

Round B China New Grad Test 2014

A. Sudoku Checker[B. Meet and party](#)[C. Hex](#)[D. Dragon Maze](#)[E. Ignore all my comments](#)[Questions asked](#)

Submissions

Sudoku Checker

5pt	Not attempted 1471/2010 users correct (73%)
9pt	Not attempted 1146/1443 users correct (79%)

Meet and party

9pt	Not attempted 496/823 users correct (60%)
15pt	Not attempted 47/409 users correct (11%)

Hex

12pt	Not attempted 19/260 users correct (7%)
13pt	Not attempted 14/18 users correct (78%)

Dragon Maze

8pt	Not attempted 336/594 users correct (57%)
12pt	Not attempted 229/330 users correct (69%)

Ignore all my comments

17pt	Not attempted 216/468 users correct (46%)
0pt	Not attempted

Top Scores

TankEngineer	100
Nekosyndrome	100
I521530	100
W.Junqiao	100
LTzycLT	100
iloahz	100
drazil	87
navi	85
wishstudio	85
redsniper	76

Problem A. Sudoku Checker

This contest is open for practice. You can try every problem as many times as you like, though we won't keep track of which problems you solve. Read the [Quick-Start Guide](#) to get started.

Small input
5 points

Solve A-small

Large input
9 points

Solve A-large

Problem

Sudoku is a popular single player game. The objective is to fill a 9x9 matrix with digits so that each column, each row, and all 9 non-overlapping 3x3 sub-matrices contain all of the digits from 1 through 9. Each 9x9 matrix is partially completed at the start of game play and typically has a unique solution.

5	3			7				
6			1	9	5			
	9	8					6	
8				6				3
4			8		3			1
7				2				6
	6					2	8	
			4	1	9			5
				8			7	9

5	3	4	6	7	8	9	1	2
6	7	2	1	9	5	3	4	8
1	9	8	3	4	2	5	6	7
8	5	9	7	6	1	4	2	3
4	2	6	8	5	3	7	9	1
7	1	3	9	2	4	8	5	6
9	6	1	5	3	7	2	8	4
2	8	7	4	1	9	6	3	5
3	4	5	2	8	6	1	7	9

Given a completed $N^2 \times N^2$ Sudoku matrix, your task is to determine whether it is a *valid* solution. A *valid* solution must satisfy the following criteria:

- Each row contains each number from **1** to N^2 , once each.
- Each column contains each number from **1** to N^2 , once each.
- Divide the $N^2 \times N^2$ matrix into N^2 non-overlapping $N \times N$ sub-matrices. Each sub-matrix contains each number from **1** to N^2 , once each.

You don't need to worry about the uniqueness of the problem. Just check if the given matrix is a valid solution.

Input

The first line of the input gives the number of test cases, **T**. **T** test cases follow. Each test case starts with an integer **N**. The next N^2 lines describe a completed Sudoku solution, with each line contains exactly N^2 integers. All input integers are positive and less than 1000.

Output

For each test case, output one line containing "Case #x: y", where x is the case number (starting from 1) and y is "Yes" (quotes for clarity only) if it is a valid solution, or "No" (quotes for clarity only) if it is invalid. Note that the judge is case-sensitive, so answers of "yes" and "no" will not be accepted.

Limits

$1 \leq T \leq 100$.

Small dataset

N = 3.

Large dataset

$3 \leq \mathbf{N} \leq 6$.

