Problem A. Array

Time limit: 4 seconds

Yukikaze received an array $(a_1, a_2, \cdots a_n)$ as a gift. She decided to play with it. The game consists of q turns. In each turn, she will perform some kind of operation (listed below) on all elements in a subarray of a.

1. Given l, r, k, for every element a_i in subarray $(a_l, a_{l+1}, \cdots a_r)$, replace a_i by $a_i + k$.
2. Given l, r, k, for every element a_i in subarray $(a_l, a_{l+1}, \cdots a_r)$, replace a_i by $a_i \times k$.
3. Given l, r, k, for every element a_i in subarray $(a_l, a_{l+1}, \cdots a_r)$, replace a_i by a_i^k.
4. Given l, r, k, find the sum of the k-th powers of elements in the subarray $(a_l, a_{l+1}, \cdots a_r)$. In other words, find the value of $\sum_{i=l}^{r} a_i^k$.
5. Given l, r, find the product of all elements in the subarray $(a_l, a_{l+1}, \cdots a_r)$. In other words, find the value of $\prod_{i=l}^{r} a_i$.

In this problem, we define that $0^0 = 1$.

Since the result of operations of the last two kinds may be large, you only have to find it modulo a small integer p.

Input

The first line of the input contains two integers n, p $(1 \le n \le 10^5, 2 \le p \le 30)$, denoting the number of elements in the array a and the modulus for the operations of type 4 and 5.

The next line contains n integers $a_1, a_2, \cdots, a_n$ $(0 \le a_i \le 10^9)$, denoting the initial elements in the array a.

The third line contains one integer q $(1 \le q \le 10^5)$, denoting the number of operations to be performed.

Each of the following q lines contains 4 integers t, l, r, k $(1 \le t \le 5, 1 \le l \le r \le n, 0 \le k \le 10^9)$ represents an operation of kind t. Note that if $t = 5$, it's guaranteed that $k = 0$.

Output

For each operation of type 4 or 5, output the result modulo p as an integer in a single line.

Sample Input 1

```
5 29
5 2 4 1 3
9
4 2 4 1
1 1 3 2
2 2 4 3
3 3 5 2
5 3 3 0
4 1 5 2
2 3 5 0
3 2 4 0
4 3 4 1
```

Sample Output 1

```
7
5
3
2
```

This page is intentionally left blank.

Problem B. Building Blocks

Time limit: 1 second

You are playing with building blocks and have piled up n castles in a row, numbered $1, 2, \ldots, n$ from left to right. The height of the i-th castle is h_i. Now you can perform each of the following two operations any times you want:

- Choose an integer i ($1 \le i \le n$) and put some building blocks on the i-th castle to increase its height by 1, which will cost you a_i seconds.
- Choose an integer i ($1 \le i \le n$) and remove some building blocks from the i-th castle to decrease its height by 1, which will cost you d_i seconds. Note that the height of a castle can not be negative.

Your goal is to make the sequence $h_1, h_2, \ldots, h_n$ non-decreasing (i.e., $h_i \le h_{i+1}$ for all $1 \le i < n$). Please calculate the minimum total seconds you need.

Input

The first line contains an integer T ($1 \le T \le 10^5$) — the number of test cases.

For each test case:

- The first line contains an interger n ($1 \le n \le 10^5$) — the number of castles.
- The second line contains n integers $h_1, h_2, \ldots, h_n$ ($1 \le h_i \le 10^8$) — the height of each castle.
- The third line contains n integers $a_1, a_2, \ldots, a_n$ ($1 \le a_i \le 10^5$) — the seconds you need to increase the height of the i-th castle by 1.
- The fourth line contains n integers $d_1, d_2, \ldots, d_n$ ($1 \le d_i \le 10^5$) — the seconds you need to decrease the height of the i-th castle by 1.

It's guaranteed that the sum of n among all test cases will not exceed 10^5.

Output

For each test case, print the minimum seconds you need in a separate line.

