A. Three Strings

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

You are given three strings a, b and c of the same length n. The strings consist of lowercase English letters only. The i-th letter of a is a_i, the i-th letter of b is b_i, the i-th letter of c is c_i.

For every i ($1 \leq i \leq n$) you **must** swap (i.e. exchange) c_i with either a_i or b_i. So in total you'll perform exactly n swap operations, each of them either $c_i \leftrightarrow a_i$ or $c_i \leftrightarrow b_i$ (i iterates over all integers between 1 and n, inclusive).

For example, if a is "code", b is "true", and c is "help", you can make c equal to "crue" taking the 1-st and the 4-th letters from a and the others from b. In this way a becomes "hodp" and b becomes "tele".

Is it possible that after these swaps the string a becomes exactly the same as the string b?

Input

The input consists of multiple test cases. The first line contains a single integer t ($1 \leq t \leq 100$) — the number of test cases. The description of the test cases follows.

The first line of each test case contains a string of lowercase English letters a.

The second line of each test case contains a string of lowercase English letters b.

The third line of each test case contains a string of lowercase English letters c.

It is guaranteed that in each test case these three strings are non-empty and have the same length, which is not exceeding 100.

Output

Print t lines with answers for all test cases. For each test case:

If it is possible to make string a equal to string b print "YES" (without quotes), otherwise print "NO" (without quotes).

You can print either lowercase or uppercase letters in the answers.

Example

input
4 aaa bbb ccc abc bca bca aabb bbaa baba imi mii iim

output
NO YES YES NO

Note

In the first test case, it is impossible to do the swaps so that string a becomes exactly the same as string b.

In the second test case, you should swap c_i with a_i for all possible i. After the swaps a becomes "bca", b becomes "bca" and c becomes "abc". Here the strings a and b are equal.

In the third test case, you should swap c_1 with a_1, c_2 with b_2, c_3 with b_3 and c_4 with a_4. Then string a becomes "baba", string b becomes "baba" and string c becomes "abab". Here the strings a and b are equal.

In the fourth test case, it is impossible to do the swaps so that string a becomes exactly the same as string b.

B. Motarack's Birthday

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Dark is going to attend Motarack's birthday. Dark decided that the gift he is going to give to Motarack is an array a of n non-negative integers.

Dark created that array 1000 years ago, so some elements in that array disappeared. Dark knows that Motarack hates to see an array that has two adjacent elements with a high absolute difference between them. He doesn't have much time so he wants to choose an integer k ($0 \leq k \leq 10^9$) and replaces all missing elements in the array a with k.

Let m be the maximum absolute difference between all adjacent elements (i.e. the maximum value of $|a_i - a_{i+1}|$ for all $1 \leq i \leq n-1$) in the array a after Dark replaces all missing elements with k.

Dark should choose an integer k so that m is minimized. Can you help him?

Input

The input consists of multiple test cases. The first line contains a single integer t ($1 \leq t \leq 10^4$) — the number of test cases. The description of the test cases follows.

The first line of each test case contains one integer n ($2 \leq n \leq 10^5$) — the size of the array a.

The second line of each test case contains n integers $a_1, a_2, \ldots, a_n$ ($-1 \leq a_i \leq 10^9$). If $a_i = -1$, then the i-th integer is missing. It is guaranteed that at least one integer is missing in every test case.

It is guaranteed, that the sum of n for all test cases does not exceed $4 \cdot 10^5$.

Output
Print the answers for each test case in the following format:

You should print two integers, the minimum possible value of m and an integer k ($0 \le k \le 10^9$) that makes the maximum absolute difference between adjacent elements in the array a equal to m.

Make sure that after replacing all the missing elements with k, the maximum absolute difference between adjacent elements becomes m.

If there is more than one possible k, you can print any of them.

Example

input
7 5 -1 10 -1 12 -1 5 -1 40 35 -1 35 6 -1 -1 9 -1 3 -1 2 -1 -1 2 0 -1 4 1 -1 3 -1 7 1 -1 7 5 2 -1 5

output
1 11 5 35 3 6 0 42 0 0 1 2 3 4

Note
In the first test case after replacing all missing elements with 11 the array becomes $[11, 10, 11, 12, 11]$. The absolute difference between any adjacent elements is 1. It is impossible to choose a value of k, such that the absolute difference between any adjacent element will be ≤ 0. So, the answer is 1.

In the third test case after replacing all missing elements with 6 the array becomes $[6, 6, 9, 6, 3, 6]$.

- $|a_1 - a_2| = |6 - 6| = 0$;
- $|a_2 - a_3| = |6 - 9| = 3$;
- $|a_3 - a_4| = |9 - 6| = 3$;
- $|a_4 - a_5| = |6 - 3| = 3$;
- $|a_5 - a_6| = |3 - 6| = 3$.

So, the maximum difference between any adjacent elements is 3.

C. Ayoub's function

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

Ayoub thinks that he is a very smart person, so he created a function $f(s)$, where s is a binary string (a string which contains only symbols "0" and "1"). The function $f(s)$ is equal to the number of substrings in the string s that contains at least one symbol, that is equal to "1".

More formally, $f(s)$ is equal to the number of pairs of integers (l, r), such that $1 \le l \le r \le |s|$ (where $|s|$ is equal to the length of string s), such that at least one of the symbols $s_l, s_{l+1}, \ldots, s_r$ is equal to "1".

For example, if $s =$"01010" then $f(s) = 12$, because there are 12 such pairs (l, r):
$(1, 2), (1, 3), (1, 4), (1, 5), (2, 2), (2, 3), (2, 4), (2, 5), (3, 4), (3, 5), (4, 4), (4, 5)$.

Ayoub also thinks that he is smarter than Mahmoud so he gave him two integers n and m and asked him this problem. For all binary strings s of length n which contains exactly m symbols equal to "1", find the maximum value of $f(s)$.

Mahmoud couldn't solve the problem so he asked you for help. Can you help him?

Input
The input consists of multiple test cases. The first line contains a single integer t ($1 \le t \le 10^5$) — the number of test cases. The description of the test cases follows.

The only line for each test case contains two integers n, m ($1 \le n \le 10^9$, $0 \le m \le n$) — the length of the string and the number of symbols equal to "1" in it.

Output
For every test case print one integer number — the maximum value of $f(s)$ over all strings s of length n, which has exactly m symbols, equal to "1".

Example

input
5 3 1 3 2 3 3 4 0 5 2

output
4 5 6 0 12

Note

In the first test case, there exists only 3 strings of length 3, which has exactly 1 symbol, equal to "1". These strings are: s_1 ="100", s_2 ="010", s_3 ="001". The values of f for them are: $f(s_1) = 3, f(s_2) = 4, f(s_3) = 3$, so the maximum value is 4 and the answer is 4.

In the second test case, the string s with the maximum value is "101".

In the third test case, the string s with the maximum value is "111".

In the fourth test case, the only string s of length 4, which has exactly 0 symbols, equal to "1" is "0000" and the value of f for that string is 0, so the answer is 0.

In the fifth test case, the string s with the maximum value is "01010" and it is described as an example in the problem statement.

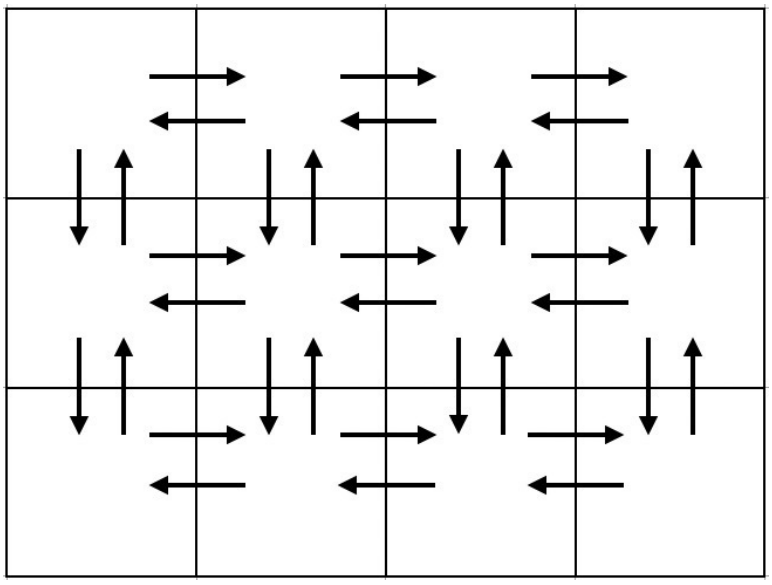
D. Time to Run

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

Bashar was practicing for the national programming contest. Because of sitting too much in front of the computer without doing physical movements and eating a lot Bashar became much fatter. Bashar is going to quit programming after the national contest and he is going to become an actor (just like his father), so he should lose weight.

In order to lose weight, Bashar is going to run for k kilometers. Bashar is going to run in a place that looks like a grid of n rows and m columns. In this grid there are two one-way roads of one-kilometer length between each pair of adjacent by side cells, one road is going from the first cell to the second one, and the other road is going from the second cell to the first one. So, there are exactly $(4nm - 2n - 2m)$ roads.

Let's take, for example, $n = 3$ and $m = 4$. In this case, there are 34 roads. It is the picture of this case (arrows describe roads):



Bashar wants to run by these rules:

- He starts at the top-left cell in the grid;
- In one move Bashar may go up (the symbol 'U'), down (the symbol 'D'), left (the symbol 'L') or right (the symbol 'R'). More formally, if he stands in the cell in the row i and in the column j, i.e. in the cell (i, j) he will move to:
  - in the case 'U' to the cell $(i - 1, j)$;
  - in the case 'D' to the cell $(i + 1, j)$;
  - in the case 'L' to the cell $(i, j - 1)$;
  - in the case 'R' to the cell $(i, j + 1)$;
- He wants to run exactly k kilometers, so he wants to make exactly k moves;
- Bashar can finish in any cell of the grid;
- He can't go out of the grid so at any moment of the time he should be on some cell;
- Bashar doesn't want to get bored while running so he must **not** visit the same road twice. **But he can visit the same cell any number of times**.

Bashar asks you if it is possible to run by such rules. If it is possible, you should tell him how should he run.

You should give him a steps to do and since Bashar can't remember too many steps, a should not exceed 3000. In every step, you should give him an integer f and a string of moves s of length at most 4 which means that he should repeat the moves in the string s for f times. He will perform the steps in the order you print them.

For example, if the steps are 2 RUD, 3 UUL then the moves he is going to move are RUD $+$ RUD $+$ UUL $+$ UUL $+$ UUL $=$ RUDRUDUULUULUUL.

Can you help him and give him a correct sequence of moves such that the total distance he will run is equal to k kilometers or say, that it is impossible?

Input

The only line contains three integers n, m and k ($1 \le n, m \le 500$, $1 \le k \le 10^9$), which are the number of rows and the number of columns in the grid and the total distance Bashar wants to run.

Output

If there is no possible way to run k kilometers, print "NO" (without quotes), otherwise print "YES" (without quotes) in the first line.

If the answer is "YES", on the second line print an integer a ($1 \le a \le 3000$) — the number of steps, then print a lines describing the steps.

To describe a step, print an integer f ($1 \le f \le 10^9$) and a string of moves s of length at most 4. Every character in s should be 'U', 'D', 'L' or 'R'.

Bashar will start from the top-left cell. Make sure to move exactly k moves without visiting the same road twice and without going

outside the grid. **He can finish at any cell**.

We can show that if it is possible to run exactly k kilometers, then it is possible to describe the path under such output constraints.

