000_2 = 1101_2 = 13$

The median value of a sequence of numbers in increasing order $v_1 \le v_2 \le \ldots \le v_n$ is $v_{\frac{n}{2}}$ if n is even and $v_{\frac{n+1}{2}}$ if n is odd.

Input

The first line of the input contain an integer t $(1 \le t \le 75)$ — the number of test cases.

In the last t lines, each contains four integers a, b, c, d $(0 \le a, b, c, d < 10^{200}, a \le b, c \le d)$ representing a test case. All numbers are in decimal form.

Output

For each test case, write a single integer on a single line denoting the median value of S. All numbers should be in decimal form.

Sample Input 1	Sample Output 1
2 3 5 6 9 11 13 20 22	5 26