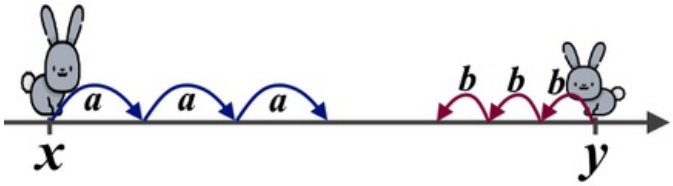
Codeforces Round #620 (Div. 2)

A. Two Rabbits

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

Being tired of participating in too many Codeforces rounds, Gildong decided to take some rest in a park. He sat down on a bench, and soon he found two rabbits hopping around. One of the rabbits was taller than the other.

He noticed that the two rabbits were hopping *towards each other*. The positions of the two rabbits can be represented as integer coordinates on a horizontal line. The taller rabbit is currently on position x, and the shorter rabbit is currently on position y ($x < y$). Every second, each rabbit hops to another position. The taller rabbit hops to the positive direction by a, and the shorter rabbit hops to the negative direction by b.



For example, let's say $x = 0$, $y = 10$, $a = 2$, and $b = 3$. At the 1-st second, each rabbit will be at position 2 and 7. At the 2-nd second, both rabbits will be at position 4.

Gildong is now wondering: *Will the two rabbits be at the same position **at the same moment**? If so, how long will it take?* Let's find a moment in time (in seconds) after which the rabbits will be at the same point.

Input

Each test contains one or more test cases. The first line contains the number of test cases t ($1 \le t \le 1000$).

Each test case contains exactly one line. The line consists of four integers x, y, a, b ($0 \le x < y \le 10^9$, $1 \le a, b \le 10^9$) — the current position of the taller rabbit, the current position of the shorter rabbit, the hopping distance of the taller rabbit, and the hopping distance of the shorter rabbit, respectively.

Output

For each test case, print the single integer: number of seconds the two rabbits will take to be at the same position.

If the two rabbits will never be at the same position simultaneously, print -1.

Example

input
5 0 10 2 3 0 10 3 3 900000000 1000000000 1 9999999 1 2 1 1 1 3 1 1
output
2 -1 10 -1 1

Note

The first case is explained in the description.

In the second case, each rabbit will be at position 3 and 7 respectively at the 1-st second. But in the 2-nd second they will be at 6 and 4 respectively, and we can see that they will never be at the same position since the distance between the two rabbits will only increase afterward.

B. Longest Palindrome

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

Returning back to problem solving, Gildong is now studying about palindromes. He learned that a *palindrome* is a string that is the same as its reverse. For example, strings "pop", "noon", "x", and "kkkkkk" are palindromes, while strings "moon", "tv", and "abab" are not. **An empty string is also a palindrome.**

Gildong loves this concept so much, so he wants to play with it. He has n *distinct* strings of equal length m. He wants to discard some of the strings (possibly none or all) and reorder the remaining strings so that the concatenation becomes a palindrome. He also wants the palindrome to be as long as possible. Please help him find one.

Input
The first line contains two integers n and m ($1 \le n \le 100$, $1 \le m \le 50$) — the number of strings and the length of each string.

Next n lines contain a string of length m each, consisting of lowercase Latin letters only. All strings are *distinct*.

Output
In the first line, print the length of the longest palindrome string you made.

In the second line, print that palindrome. If there are multiple answers, print any one of them. If the palindrome is empty, print an empty line or don't print this line at all.

Examples

input
3 3 tab one bat
output
6 tabbat

input
4 2 oo ox xo xx
output
6 oxxxxo

input
3 5 hello codef orces
output
0

input
9 4 abab baba abcd bcde cdef defg wxyz zyxw ijji
output
20 ababwxyzijjizyxwbaba

Note
In the first example, "battab" is also a valid answer.

In the second example, there can be 4 different valid answers including the sample output. We are not going to provide any hints for what the others are.

In the third example, the empty string is the only valid palindrome string.

C. Air Conditioner

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

input: standard input
output: standard output

Gildong owns a bulgogi restaurant. The restaurant has a lot of customers, so many of them like to make a reservation before visiting it.