Sample Input 1	Sample Output 1
2 3 3 2 1 3 2 1 1 2 3 10 14 3 4 1 7 18 11 3 8 3 18 19 20 3 17 8 14 18 19 8 7 12 20 5 10 16 17 6 20 8	2 427

This page is intentionally left blank.

Problem C. Cats

Time limit: 1 second

Little E has n cathouses in a line. For each integer height between 1 and 20, there are sufficiently many cats with that height. He should choose n cats to place in these cathouses. Every cathouse should contain exactly one cat. However, these cats have a special habit. For every two different cats with the same height, they can't bear to live adjacently, and they can't bear the height of the shortest cat living in the cathouse between them is greater than or equal to theirs.

It's too hard for Little E to find a scheme to make all the cats living in cathouses satisfied. Can you help him?

Input

The first line contains an integer n $(1 \le n \le 10^5)$ — the number of cathouses.

Output

Output n integers in a line. For the i-th integer, output the height of the cat living in the i-th cathouse. Any scheme which can make all the cats living in cathouses satisfied is acceptable.

Sample Input 1	Sample Output 1
1	1

Sample Input 2	Sample Output 2
3	1 2 3

This page is intentionally left blank.

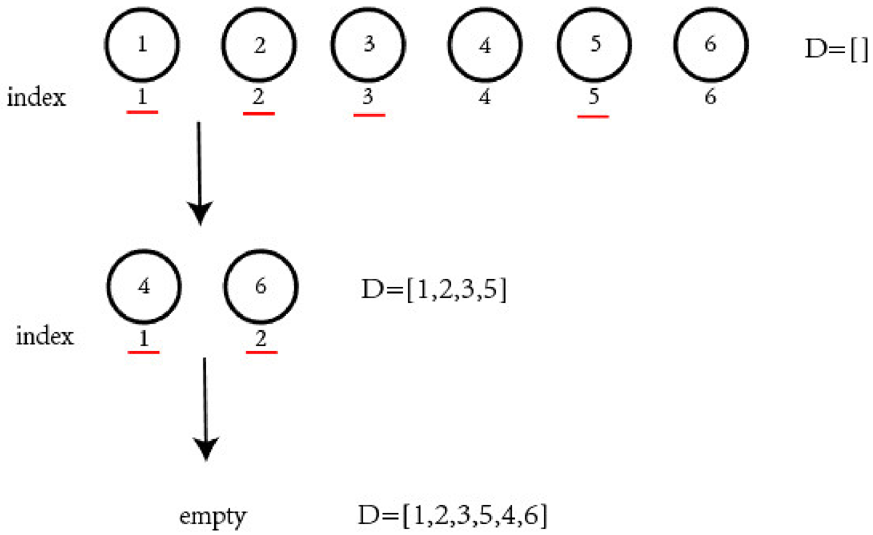
Problem D. Delete Prime

Time limit: 2 seconds

There are n balls lie in a line. Each ball has a number on it, and the number on the i-th ball is i $(1 \le i \le n)$.

For each round, we choose the balls **whose index is** 1 **or a prime number** and kick them out from the line in order. After that, we start the next round with the remaining balls while their relative position does not change and their indices are relabeled in order from 1 (but the numbers on the balls will not change). We repeat this process until no ball is left. There is a kick-out sequence D, which is empty in the very beginning. Every time a ball is kicked out, the number on it will be appended to D.

For example, when $n = 6$ we have balls $[1, 2, 3, 4, 5, 6]$ lie in a line. In the first round, we kick out balls $[1, 2, 3, 5]$ (index 1, 2, 3, 5) in order and the remaining balls are $[4, 6]$. In the second round, we kick out $[4, 6]$ (index 1, 2) in order and there is no ball left. So the kicked-out sequence is $D = [1, 2, 3, 5, 4, 6]$. Note that the index of D always starts from 1.