Examples

input
3 3 4
output
YES 2 2 R 2 L

input
3 3 1000000000
output
NO

input
3 3 8
output
YES 3 2 R 2 D 1 LLRR

input
4 4 9
output
YES 1 3 RLD

input
3 4 16
output
YES 8 3 R 3 L 1 D 3 R 1 D 1 U 3 L 1 D

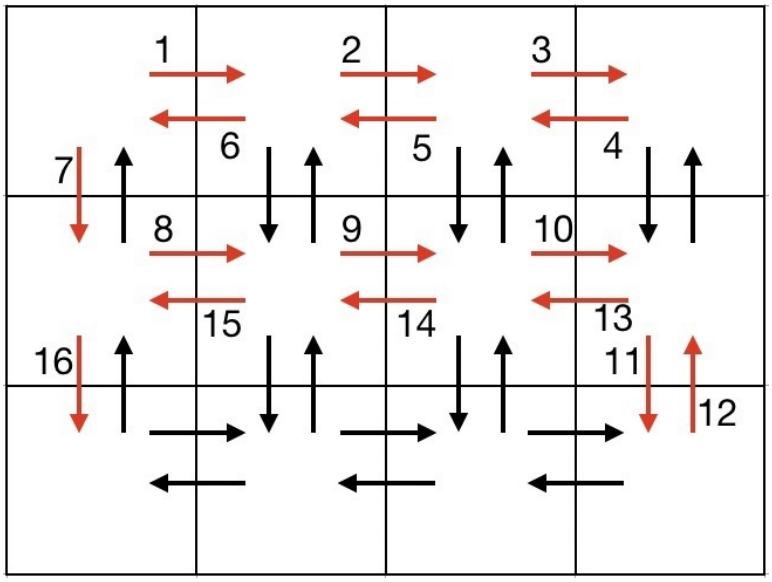
Note

The moves Bashar is going to move in the first example are: "RRLL".

It is not possible to run 1000000000 kilometers in the second example because the total length of the roads is smaller and Bashar can't run the same road twice.

The moves Bashar is going to move in the third example are: "RRDDLLRR".

The moves Bashar is going to move in the fifth example are: "RRRLLLDRRRDULLLD". It is the picture of his run (the roads on this way are marked with red and numbered in the order of his running):



E. Nanosoft

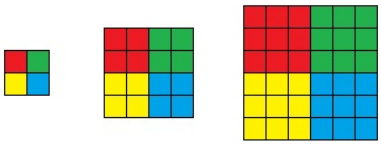
time limit per test: 2 seconds
memory limit per test: 512 megabytes
input: standard input
output: standard output

Warawreh created a great company called Nanosoft. The only thing that Warawreh still has to do is to place a large picture containing its logo on top of the company's building.

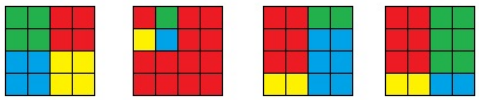
The logo of Nanosoft can be described as four squares of the same size merged together into one large square. The top left square

is colored with red, the top right square is colored with green, the bottom left square is colored with yellow and the bottom right square is colored with blue.

An Example of some correct logos:



An Example of some incorrect logos:



Warawreh went to Adhami's store in order to buy the needed picture. Although Adhami's store is very large he has only one picture that can be described as a grid of n rows and m columns. The color of every cell in the picture will be green (the symbol 'G'), red (the symbol 'R'), yellow (the symbol 'Y') or blue (the symbol 'B').

Adhami gave Warawreh q options, in every option he gave him a sub-rectangle from that picture and told him that he can cut that sub-rectangle for him. To choose the best option, Warawreh needs to know for every option the maximum area of sub-square inside the given sub-rectangle that can be a Nanosoft logo. If there are no such sub-squares, the answer is 0.

Warawreh couldn't find the best option himself so he asked you for help, can you help him?

Input

The first line of input contains three integers n, m and q $(1 \leq n, m \leq 500, 1 \leq q \leq 3 \cdot 10^5)$ — the number of row, the number columns and the number of options.

For the next n lines, every line will contain m characters. In the i-th line the j-th character will contain the color of the cell at the i-th row and j-th column of the Adhami's picture. The color of every cell will be one of these: {'G','Y','R','B'}.

For the next q lines, the input will contain four integers r_1, c_1, r_2 and c_2 $(1 \leq r_1 \leq r_2 \leq n, 1 \leq c_1 \leq c_2 \leq m)$. In that option, Adhami gave to Warawreh a sub-rectangle of the picture with the upper-left corner in the cell (r_1, c_1) and with the bottom-right corner in the cell (r_2, c_2).

Output

For every option print the maximum area of sub-square inside the given sub-rectangle, which can be a NanoSoft Logo. If there are no such sub-squares, print 0.

Examples

input
5 5 5 RRGGB RRGGY YYBBG YYBBR RBBRG 1 1 5 5 2 2 5 5 2 2 3 3 1 1 3 5 4 4 5 5

output
16 4 4 4 0

input
6 10 5 RRRGGGRRGG RRRGGGRRGG RRRGGGYYBB YYYBBBYYBB YYYBBBRGRG YYYBBBYBYB 1 1 6 10 1 3 3 10 2 2 6 6 1 7 6 10 2 1 5 10

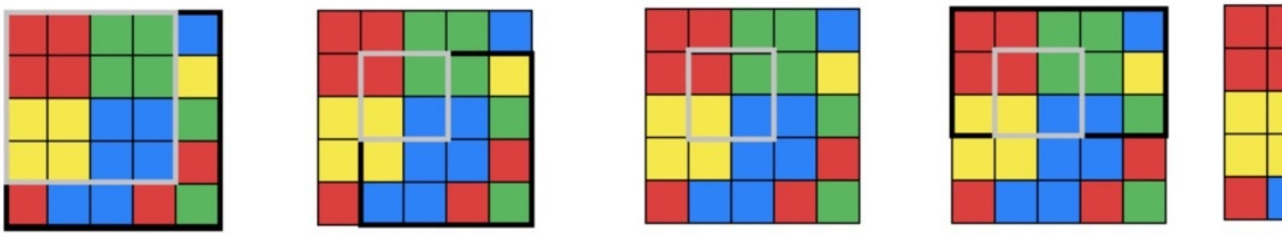
output
36 4 16 16 16

input
8 8 8 RRRRGGGG RRRRGGGG RRRRGGGG RRRRGGGG YYYYBBBB YYYYBBBB YYYYBBBB YYYYBBBB 1 1 8 8 5 2 5 7 3 1 8 6 2 3 5 8 1 2 6 8 2 1 5 5 2 1 7 7 6 5 7 5

output

```
64
0
16
4
16
4
36
0
```

Note
Picture for the first test:



The pictures from the left to the right corresponds to the options. The border of the sub-rectangle in the option is marked with black, the border of the sub-square with the maximal possible size, that can be cut is marked with gray.

F. Super Jaber

time limit per test: 5 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Jaber is a superhero in a large country that can be described as a grid with n rows and m columns, where every cell in that grid contains a different city.

Jaber gave every city in that country a specific color between 1 and k. In one second he can go from the current city to any of the cities adjacent by the side or to any city with the same color as the current city color.

Jaber has to do q missions. In every mission he will be in the city at row r_1 and column c_1, and he should help someone in the city at row r_2 and column c_2.

Jaber wants your help to tell him the minimum possible time to go from the starting city to the finishing city for every mission.

Input
The first line contains three integers n, m and k ($1 \le n, m \le 1000$, $1 \le k \le min(40, n \cdot m)$) — the number of rows, columns and colors.

Each of the next n lines contains m integers. In the i-th line, the j-th integer is a_{ij} ($1 \le a_{ij} \le k$), which is the color assigned to the city in the i-th row and j-th column.

The next line contains one integer q ($1 \le q \le 10^5$) — the number of missions.

For the next q lines, every line contains four integers r_1, c_1, r_2, c_2 ($1 \le r_1, r_2 \le n$, $1 \le c_1, c_2 \le m$) — the coordinates of the starting and the finishing cities of the corresponding mission.

It is guaranteed that for every color between 1 and k there is at least one city of that color.

Output
For every mission print the minimum possible time to reach city at the cell (r_2, c_2) starting from city at the cell (r_1, c_1).

Examples

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

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

Note
In the first example:

- mission 1: Jaber should go from the cell $(1, 1)$ to the cell $(3, 3)$ because they have the same colors, then from the cell $(3, 3)$ to the cell $(3, 4)$ because they are adjacent by side (two moves in total);
- mission 2: Jaber already starts in the finishing cell.

In the second example:

- mission 1: $(1, 1) \to (1, 2) \to (2, 2)$;
- mission 2: $(1, 1) \to (3, 2) \to (3, 3) \to (3, 4)$;

- mission 3: $(1,1) \to (3,2) \to (3,3) \to (2,4)$;
- mission 4: $(1,1) \to (1,2) \to (1,3) \to (1,4) \to (4,4)$.
