PROBLEM-1: The Water-Pipe

You have to lay a water-pipe connection through the great city, Chennai. The city is represented by a grid of N x N cells. Columns are numbered 1 to N from left to right. Rows are numbered 1 to N from top to bottom. In order to lay the water-pipe, you have to dig in the cells through which the water-pipe goes through. There is a cost involved with digging up each cell. Your task is to minimize the total cost for digging up cells.

Furthermore, the water-pipe should be laid from left column (column 1) to right column (column N) of the city satisfying the following conditions:

1. In the city, it doesn't matter from which row the water-pipe starts and ends; i.e., the water-pipe should start from any cell of column 1 and should end at any cell of column N.
2. When the water-pipe enters into the column i (1 <= i < N), the water-pipe can be laid to column (i+1) using one of following ways: (lets say water-pipe enters the column i at row j)
    - Lay the water-pipe upward any number of cells (say, d) within the city boundary and move to column (i+1) at row (j-d); i.e., (d+1) cells in column i will be digged up
    - Lay the water-pipe downward any number of cells (say d) within the city boundary and move to column (i+1) at row (j+d); i.e., (d+1) cells in column 1 will be digged up
    - Lay water-pipe to column (i+1) through current cell; i.e., only 1 cell will be digged up in column i

Task

Given digging up costs for each cell in the city, find the minimum cost for laying the water-pipe from column 1 to column N. Let, c(i,j) = cost of digging up the cell at column i and row j

1 <= N <= 1000 and 0 <= c(i,j) <= 1,000,000

Input Format

First line of the input will contain an integer N, the number of rows and columns in a city.

Next N lines will represent row 1 to row N, respectively.

Each row will have N integers separated by spaces representing costs of column 1 to column N respectively of the corresponding rows.

Output Format

Output should contain only one integer, minimum cost for digging up the cells that water-pipe goes through. The output (integer) will be followed by a NEWLINE (“\n”).

Example

Sample Input:

```
5
1 1 9 1 1
3 1 9 7 1
4 1 9 1 1
5 1 1 1 5
6 1 9 3 1
```

Sample Output:

```
9
```

Explanation:

The water-pipe path can be laid through the following cells (column,row):

(1,1) -> (2,1) -> (2,2) -> (2,3) -> (2,4) -> (3,4) -> (4,4) -> (4,3) -> (5,3)

The water-pipe starts from (1,1). Moves to column right at (2,1). Then, the water-pipe moves three cells downward to (2,4) and move to two columns right till (4,4). Then, the water-pipe moves one cell upward to (4,3) and move to the column right at (5,3).

Total cost of the path = 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 = 9

Note: Other paths would produce larger answers. For example, if you consider taking the first row across, then the answer would be 1+1+9+1+1=13, which is larger than the shortest path identified.

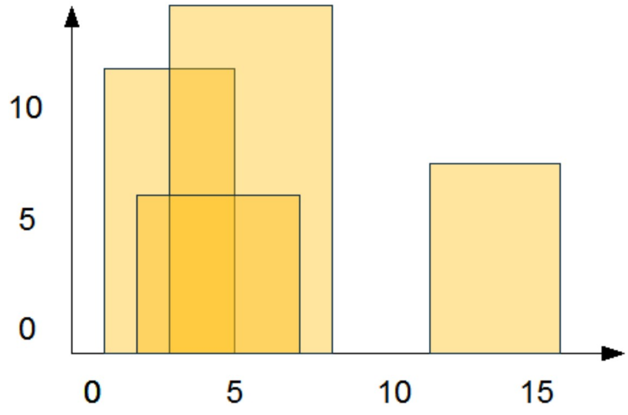
PROBLEM-2: The Machine Load

You have to write an output generator for a machine load display program. The idea is that we have a series of machines which return their load over time as rectangle coordinates, indicating the load (in the X axis) during the period of time (in the Y axis).

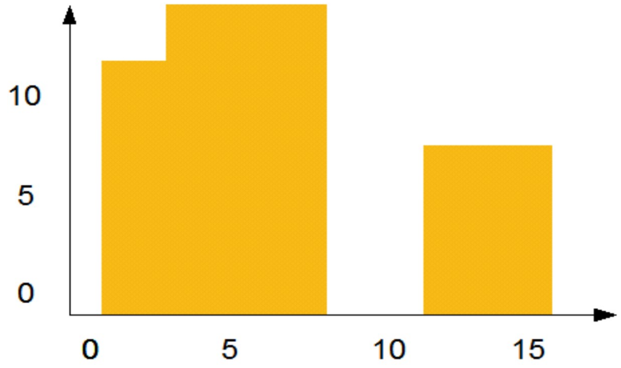
Input for each machine are three numbers indicating the time when we start measuring, the load level during that time and the time when we finish measuring. For example, an input for four machines like:

```
1  11  5
2  6  7
3  13  9
12  7  16
```