Gildong tries so hard to satisfy the customers that he even memorized all customers' preferred temperature ranges! Looking through the reservation list, he wants to satisfy all customers by controlling the temperature of the restaurant.

The restaurant has an air conditioner that has 3 states: *off*, *heating*, and *cooling*. When it's *off*, the restaurant's temperature remains the same. When it's *heating*, the temperature increases by 1 in one minute. Lastly, when it's *cooling*, the temperature decreases by 1 in one minute. Gildong can change the state as many times as he wants, at any integer minutes. The air conditioner is *off* initially.

Each customer is characterized by three values: t_i — the time (in minutes) when the i-th customer visits the restaurant, l_i — the lower bound of their preferred temperature range, and h_i — the upper bound of their preferred temperature range.

A customer is satisfied if the temperature is within the preferred range at the instant they visit the restaurant. Formally, the i-th customer is satisfied if and only if the temperature is between l_i and h_i (inclusive) in the t_i-th minute.

Given the initial temperature, the list of reserved customers' visit times and their preferred temperature ranges, you're going to help him find if it's possible to satisfy all customers.

Input
Each test contains one or more test cases. The first line contains the number of test cases q ($1 \le q \le 500$). Description of the test cases follows.

The first line of each test case contains two integers n and m ($1 \le n \le 100$, $-10^9 \le m \le 10^9$), where n is the number of reserved customers and m is the initial temperature of the restaurant.

Next, n lines follow. The i-th line of them contains three integers t_i, l_i, and h_i ($1 \le t_i \le 10^9$, $-10^9 \le l_i \le h_i \le 10^9$), where t_i is the time when the i-th customer visits, l_i is the lower bound of their preferred temperature range, and h_i is the upper bound of their preferred temperature range. The preferred temperature ranges are **inclusive**.

The customers are given in non-decreasing order of their visit time, and the current time is 0.

Output
For each test case, print "YES" if it is possible to satisfy all customers. Otherwise, print "NO".

You can print each letter in any case (upper or lower).

Example

input
4 3 0 5 1 2 7 3 5 10 -1 0 2 12 5 7 10 10 16 20 3 -100 100 0 0 100 -50 50 200 100 100 1 100 99 -100 0
output
YES NO YES NO

Note
In the first case, Gildong can control the air conditioner to satisfy all customers in the following way:

- At 0-th minute, change the state to *heating* (the temperature is 0).
- At 2-nd minute, change the state to *off* (the temperature is 2).
- At 5-th minute, change the state to *heating* (the temperature is 2, the 1-st customer is satisfied).
- At 6-th minute, change the state to *off* (the temperature is 3).
- At 7-th minute, change the state to *cooling* (the temperature is 3, the 2-nd customer is satisfied).
- At 10-th minute, the temperature will be 0, which satisfies the last customer.

In the third case, Gildong can change the state to *heating* at 0-th minute and leave it be. Then all customers will be satisfied. Note that the 1-st customer's visit time equals the 2-nd customer's visit time.

In the second and the fourth case, Gildong has to make at least one customer unsatisfied.

D. Shortest and Longest LIS

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

Gildong recently learned how to find the longest increasing subsequence (LIS) in $O(n \log n)$ time for a sequence of length n. He wants to test himself if he can implement it correctly, but he couldn't find any online judges that would do it (even though there are actually many of them). So instead he's going to make a quiz for you about making permutations of n distinct integers between 1 and n, inclusive, to test his code with your output.

The quiz is as follows.

Gildong provides a string of length $n-1$, consisting of characters '<' and '>' only. The i-th (1-indexed) character is the comparison result between the i-th element and the $i+1$-st element of the sequence. If the i-th character of the string is '<', then the i-th element of the sequence is less than the $i+1$-st element. If the i-th character of the string is '>', then the i-th element of the sequence is greater than the $i+1$-st element.

He wants you to find two possible sequences (not necessarily distinct) consisting of n distinct integers between 1 and n, inclusive, each satisfying the comparison results, where the length of the LIS of the first sequence is minimum possible, and the length of the LIS of the second sequence is maximum possible.

Input

Each test contains one or more test cases. The first line contains the number of test cases t ($1 \le t \le 10^4$).