Now you need to answer two types of questions:

- **Type 1** Given n and k, what is the index of k in the kick-out sequence? In other words, find x such that $D[x] = k$.
- **Type 2** Given n and k, what is the k-th number in the kick-out sequence? In other words, find x such that $D[k] = x$.

Input

The first line of the input contains an integer T $(1 \le T \le 2 \cdot 10^5)$, denoting the number of queries.

For the following T lines, each line contains three integers t, n, k $(t \in \{1, 2\}, 1 \le k \le n \le 10^6)$, representing a query of type t mentioned above.

Output

For each query, output the answer in a single line.

Sample Input 1	Sample Output 1
10 1 5 1 1 5 2 1 5 3 1 5 4 1 5 5 2 5 1 2 5 2 2 5 3 2 5 4 2 5 5	1 2 3 5 4 1 2 3 5 4

Problem E. Eliminate the Virus

Time limit: 1 second

Servers in the Biscotti kingdom network has been attacked by a computer virus! Yukikaze, the Administrator, is trying to eliminate the virus using a specialized anti-virus program.

The network of the Biscotti kingdom is formed by some nodes and some bidirectional links between the nodes. Yukikaze knows that at the end of each second, the virus will transmit its clone from a node to the neighbors of the node, then destroy itself in the origin node. The neighbors of a node are nodes directly connected to it.

Yukikaze needs to design a strategy for the program, then the program will follow the strategy to find and destroy the virus. A strategy consists of a sequence of (a finite number of) steps. In each step, the anti-virus program will clean up several nodes in the network. At the beginning of each second, a step will be executed by the program. Steps are executed sequentially (step by step, from the first step to the last step). When every step in the designated strategy is executed at least once, the program halts. If the virus doesn't exist in the network anymore after the program halts, then the strategy is considered valid.

Unfortunately, the ability of the anti-virus program is limited by a parameter k, denoting the maximum number of nodes the program can affect in each step. Before the anti-virus program launches, every node in the network was infected by the virus. Design a strategy to eliminate the virus from every node in the network or declare it's impossible.

Input

The first line contains three integers n $(2 \le n \le 16)$, m $(1 \le m \le \frac{1}{2}n(n-1))$, k $(1 \le k \le n)$, denoting the number of nodes in the network, the number of distinct links in the network and the parameter of anti-virus program.

For the next m lines, each line contains two integers u, v $(1 \le u, v \le n, u \ne v)$, denoting an bidirectional link between node u and v.

It's guaranteed that there will be no isolated node in the network (i.e., a node which is not connected to any other node) and no pair of nodes is connected by more than one link.

Output

If the virus won't be eliminated from the network in any possible strategy, output -1 in a single line.

Otherwise, output the strategy in the following format.

The first line of the output should contain a single integer L $(1 \le L \le 100)$, denoting the number of steps.

Each of the following L lines should contain a non-empty string consisting of lowercase letters which length is less or equal to k, denoting the nodes to clean up in each step. If the i-th lower case letter is presented in the $(j+1)$-th line of the output, then the i-th node will be cleaned up in the j-th step.

It's guaranteed that for every input test case if there exists a valid strategy, then there exists a valid strategy using less than or equal to 100 steps.

Any valid strategy satisfying the constraint $1 \le L \le 100$ is acceptable.

Sample Input 1	Sample Output 1
5 4 2 1 2 2 3 3 4 2 5	2 bd bd

Sample Input 2	Sample Output 2
2 1 1 1 2	2 b b

Sample Input 3	Sample Output 3
5 5 2 1 2 2 3 3 4 4 5 5 1	4 ce be de ac

Note

Try to consider the strategy on a 4×3 grid graph.

Problem F. Flee from Maze

Time limit: 10 seconds

Little Rabbit is trapped in a maze, which can be represented as a grid with n rows and m columns. Some blocks of the maze are empty lands, and some are walls. Little Rabbit can only step on empty lands. He cannot step on walls or outside the maze.