Should represent something like this (see right):



However, we are only interested in getting the maximum level at each time, so would prefer to see something like this (see right):



We have a drawing program that takes a list of number pairs, each one representing an X coordinate and a value, and draws a surface with height equal to the given value. Basically, for drawing the graphic shown right, it requires the following data:

```
1  11
3  13
9  0
12  7
16  0
```

Which should read something like: "from 1 on, value is 11, until we reach 3, and then value is 13, until we reach 9, when value goes 0, but then on 12 value goes to 7 and then on 16 it gets back to zero" (more or less).

Task

Write a program that receives the inputs for each of the machines (there can be up to 1,00,000 machines) as a list of number triplets in parantheses (as shown in the examples), as a parameter, and prints a string of comma-separated values between parentheses representing the appropiately formatted entry to our drawing system (also as shown) or the word **ERROR** if the input is not valid. Please note the times and load values will always be integers.

Input Format

First line of the input will contain an integer N, the number of machines.
Next N lines will represent the inputs for each machines where each line consists of three integers indicating the time when we start measuring, the load level during that time and the time when we finish measuring. Each of this three numbers are integers and separated by spaces.

Output Format

Output lines should contain a list of integer pairs (separated by space and ends with a NEWLINE "\n"), where each line contain an integer pair, first one representing an X-coordinate (time) and second representing the value (machine load). The output will end by a NEWLINE ("\n").

Example

Sample Input:

```
4
1 11 5
2 6 7
3 13 9
12 7 16
```

Sample Output:

```
1 11
3 13
9 0
12 7
16 0
```

Sample Input:

```
2
1 5 3
3 5
```

Sample Output:

```
ERROR
```

Sample Input:

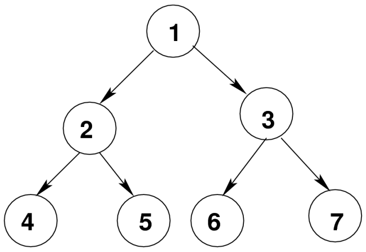
```
5
12 7 16
14 3 25
19 18 22
23 13 29
24 4 28
```

Sample Output:

```
12 7
16 3
19 18
22 3
23 13
29 0
```

PROBLEM-3: Thief and Police Game

A group of Police are chasing a Thief in a planned city where the streets follow a binary tree structure. In every turn, the Thief moves and then all the Polices are moving. Each move is along an edge of the tree. If any of the Police is on the same node as the Thief, before or after Thief's move, then the Polices are able to catch the thief.



The streets in the city are planned so well that these always form a Full Binary Tree structure. The tree's height is H and nodes are numbered from 1 to 2^{H-1}. For example, see the tree of height 3, given in the right side.

Now, any Police as well as the Thief can move one step in the following way (each move is characterized by one letter):

- **U** – moving up the tree (for example, from node 7 to node 3 in the example above)
- **L** – moving to the left child (for example, from node 1 to node 2 in the example above)
- **R** – moving to the right child (for example, from node 2 to node 5 in the example above)
- **S** – stay in the same place (for example, from node 4 to node 4 in the example above)

The Poilice station is always situated at the root node of the tree. So, all the Polices start moving from the root, while the Thief can start from any node. Note that, any Police catches the Thief if he/she is on the same spot before or after the move; or if they cross each other on their way.

Task

Please write a C-program that gets the moves of the Thief and the moves of all the Polices and decides when the Thief is caught or whether the Thief has escaped.

Input Format

The first line contains the number of problems (an Integer), P.

Each of next P lines contains three numbers (space separated) and M strings of letters (each string is space separated): `N M L STR_1 STR_2 ... STR_M`, where

`N` = number of Polices; `M` = total number of moves/steps that are allowed to make; and `L` = Bugs's initial location. These numbers are followed by `M` groups/strings of `N+1` letters each which define the moves of Thief (the first letter) and each one of the `N` Polices (the next `N` letters).

Output Format

The move number after which the Thief is caught, and the index of the Police (Police indexes start from 1) who caught the Thief (if several has, give the minimal one), or 0 0 if the Thief survives the search (not get caught).

Note: There is a new line character at the end of the result.

Example

Sample Input

```
2
4 2 3 LLLRR SLRLR
2 6 8 LLR ULS SRS USR USU SUU
```

Sample Output

```
2 3
0 0
```