Each test case contains exactly one line, consisting of an integer and a string consisting of characters '<' and '>' only. The integer is n ($2 \le n \le 2 \cdot 10^5$), the length of the permutation you need to find. The string is the comparison results explained in the description. The length of the string is $n-1$.

It is guaranteed that the sum of all n in all test cases doesn't exceed $2 \cdot 10^5$.

Output

For each test case, print two lines with n integers each. The first line is the sequence with the minimum length of the LIS, and the second line is the sequence with the maximum length of the LIS. If there are multiple answers, print any one of them. Each sequence should contain all integers between 1 and n, inclusive, and should satisfy the comparison results.

It can be shown that at least one answer always exists.

Example

input
3 3 << 7 >><>>< 5 >>><
output
1 2 3 1 2 3 5 4 3 7 2 1 6 4 3 1 7 5 2 6 4 3 2 1 5 5 4 2 1 3

Note

In the first case, $1\ 2\ 3$ is the only possible answer.

In the second case, the shortest length of the LIS is 2, and the longest length of the LIS is 3. In the example of the maximum LIS sequence, 4 '3' $1\ 7$ '5' 2 '6' can be one of the possible LIS.

E. 1-Trees and Queries

time limit per test: 4 seconds
memory limit per test: 512 megabytes
input: standard input
output: standard output

Gildong was hiking a mountain, walking by millions of trees. Inspired by them, he suddenly came up with an interesting idea for trees in data structures: *What if we add another edge in a tree?*

Then he found that such tree-like graphs are called *1-trees*. Since Gildong was bored of solving too many tree problems, he wanted to see if similar techniques in trees can be used in 1-trees as well. Instead of solving it by himself, he's going to test you by providing queries on 1-trees.

First, he'll provide you a tree (not 1-tree) with n vertices, then he will ask you q queries. Each query contains 5 integers: x, y, a, b, and k. This means you're asked to determine if there exists a path from vertex a to b that contains exactly k edges after adding a bidirectional edge between vertices x and y. **A path can contain the same vertices and same edges multiple times**. All queries are independent of each other; i.e. the added edge in a query is removed in the next query.

Input

The first line contains an integer n ($3 \le n \le 10^5$), the number of vertices of the tree.

Next $n - 1$ lines contain two integers u and v ($1 \le u, v \le n$, $u \ne v$) each, which means there is an edge between vertex u and v. All edges are bidirectional and distinct.

Next line contains an integer q ($1 \le q \le 10^5$), the number of queries Gildong wants to ask.

Next q lines contain five integers x, y, a, b, and k each ($1 \le x, y, a, b \le n$, $x \ne y$, $1 \le k \le 10^9$) – the integers explained in the description. It is guaranteed that the edge between x and y does not exist in the original tree.

Output

For each query, print "YES" if there exists a path that contains exactly k edges from vertex a to b after adding an edge between vertices x and y. Otherwise, print "NO".

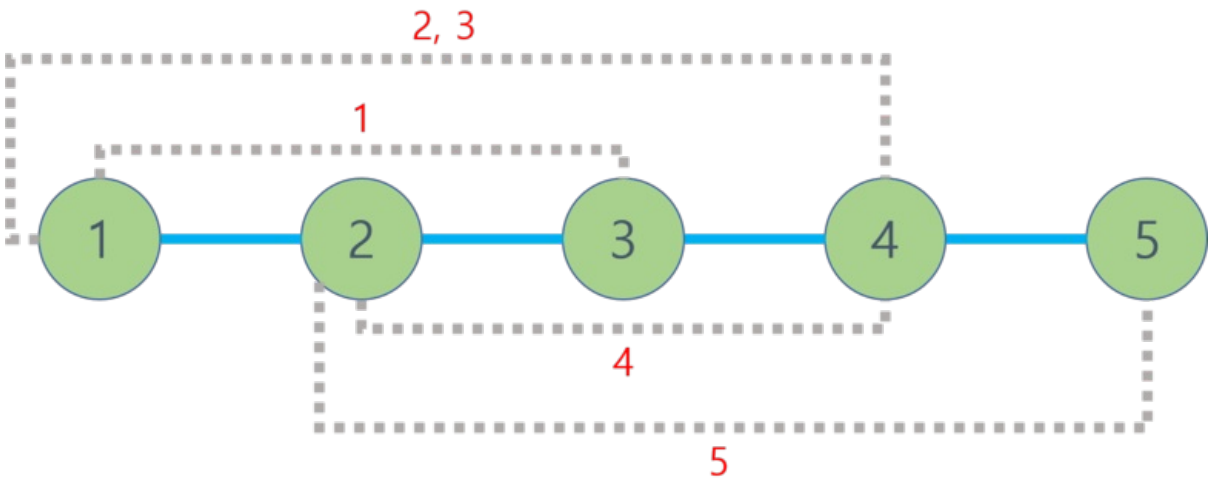
You can print each letter in any case (upper or lower).

