(100 Marks)

You and your friend go to a game arcade where you choose to play the game **Lucky Pick**. In the game, there is a square grid and on each block, some money is placed on it. When a player chooses a block, the machine randomly chooses a block from the neighboring ones and the chosen block (consider 8 neighborhood). The player is awarded the money that is placed on the block that the machine selects. Your friend needs help choosing the block.

Your job is to return the block position(s) that will maximize the minimum amount your friend will win for sure. If there are more than one such block positions then the output must return for all these positions.

You will be given the Grid Description as -
The first line consists of the size of the square grid (N)
The next N lines each containing N numbers separated by '#', each number representing the amount of money put on that block
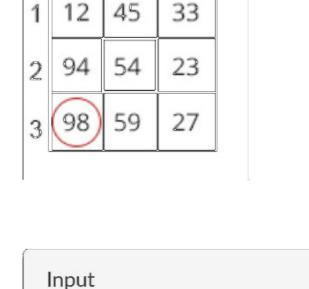
$1 < N < 500$

You need to print the array of string containing the position(s) of a block choosing which will give the maximum amount of money which your friend will definitely win.
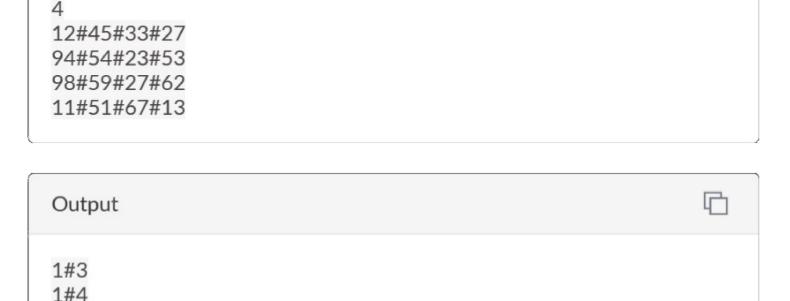
Input	
3 12#45#33 94#54#23 98#59#27	

Output	
3#1	

In the above example, if he selects the block (3,1), then under the best case, he could win is 98 and under the worst case the maximum he could win is 54. In such scenario, the worst case of block (3,1) gives your friend more money than the worst case of other blocks.

	1	2	3
1	12	45	33
2	94	54	23
3	(98)	59	27

Input	
4 12#45#33#27 94#54#23#53 98#59#27#62 11#51#67#13	

Output	
1#3 1#4 2#3 2#4	

	1	2	3	4
1	12	45	(33)	(27)
2	94	54	(23)	(53)
3	98	59	27	62
4	11	51	63	13

Note: If the output array contains multiple strings(block's positions), all the positions must be in the row-wise traversal order. In **Example 2**, the output is {1#3,1#4,2#3,2#4}. If your function is returning an array that has same elements (block's position) but in the different order, then the output array will be incorrect.

Time Limit(X):	1.00 sec(s) for each input.
Memory Limit:	512 MB
Source Limit:	100 KB
Allowed Languages:	C, C++, C++11, C++14, C#, Java, Java 8, PHP, PHP 7, Python, Python 3, Perl, Ruby, Node Js, Scala, Clojure, Haskell, Lua, Erlang, Swift, VBnet, Js, Objc, Pascal, Go, F#, D, Groovy, Tcl, Ocaml, Smalltalk, Cobol, Racket, Bash, GNU Octave, Rust, Common LISP, R, Julia, Fortran, Ada, Prolog, Icon, Elixir, CoffeeScript, Brainfuck, Pypy, Lolcode, Nim, Picolisp, Pike

Judge Environment