Not knowing where to go, Little Rabbit decides to follow a fixed pattern. The pattern is given as a string s of length l, which contains only `L`, `R`, `D` and `U`. We use (x, y) to represent a block in the x-th row and the y-th column. The meanings of `L`, `R`, `D` and `U` are as follows.

- `L`: Move from (x, y) to $(x, y - 1)$.
- `R`: Move from (x, y) to $(x, y + 1)$.
- `D`: Move from (x, y) to $(x + 1, y)$.
- `U`: Move from (x, y) to $(x - 1, y)$.

Given a certain maze and a certain pattern s, there are q queries. In each query, $x_1, y_1, z, x_2, y_2, a, b$ are given. In the beginning, Little Rabbit is located at (x_1, y_1). Little Rabbit will start from s_z, and follow the pattern cyclically and infinitely as $s_z, s_{z+1}, \cdots, s_l, s_1, s_2, \cdots, s_l, s_1, s_2, \cdots$. If a movement is invalid (such as moving into a wall or outside the maze), he will just stay at the current position, which is also considered as a step. You need to tell Little Rabbit the number of integers u in $[a, b]$, such that he is located at (x_2, y_2) after the u-th step. As a special case, it is considered that he is located at (x_1, y_1) after the 0-th step.

Input

The first line of the input contains an integer T $(1 \le T \le 10)$ — the number of test cases.

The first line of each test case contains three integers n, m, q $(1 \le n, m \le 30, 1 \le q \le 3 \cdot 10^5)$ — the number of rows, columns and queries.

Then in the next n lines, each line contains a string of length m, which contains only `.` and `*`. If the y-th character of the x-th string is `*`, the block (x, y) is a wall. Otherwise, the block (x, y) is an empty land.

Then the next line contains a string of length l $(1 \le l \le 5000)$, which contains only `L`, `R`, `D` and `U`, indicating the fixed pattern.

Then in the next q lines, each line contains seven integers $x_1, y_1, z, x_2, y_2, a, b$ $(1 \le x_1, x_2 \le n, 1 \le y_1, y_2 \le m, 1 \le z \le l, 0 \le a \le b \le 10^9)$, indicating a query.

Output

The output of the x-th test case begins with `Case #x:` in a single line.

Then in the next q lines, the i-th line contains an integer indicating the answer of the i-th query.

Sample Input 1

```
1
3 3 3
...
.*.
...
RRDDLLUU
1 1 1 3 3 1 1000
1 1 1 3 3 4 4
3 3 1 1 1 1 1000
```

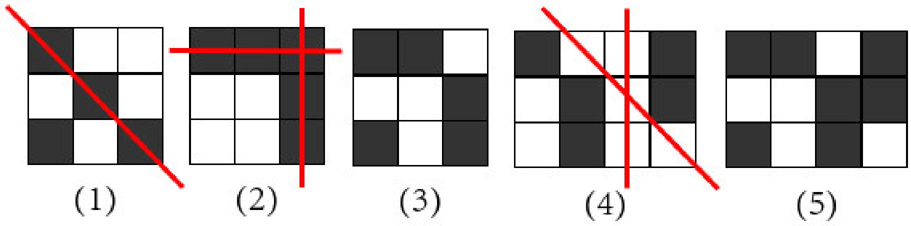
Sample Output 1

```
Case #1:
125
1
125
```

Problem G. Grid Coloring

Time limit: 1 second

Bob likes coloring in grids. Today he wants to color every cell in a $n \cdot m$ grid black or white. Bob thinks a coloring way is beautiful if there are no three consecutive cells of the same color in horizontal, vertical, or diagonal directions. Here are some examples to help you understand.



(1), (2), (4) are not beautiful, and (3), (5) are beautiful.

Now here comes the question: given n and m, can you tell Bob how many beautiful coloring ways are there? Let x be the answer, you just need to output $x \bmod 1\,000\,000\,007$.

Input

The first line of the input contains an integer T $(1 \le T \le 20)$, denoting the number of queries.

Each of the following T lines contains two integers n, m $(1 \le n, m \le 10^9)$, representing a query.

Output

For each query, output the answer in a single line.

Sample Input 1	Sample Output 1
6 1 1 2 2 2 3 3 3 3 4 4 4	2 16 36 32 44 18

This page is intentionally left blank.

Problem H. Happy Morse Code

Time limit: 1 second

Little Rabbit and Little Horse recently learned about Morse code and found that just only two symbols of dash and dot can express countless words, for that each letter has a unique dash-dot string correspondence. Little Rabbit and Little Horse get the wrong conclusion because a dash-dot string does not necessarily correspond to a letter, it may also correspond to a number, or it may correspond to a sentence indicating start, interrupt, and end.

Anyway, they plan to use 0 to represent a dot, 1 to represent dash, and binary strings to express their own happy Morse code. They randomly assign a binary string for each letter and made it into a cipher book.

Given a binary string, can Little Rabbit and Little Horse's cipher book uniquely interpret the meaning of the string? If it is, output `happymorsecode`; if there is more than one possible interpretation, output `puppymousecat` and the number of feasible interpretations modulo 128; if it can't be interpreted at all, output `nonono`.

Input

The input contains several test cases.

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

For each test case: the first line contains two integers n $(1 \le n \le 10^5)$ and m $(1 \le m \le 26)$, indicating the length of the given binary string s and the number of letters in the cipher book. For the following m lines, each line contains a unique letter and its binary string correspondence t $(1 \le |t| \le 5)$, where $|t|$ denotes the length of string t. The last line contains the given binary string s.

It is guaranteed that the sum of n will not exceed 10^5.

Output

For each test case, output following content in a line: if the cipher book can interpret the unique meaning of the string, output `happymorsecode`; if the book can interpret more than one meanings of the string, output `puppymousecat` and the number of feasible interpretation modulo 128; if the string can't be interpreted at all, output `nonono`.

Sample Input 1	Sample Output 1
3 4 2 a 01 b 10 0110 4 4 a 01 b 10 c 01 d 0110 0110 4 2 a 1 b 10 0110	happymorsecode puppymousecat 3 nonono

This page is intentionally left blank.

Problem I. Intersections

Time limit: 1 second

In the city, there are n rows and m columns totaling $n \cdot m$ intersections, and the intersection of row i, column j has two properties $a_{i,j}$, $b_{i,j}$. We may use a pair of integers (i, j) to denote the intersection of row i and column j.

When the pedestrian is at intersection (i, j), for any non-negative integer k:

- If the current time is in $[k \cdot a_{i,j} + k \cdot b_{i,j}, (k+1) \cdot a_{i,j} + k \cdot b_{i,j})$, the pedestrian can choose to walk to intersection $(i-1, j)$ if $i > 1$, or intersection $(i+1, j)$ if $i < n$.
- If the current time is in $[(k+1) \cdot a_{i,j} + k \cdot b_{i,j}, (k+1) \cdot a_{i,j} + (k+1) \cdot b_{i,j})$, the pedestrian can choose to walk to intersection $(i, j-1)$ if $j > 1$, or intersection $(i, j+1)$ if $j < m$.

You can choose to remain stationary in place. It takes $c_{i,j}$ time to walk between (i, j) and $(i, j+1)$ in either direction, and $w_{i,j}$ time to walk between (i, j) and $(i+1, j)$ in either direction. It takes no time to pass through the intersection.

At the moment 0, you are at intersection (x_s, y_s) and you want to go to intersection (x_t, y_t). What is the minimum amount of time it will take?

Input

The first line of six positive integers n, m, x_s, y_s, x_t, y_t ($2 \le n, m \le 500, 1 \le x_s, x_t \le n, 1 \le y_s, y_t \le m$), with the meanings described in the problem statement.

For the next n lines, each contains m positive integers. The j-th number in line i represents the property $a_{i,j}$ ($1 \le a_{i,j} \le 10^9$) of intersection (i, j).