Example

input
5 1 2 2 3 3 4 4 5 5 1 3 1 2 2 1 4 1 3 2 1 4 1 3 3 4 2 3 3 9 5 2 3 3 9
output
YES YES NO YES NO

Note

The image below describes the tree (circles and solid lines) and the added edges for each query (dotted lines).



Possible paths for the queries with "YES" answers are:

- 1-st query: $1 - 3 - 2$
- 2-nd query: $1 - 2 - 3$
- 4-th query: $3 - 4 - 2 - 3 - 4 - 2 - 3 - 4 - 2 - 3$

F1. Animal Observation (easy version)

time limit per test: 3 seconds
memory limit per test: 512 megabytes
input: standard input
output: standard output

The only difference between easy and hard versions is the constraint on k.

Gildong loves observing animals, so he bought two cameras to take videos of wild animals in a forest. The color of one camera is red, and the other one's color is blue.

Gildong is going to take videos for n days, starting from day 1 to day n. The forest can be divided into m areas, numbered from 1 to m. He'll use the cameras in the following way:

- On every odd day (1-st, 3-rd, 5-th, ...), bring the red camera to the forest and record a video for 2 days.
- On every even day (2-nd, 4-th, 6-th, ...), bring the blue camera to the forest and record a video for 2 days.
- If he starts recording on the n-th day with one of the cameras, the camera records for only one day.

Each camera can observe k consecutive areas of the forest. For example, if $m = 5$ and $k = 3$, he can put a camera to observe one of these three ranges of areas for two days: $[1, 3]$, $[2, 4]$, and $[3, 5]$.

Gildong got information about how many animals will be seen in each area each day. Since he would like to observe as many animals as possible, he wants you to find the best way to place the two cameras for n days. **Note that if the two cameras are observing the same area on the same day, the animals observed in that area are counted only once.**

Input
The first line contains three integers n, m, and k ($1 \le n \le 50$, $1 \le m \le 2 \cdot 10^4$, $1 \le k \le min(m, 20)$) – the number of days Gildong is going to record, the number of areas of the forest, and the range of the cameras, respectively.

Next n lines contain m integers each. The j-th integer in the $i + 1$-st line is the number of animals that can be seen on the i-th day in the j-th area. Each number of animals is between 0 and 1000, inclusive.

Output
Print one integer – the maximum number of animals that can be observed.

Examples

input
4 5 2 0 2 1 1 0 0 0 3 1 2 1 0 4 3 1 3 3 0 0 4
output
25

input
3 3 1 1 2 3 4 5 6 7 8 9
output
31

input
3 3 2 1 2 3 4 5 6 7 8 9
output
44

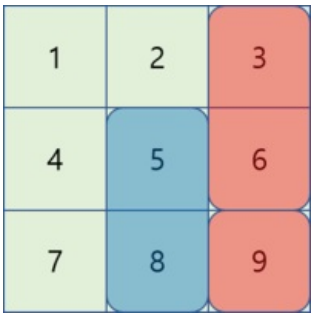
input
3 3 3 1 2 3 4 5 6 7 8 9
output
45

Note
The optimal way to observe animals in the four examples are as follows:

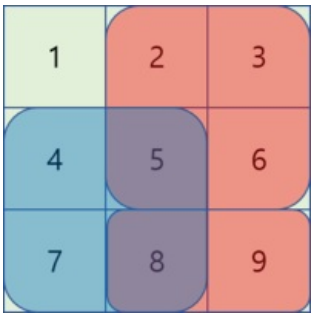
Example 1:

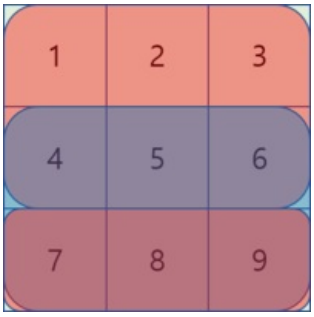
Example 2:

Example 3:



Example 4:





F2. Animal Observation (hard version)

time limit per test: 3 seconds
memory limit per test: 512 megabytes
input: standard input
output: standard output

The only difference between easy and hard versions is the constraint on k.

Gildong loves observing animals, so he bought two cameras to take videos of wild animals in a forest. The color of one camera is red, and the other one's color is blue.

Gildong is going to take videos for n days, starting from day 1 to day n. The forest can be divided into m areas, numbered from 1 to m. He'll use the cameras in the following way:

- On every odd day (1-st, 3-rd, 5-th, ...), bring the red camera to the forest and record a video for 2 days.
- On every even day (2-nd, 4-th, 6-th, ...), bring the blue camera to the forest and record a video for 2 days.
- If he starts recording on the n-th day with one of the cameras, the camera records for only one day.

Each camera can observe k consecutive areas of the forest. For example, if $m = 5$ and $k = 3$, he can put a camera to observe one of these three ranges of areas for two days: $[1, 3]$, $[2, 4]$, and $[3, 5]$.

Gildong got information about how many animals will be seen in each area on each day. Since he would like to observe as many

animals as possible, he wants you to find the best way to place the two cameras for n days. **Note that if the two cameras are observing the same area on the same day, the animals observed in that area are counted only once.**

Input

The first line contains three integers n, m, and k ($1 \le n \le 50$, $1 \le m \le 2 \cdot 10^4$, $1 \le k \le m$) – the number of days Gildong is going to record, the number of areas of the forest, and the range of the cameras, respectively.

Next n lines contain m integers each. The j-th integer in the $i+1$-st line is the number of animals that can be seen on the i-th day in the j-th area. Each number of animals is between 0 and 1000, inclusive.

Output

Print one integer – the maximum number of animals that can be observed.

Examples

input
4 5 2 0 2 1 1 0 0 0 3 1 2 1 0 4 3 1 3 3 0 0 4
output
25

input
3 3 1 1 2 3 4 5 6 7 8 9
output
31

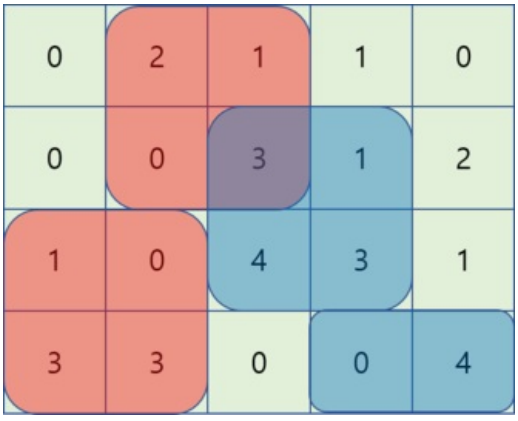
input
3 3 2 1 2 3 4 5 6 7 8 9
output
44

input
3 3 3 1 2 3 4 5 6 7 8 9
output
45

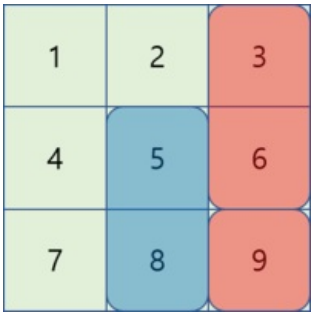
Note

The optimal way to observe animals in the four examples are as follows:

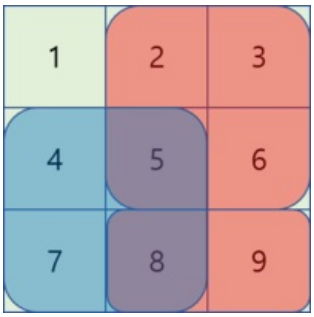
Example 1:



Example 2:



Example 3:



Example 4: