Problem A Phone Number

We know that if a phone number A is another phone number B's prefix, B is not able to be called. For an example, A is 123 while B is 12345, after pressing 123, we call A, and not able to call B.

Given N phone numbers, your task is to find whether there exits two numbers A and B that A is B's prefix.

Input

The input consists of several test cases.

The first line of input in each test case contains one integer N ($0<N<1001$), represent the number of phone numbers.

The next line contains N integers, describing the phone numbers.

The last case is followed by a line containing one zero.

Output

For each test case, if there exits a phone number that cannot be called, print "NO", otherwise print "YES" instead.

Sample Input	Output for the Sample Input
2 012 012345 2 12 012345 0	NO YES

Problem B Balloons

Both Saya and Kudo like balloons. One day, they heard that in the central park, there will be thousands of people fly balloons to pattern a big image.

They were very interested about this event, and also curious about the image.

Since there are too many balloons, it is very hard for them to compute anything they need. Can you help them?

You can assume that the image is an $N*N$ matrix, while each element can be either balloons or blank.

Suppose element A and element B are both balloons. They are connected if:

i) They are adjacent;

ii) There is a list of element C_1, C_2, … , C_n, while A and C_1 are connected, C_1 and C_2 are connected …C_n and B are connected.

And a connected block means that every pair of elements in the block is connected, while any element in the block is not connected with any element out of the block.

To Saya, element $A(xa, ya)$ and $B(xb, yb)$ is adjacent if $|xa - xb| + |ya - yb| \le 1$

But to Kudo, element $A(xa, ya)$ and element $B(xb, yb)$ is adjacent if $|xa - xb| \le 1$ and $|ya - yb| \le 1$

They want to know that there's how many connected blocks with there own definition of adjacent?

Input

The input consists of several test cases.

The first line of input in each test case contains one integer N ($0<N\leqslant 100$), which represents the size of the matrix.

Each of the next N lines contains a string whose length is N, represents the elements of the matrix. The string only consists of 0 and 1, while 0 represents a block and 1represents balloons.

The last case is followed by a line containing one zero.

Output

For each case, print the case number (1, 2 …) and the connected block's numbers with Saya and Kudo's definition. Your output format should imitate the sample output. Print a blank line after each test case.

Sample Input	Output for the Sample Input
5 11001 00100 11111 11010	Case 1: 3 2

10010 0	

Problem C Clockwise

Saya have a long necklace with N beads, and she signs the beads from 1 to N. Then she fixes them to the wall to show N-1 vectors – vector i starts from bead i and end up with bead i+1.

One day, Kudo comes to Saya's home, and she sees the beads on the wall. Kudo says it is not beautiful, and let Saya make it better.

She says: "I think it will be better if it is clockwise rotation. It means that to any vector i ($i<N-1$), it will have the same direction with vector i+1 after clockwise rotate T degrees, while $0\leqslant T<180$."

It is hard for Saya to reset the beads' places, so she can only remove some beads. To saving the beads, although she agrees with Kudo's suggestion, she thinks counterclockwise rotation is also acceptable. A counterclockwise rotation means to any vector i ($i<N-1$), it will have the same direction with vector i+1 after counterclockwise rotate T degrees, while $0<T\leqslant 180$."

Saya starts to compute at least how many beads she should remove to make a clockwise rotation or a counterclockwise rotation.

Since the necklace is very-very long, can you help her to solve this problem?

Input

The input consists of several test cases.

The first line of input in each test case contains one integer N ($2<N\leqslant 300$), which represents the number of beads.

Each of the next N lines contains two integer x and y, represents the coordinate of the beads. You can assume that $0<x, y<10000$.

The last case is followed by a line containing one zero.

Output

For each case, print your answer with the following format:

If it is clockwise rotation without removing any beads, please print "C; otherwise if it is counterclockwise rotation without removing any beads, print "CC" instead; otherwise, suppose remove at least x beads to make a clockwise rotation and remove at least y beads to make a counterclockwise rotation. If $x \leqslant y$, print "Remove x bead(s), C", otherwise print "Remove x bead(s), CC" instead.

Your output format should imitate the sample output. Print a blank line after each test case.

Sample Input	Output for the Sample Input
3 1 1 2 2 3 3 3 1 1 2 2	C CC Remove 1 bead(s), C

1 1 4 1 1 2 2 3 3 2 2 0	

Problem D Shopping

Saya and Kudo go shopping together.

You can assume the street as a straight line, while the shops are some points on the line.

They park their car at the leftmost shop, visit all the shops from left to right, and go back to their car.

Your task is to calculate the length of their route.

Input

The input consists of several test cases.

The first line of input in each test case contains one integer N ($0<N<100001$), represents the number of shops.

The next line contains N integers, describing the situation of the shops. You can assume that the situations of the shops are non-negative integer and smaller than 2^30.

The last case is followed by a line containing one zero.

Output

For each test case, print the length of their shopping route.

Sample Input	Output for the Sample Input
4 24 13 89 37 6 7 30 41 14 39 42 0	152 70

Explanation for the first sample: They park their car at shop 13; go to shop 24, 37 and 89 and finally return to shop 13. The total length is (24-13) + (37-24) + (89-37) + (89-13) = 152

Problem E Emergency

Kudo's real name is not Kudo. Her name is Kudryavka Anatolyevna Strugatskia, and Kudo is only her nickname.

Now, she is facing an emergency in her hometown:

Her mother is developing a new kind of spacecraft. This plan costs enormous energy but finally failed. What's more, because of the failed project, the government doesn't have enough resource take measure to the rising sea levels caused by global warming, lead to an island flooded by the sea.

Dissatisfied with her mother's spacecraft and the government, civil war has broken out. The foe wants to arrest the spacecraft project's participants and the "Chief criminal" – Kudo's mother – Doctor T's family.

At the beginning of the war, all the cities are occupied by the foe. But as time goes by, the cities recaptured one by one.

To prevent from the foe's arrest and boost morale, Kudo and some other people have to distract from a city to another. Although they can use some other means to transport, the most convenient way is using the inter-city roads. Assuming the city as a node and an inter-city road as an edge, you can treat the map as a weighted directed multigraph. An inter-city road is available if both its endpoint is recaptured.

Here comes the problem.

Given the traffic map, and the recaptured situation, can you tell Kudo what's the shortest path from one city to another only passing the recaptured cities?

Input

The input consists of several test cases.

The first line of input in each test case contains three integers N ($0<N\leqslant 300$), M ($0<M\leqslant 100000$) and Q ($0<Q\leqslant 100000$), which represents the number of cities, the numbers of inter-city roads and the number of operations.

Each of the next M lines contains three integer x, y and z, represents there is an inter-city road starts from x, end up with y and the length is z. You can assume that $0<z\leqslant 10000$.

Each of the next Q lines contains the operations with the following format:

a) 0 x – means city x has just been recaptured.

b) 1 x y – means asking the shortest path from x to y only passing the recaptured cities.

The last case is followed by a line containing three zeros.

Output

For each case, print the case number (1, 2 …) first.

For each operation 0, if city x is already recaptured (that is,the same 0 x operation appears again), print "City x is already recaptured."

For each operation 1, if city x or y is not recaptured yet, print "City x or y is not available." otherwise if Kudo can go from city x to city y only passing the recaptured cities, print the shortest path's length; otherwise print "No such path."

Your output format should imitate the sample output. Print a blank line after each test case.

Sample Input | **Output for the Sample Input**

Sample Input	Output for the Sample Input
3 3 6 0 1 1 1 2 1 0 2 3 1 0 2 0 0 0 2 1 0 2 1 2 0 0 2 0 0 0	Case 1: City 0 or 2 is not available. 3 No such path. City 2 is already recaptured.

Problem F Fairy tale

It is said that in a school's underground, there is a huge treasure which can meet any desire of the owner.

The Spy Union (SU) is very interest in this legend. After much investigation, SU finally get the answer and let the youngest spy sneak into the school. That's why Saya is now here.

Today, Saya found the entrance eventually.

She enters the entrance, and found her in a fairy-tale world.

"Welcome!" says a fairy, "I am Ivan. My responsibility is to protect the treasure, and give it to the one who have the ability to own it."

Then Ivan gives Saya three problems.

With your help, Saya finished the first and the second problem (Problem H and I). Here comes the third. If Saya can solve this problem, the treasure belongs to her.

There is a big maze protecting the treasure. You can assume the maze as an $N*N$ matrix while each element in the matrix might be N (North), S (South), W (West) or E (East). At first, Saya is at the element (1, 1) – the north-west corner, and the treasure is at (N, N) – the south-east corner.

The designer have enchant to this matrix, so that the treasure might move from time to time respecting the following rules:

Suppose the treasure is in an element which marked with E. The treasure might eastward move a cell after a unit time. It is similar to S, W and N.

After a unit time, all the mark will change: E to W, W to S, S to N, and N to E. In another word, suppose an element which marked with E at time 0. At time 1, it might change to W, change to S at time 2, change to N at time 3, change to E at time 4, and so on.

Saya doesn't know the initial status of the marks. She is affected by this rule, but she decides to do something more.

Ivan gave Saya a special prop which called Riki. With Riki's help, she can get the position of the treasure.

In each unit time, Saya will do all of the following three things:

Firstly, she will check the treasure's position with Riki.

Secondly, she will move follow the designer's magic the same with the treasure.(If no element exists in the direction of movement,the movement will be cancelled.)

Thirdly, Saya can either move to one direction (N, S, E, and W) a cell or stay there. Saya prefers to be closer with the treasure; if there are many ways with the same geometrical distance, Saya prefers to stay there than move, prefer E than W, W than N, and N than S. Here we should use the position checked at the first step.

You are given the size of the matrix and all the marks of the elements at time 0. Your task is to simulate Saya and the treasure's movement, and then tell Saya the result.

Input

The input consists of several test cases.

The first line of input in each test case contains one integer N ($0<N\leqslant 30$), which represents the size of the matrix.

Each of the next *N* lines contains a string whose length is *N*, represents the elements of the matrix. The string only consists of N, E, W and S.

The last case is followed by a line containing one zero.

Output

For each case, print the case number (1, 2 …) and the result of Saya's explore.

If Saya can get the treasure at step x (x≤100)(that means at the begainning of time x, Saya and the treasure stay in the same cell), print "Get the treasure! At step *x*.".

If after simulating *x* (*x* ≤ 100) steps, you found out that Saya can't get the treasure, print "Impossible. At step *x*."

If you have simulated 100 steps but don't know whether Saya can get the treasure, print "Not sure."

Your output format should imitate the sample output. Print a blank line after each test case.

Sample Input	Output for the Sample Input
5 EWSSE NNENN EENNE SWSEW NSNSW 0	Case 1: Get the treasure! At step 12.

Q&A

Q: How can I know it's impossible for Saya to get the treasure?

A: Suppose at time 5, Saya at (1, 1) while the treasure at (2, 2); at time 13, Saya at (1, 1) while the treasure at (2, 2) again. Since both Saya and the treasure go to the same place and have the same direction again, it is a loop, and they will just repeats the moves forever. So at time 13, we can judge it is impossible.

Problem G　Greatest Number

Saya likes math, because she think math can make her cleverer.

One day, Kudo invited a very simple game:

Given N integers, then the players choose no more than four integers from them (can be repeated) and add them together. Finally, the one whose sum is the largest wins the game. It seems very simple, but there is one more condition: the sum shouldn't larger than a number M.

Saya is very interest in this game. She says that since the number of integers is finite, we can enumerate all the selecting and find the largest sum. Saya calls the largest sum Greatest Number (GN). After reflecting for a while, Saya declares that she found the GN and shows her answer.

Kudo wants to know whether Saya's answer is the best, so she comes to you for help.

Can you help her to compute the GN?

Input

The input consists of several test cases.

The first line of input in each test case contains two integers N ($0<N\leqslant 1000$) and M($0<M\leqslant 1000000000$), which represent the number of integers and the upper bound.

Each of the next N lines contains the integers. (Not larger than 1000000000)

The last case is followed by a line containing two zeros.

Output

For each case, print the case number (1, 2 …) and the GN.

Your output format should imitate the sample output. Print a blank line after each test case.

Sample Input	Output for the Sample Input
2 10 100 2 0 0	Case 1: 8

Problem H Hello World!

We know that Ivan gives Saya three problems to solve (Problem F), and this is the first problem.

"We need a programmer to help us for some projects. If you show us that you or one of your friends is able to program, you can pass the first hurdle.

I will give you a problem to solve. Since this is the first hurdle, it is very simple."

We all know that the simplest program is the "Hello World!" program. This is a problem just as simple as the "Hello World!"

In a large matrix, there are some elements has been marked. For every marked element, return a marked element whose row and column are larger than the showed element's row and column respectively. If there are multiple solutions, return the element whose row is the smallest; and if there are still multiple solutions, return the element whose column is the smallest. If there is no solution, return -1 -1.

Saya is not a programmer, so she comes to you for help.

Can you solve this problem for her?

Input

The input consists of several test cases.

The first line of input in each test case contains one integer N ($0<N\leqslant 1000$), which represents the number of marked element.

Each of the next N lines containing two integers r and c, represent the element's row and column. You can assume that $0<r$, $c\leqslant 300$. A marked element can be repeatedly showed.

The last case is followed by a line containing one zero.

Output

For each case, print the case number (1, 2 …), and for each element's row and column, output the result. Your output format should imitate the sample output. Print a blank line after each test case.

Sample Input	Output for the Sample Input

<pre>3
1 2
2 3
2 3
 	
0</pre>
	<pre>Case 1:
	2 3
	-1 -1
	-1 -1</pre>

Problem I   Ivan comes again!

The Fairy Ivan gave Saya three problems to solve (Problem F). After Saya finished the first problem (Problem H), here comes the second.

This is the enhanced version of Problem H.

There is a large matrix whose row and column are less than or equal to 1000000000. And there are three operations for the matrix:

1) *add*: Mark an element in the matrix. The element wasn't marked before it is marked.
2) *remove*: Delete an element's mark. The element was marked before the element's mark is deleted.
3) *find*: Show an element's row and column, and return a marked element's row and column, where the marked element's row and column are larger than the showed element's row and column respectively. If there are multiple solutions, return the element whose row is the smallest; and if there are still multiple solutions, return the element whose column is the smallest. If there is no solution, return -1.

Of course, Saya comes to you for help again.

Input

The input consists of several test cases.

The first line of input in each test case contains one integer N ($0<N \leqslant 200000$), which represents the number of operations.

Each of the next N lines containing an operation, as described above.

The last case is followed by a line containing one zero.

Output

For each case, print the case number (1, 2 …) first. Then, for each "*find*" operation, output the result. Your output format should imitate the sample output. Print a blank line after each test case.

Sample Input	Output for the Sample Input
4 add 2 3 find 1 2 remove 2 3 find 1 2 0	Case 1: 2 3 -1

Problem J Jerry Mouse

Kudo and Saya are good friends, and they are always together.

But today, since Saya is not here (Where is Saya? You can get the answer at Problem F), Kudo fells very bored. So Kudo starts to watch TV for fun.

Now, she is watching the famous cartoon “Tom and Jerry”. Jerry’s home have many entrance, he always enter his home in an entrance and get out from another.

Kudo suddenly thought that what will happen if there are many Jerrys?

Kudo finds out a paper box and dig many holes in the side of the box. Then, she marks every hole with an integer, representing its owner. Finally, she signs the box “Kudo’s House” as shown in Figure 1.

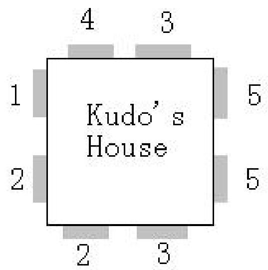

Figure 1 *The gray part represents a hole, and the numbers means its owner, i.e. the two gray parts in the lower left corner means the two holes belongs to mouse 2.*

Kudo think it is very interesting, so she makes a lot of boxes, and sign them “Saya’s House”, “Riki’s House”, “Lin’s House” and so on.

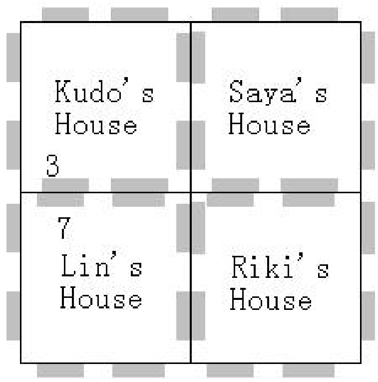

Figure 2 *The jointed gray parts represent the same hole.*

At last, she combines them together to make a large box as shown in Figure 2.

She defined mouse A and mouse B is the same mouse if and only if:

a) mouse A and mouse B are in a same house and A equals to B;
b) mouse A and mouse B are in different houses, but they have a hole that combined together, such as mouse 3 in Kudo’s House and mouse 7 in Lin’s House;
c) There exits a lists of mouse M1, M2 … Mp, while A and M1 are the same, M1 and M2 are the same … Mp and B are the same.

But there is a problem: Suppose mouse 3 in Kudo's House is the same with mouse 7 in Lin's House, while mouse 7 in Lin's House is the same with mouse 4 in Kudo's House. It means that mouse 3 and 4 in Kudo's House are the same!

Kudo is very confused, so she comes to you for help.

Given the initial *N* boxes, can you tell her finally each hole belongs to whom?

Here, *N* always equals to 1, 4, 9, 16, 25, 36, 49, 64, 81 or 100. Kudo always combines the boxes as shown in Figure 3, while $n = \sqrt{N}$.

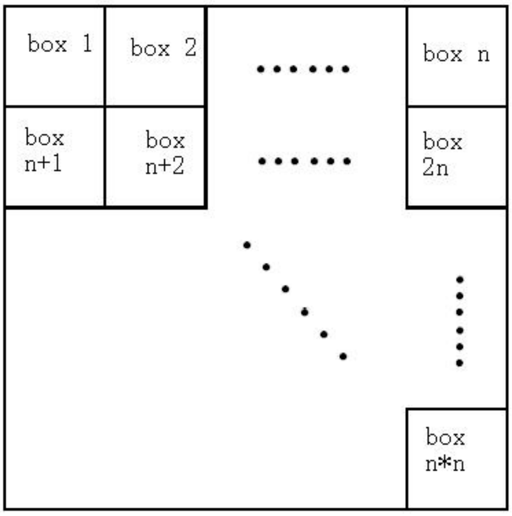

Figure 3 *How Kudo combines the boxes.*

Input

The input consists of several test cases.

The first line of input in each test case contains two integers *N* ($0<N\leqslant 100$) and *M* ($0<M\leqslant 1000$), which represent the number of boxes and the number of holes in each side of the box. In every side of the N boxes， You can assume that there are always M holes, and the M holes are arranged with equal spacing.

Each of the next *N* lines containing *4M* integers representing the holes on the boxes. The first *M* integers represent the holes on the upside, from left to right; the second *M* integers represent the holes on the downside, from left to right; the third *M* integers represent the holes on the leftward, from up to down; the forth *M* integers represent the holes on the rightward, from up to down. You can assume that the hole number is not greater than *M**2.

The last case is followed by a line containing two zeros.

Output

For each case, print the case number (1, 2 …) and *4*n*M* integers ($n = \sqrt{N}$) represents the holes on the upside, downside, leftward and rightward side of the large box, using the same format as the input file.

In your answer, the holes signed by the same numbers should belong to the same mouse, while the holes signed by different numbers should belong to different mouse. The number should be an integer, starting from 1. Since there are multiply solutions, please print the one whose first number is the least. If there are still multiply solutions, print the one whose second number is the least, and so on.

Your output format should imitate the sample output. Print a blank line after each test case.

Sample Input	**Output for the Sample Input**
4 1 2 2 1 1 2 2 1 1 1 1 2 2 1 1 2 2 0 0	`Case 1:` `1 2 1 2 3 4 3 4`