For the next n lines, each contains m positive integers. The j-th number in line i represents the property $b_{i,j}$ ($1 \le b_{i,j} \le 10^9$) of intersection (i, j).

For the next n lines, each contains $m-1$ positive integers. The j-th number in line i represents the road length $c_{i,j}$ ($1 \le c_{i,j} \le 10^9$) between intersection (i, j) and intersection $(i, j+1)$.

For the next $n-1$ lines, each contains m positive integers. The j-th number in line i represents the road length $w_{i,j}$ ($1 \le w_{i,j} \le 10^9$) between intersection (i, j) and intersection $(i+1, j)$.

Output

Output one integer in a line, representing the answer.

Sample Input 1	**Sample Output 1**
5 5 1 1 5 1 5 3 3 3 3 1 5 4 5 5 2 1 4 3 4 5 2 4 1 2 2 4 5 2 3 2 2 5 1 5 4 1 4 5 3 3 5 5 1 5 3 3 2 2 4 3 2 2 2 5 8 2 9 7 1 5 4 7 2 6 10 8 3 10 2 10 8 7 9 9 9 6 2 1 1 2 8 4 6 4 10 4 1 6 5 8 8 4 10 4	33

Problem J. Just Multiplicative Inverse

Time limit: 1 second

Yukikaze is learning number theory. She found a mysterious function to compute multiplicative inverse of integers modulo a prime p. The pseudocode of the function is as follows:

```
function F(x, p)
    if x ≤ 1 then
        return 1
    else
        return −⌊p/x⌋ · F(p mod x, p) mod p
    end if
end function
```

She wants to know the expected number of calls if she calls this function with a random integer uniformly distributed in the range $[1, p-1]$.

Input

The first line of the input contains a single integer T ($1 \le T \le 100$), denoting the number of test cases.

Each of the next T lines contains a single integer p ($2 \le p \le 10^6$, p is a prime), denoting the parameter of the function described above.

Output

For each test case, output the answer in a single line.

Your answer is considered correct if its absolute or relative error does not exceed 10^{-6}.

Formally, let your answer be a, and the jury's answer be b. Your answer will be accepted if $\frac{|a-b|}{\max\{1,|b|\}} \le 10^{-6}$.

Sample Input 1	Sample Output 1
5 2 3 5 7 999983	1.0000000000 1.5000000000 2.0000000000 2.1666666667 15.9864347558

Note

For the 4-th test case in example, we have:

$$
\begin{aligned}
&\mathrm{F}(1,7)\\
&\mathrm{F}(2,7) \to \mathrm{F}(1,7)\\
&\mathrm{F}(3,7) \to \mathrm{F}(1,7)\\
&\mathrm{F}(4,7) \to \mathrm{F}(3,7) \to \mathrm{F}(1,7)\\
&\mathrm{F}(5,7) \to \mathrm{F}(2,7) \to \mathrm{F}(1,7)\\
&\mathrm{F}(6,7) \to \mathrm{F}(1,7)
\end{aligned}
$$

So the answer is $(1+2+2+3+3+2)/6 = 2.16666666\cdots$.

This page is intentionally left blank.

Problem K. Kanade Hates Recruitment

Time limit: 1 second

Because of the thriller adventure game *The 3rd Building*, more and more students get interested in workshops. Because of this, Kanade found that when the recruitment just started, many more students participated this year than before. But Kanade hates recruitment. She is annoyed by preparing recruitment questions. It's time for her to set a question, but she has no idea.

One day, she came up with an idea. Given n binary strings, i.e., strings only consist of `0` and `1`. Now she'd like to choose **one** binary string among them, then spilt it into two non-empty binary strings. Formally, for binary string s_i whose length is l ($l \geq 2$), she will choose an integer $k \in [1, l-1]$ at first. Then let the prefix of s_i of length k be a new binary string s_{n+1}. Finally, delete that prefix from string s_i. After this operation, the length of s_i becomes $l - k$, and there are $n + 1$ binary strings.

Kanade wants to know the number of different operations which can make the exclusive-or of the $n + 1$ binary strings **only contains** `0`. Two operations consider different if the binary strings which are chosen to split are different, or the values of k are different.

Let a be a binary string whose length is n, b be a binary string whose length is m. Let c be the exclusive-or of a and b, denoted $a \oplus b$, then c is a binary string of length $\max\{n, m\}$, which is defined as

$$c_i = (a \oplus b)_i = \begin{cases} \texttt{1}, & (1 \leq i \leq \min\{n, m\}, a_i \neq b_i) \\ \texttt{0}, & (1 \leq i \leq \min\{n, m\}, a_i = b_i) \\ a_i, & (i > \min\{n, m\}, n > m) \\ b_i, & (i > \min\{n, m\}, n < m) \end{cases}$$

The exclusive-or of $n + 1$ strings $s_1, s_2, \cdots, s_n, s_{n+1}$ is thus defined as $s_1 \oplus s_2 \oplus \cdots \oplus s_n \oplus s_{n+1}$.

Note that a binary string is a string where each character is indexed from left to right. This is different from the binary form of an integer.

She thought it easy to solve, but few students managed this question. She wants to know if her standard program was wrong. Please write a program solving the question above, so that she can check her code.

Input

The first line contains an integer n ($1 \leq n \leq 10^5$) — the number of binary strings.

In the next n lines, each line contains a binary string s_i ($1 \leq |s_i| \leq 10^6$), where $|s_i|$ denotes the length of s_i. The sum of all binary strings' lengths will not exceed 10^6.

It is guaranteed that there exists a binary string whose length is greater than 1.

Output

Output one integer in a line, representing the answer.

Sample Input 1	Sample Output 1
4 001 010 011 101	4

This page is intentionally left blank.

Problem L. Leave from CPC

Time limit: 1 second

Little Rabbit and Little Horse have been fighting for collegiate programming contest for almost four years. They plan to retire this year and so is it for many other members of the animal training team. Retirement means parting, and parting is inevitably full of sadness. If the retirement of one member will make the animal training team sad, then the retirement of multiple members will make everyone **cry** and fail to complete the final contest. Little Rabbit doesn't want this to happen, and he tells Little Horse to manage the number of retirements in each contest.

They get the list of the training teams and screened out the list of members who are expected to retire this year. Members in the list either participate in one contest or two contests — that's the rule of collegiate programming contest. Then each of the members who would like to retire will choose one contest he participates in to retire at that time. A contest can be described as a tuple $\langle x, y \rangle$, where x represents the time, and y represents the location. To simplify the problem, x and y are both positive integers and two contests A and B are defined to be **close** if $|A.x - B.x| \le d_x$ or $|A.y - B.y| \le d_y$. If any two members choose to retire in the same contest or two **close** contests, then everyone will be very sad and **cry**.

Little Rabbit and Little Horse want to know whether they can arrange everyone's retirement contests so that everyone will not **cry** and make it better to complete their final contests.

Input

The input contains several test cases.

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

For each test case: the first line contains three integers n $(1 \le n \le 2 \cdot 10^4)$ indicating the number of members who will retire this year, d_x and d_y $(1 \le d_x, d_y \le 10^9)$ described above. Following n lines, the i-th line starts with one integer k $(1 \le k \le 2)$ indicating the i-th member will take k contest(s). Then k tuple(s) $\langle x, y \rangle$ $(1 \le x, y \le 10^9)$ follow, each describing the information of a contest.

It is guaranteed that the sum of n will not exceed 10^5.

Output

For each test case, output the following text in a line: `Yes` if there is some arrangement so that everyone will not **cry**, or `No` if there does not exist such arrangement.

Sample Input 1	Sample Output 1
3 2 5 5 1 10 10 1 20 20 2 1 1 2 1 1 2 2 2 1 1 2 2 2 1 1 2 1 1 3 3 2 2 2 4 4	Yes No Yes

This page is intentionally left blank.