Sample

Input	Output
3	Case #1: Yes
3	Case #2: No
5 3 4 6 7 8 9 1 2	Case #3: No
6 7 2 1 9 5 3 4 8	
1 9 8 3 4 2 5 6 7	
8 5 9 7 6 1 4 2 3	
4 2 6 8 5 3 7 9 1	
7 1 3 9 2 4 8 5 6	
9 6 1 5 3 7 2 8 4	
2 8 7 4 1 9 6 3 5	
3 4 5 2 8 6 1 7 9	
3	
1 2 3 4 5 6 7 8 9	
1 2 3 4 5 6 7 8 9	
1 2 3 4 5 6 7 8 9	
1 2 3 4 5 6 7 8 9	
1 2 3 4 5 6 7 8 9	
1 2 3 4 5 6 7 8 9	
1 2 3 4 5 6 7 8 9	
1 2 3 4 5 6 7 8 9	
1 2 3 4 5 6 7 8 9	
3	
5 3 4 6 7 8 9 1 2	
6 7 2 1 9 5 3 4 8	
1 9 8 3 4 2 5 6 7	
8 5 9 7 6 1 4 2 3	
4 2 6 8 999 3 7 9 1	
7 1 3 9 2 4 8 5 6	
9 6 1 5 3 7 2 8 4	
2 8 7 4 1 9 6 3 5	
3 4 5 2 8 6 1 7 9	

All problem statements, input data and contest analyses are licensed under the [Creative Commons Attribution License](#).

© 2008-2017 Google [Google Home](#) - [Terms and Conditions](#) - [Privacy Policies and Principles](#)

Powered by



Google Cloud Platform

Round B China New Grad Test 2014

[A. Sudoku Checker](#)

B. Meet and party

[C. Hex](#)

[D. Dragon Maze](#)

[E. Ignore all my comments](#)

[Questions asked](#)

Submissions

Sudoku Checker

5pt Not attempted
1471/2010 users
correct (73%)

9pt Not attempted
1146/1443 users
correct (79%)

Meet and party

9pt Not attempted
496/823 users
correct (60%)

15pt Not attempted
47/409 users
correct (11%)

Hex

12pt Not attempted
19/260 users
correct (7%)

13pt Not attempted
14/18 users correct
(78%)

Dragon Maze

8pt Not attempted
336/594 users
correct (57%)

12pt Not attempted
229/330 users
correct (69%)

Ignore all my comments

17pt Not attempted
216/468 users
correct (46%)

0pt Not attempted

Top Scores

TankEngineer	100
Nekosyndrome	100
I521530	100
W.Junqiao	100
LTzycLT	100
iloahz	100
drazil	87
navi	85
wishstudio	85
redsniper	76

Problem B. Meet and party

This contest is open for practice. You can try every problem as many times as you like, though we won't keep track of which problems you solve. Read the [Quick-Start Guide](#) to get started.

Small input
9 points

Solve B-small

Large input
15 points

Solve B-large

Problem

Little Sin lives in a Manhattan-grid city, a 2D plane where people can only go north, west, south or east along the grid. The distance from (x_1, y_1) to (x_2, y_2) is $|x_1 - x_2| + |y_1 - y_2|$.

Little Sin really likes to party and is hoping to host a house party in Manhattan this Sunday. Little Sin has collected a list of people who will attend, and now needs to decide at whose home she will host the party.

Little Sin invited all of the people in several rectangular areas, and all of those people have said yes. A rectangular area is denoted as (x_1, y_1, x_2, y_2) , where $x_1 \leq x_2, y_1 \leq y_2$. People who live in a rectangular area fill all integral points inside it. So there are a total of $(x_2 - x_1 + 1) * (y_2 - y_1 + 1)$ people in the rectangular area (x_1, y_1, x_2, y_2) .

Little Sin knows the coordinates of those rectangular areas. She wants the party to be hosted at the home of one of the people who is attending, but she also doesn't want everyone else to have to travel very far: she wants to minimize the sum of all distances from all attendees' houses to the party. Can you help her?

Input

The first line of the input gives the number of test cases, **T**. **T** test cases follow. Each test case starts with a line containing a single integer: the number of rectangular areas, **B**. **B** lines follow. Each line contains 4 integers: x_1, y_1, x_2, y_2 , denoting the coordinates of a rectangular area of people Little Sin has invited to her party.

Output

For each test case, output one line containing "Case #t: x y d", where t is the case number (starting from 1) and (x, y) is the coordinates of the person whose home the party should be hosted. If there are multiple positions with the same minimum total distance, choose the one with the smallest x. If there are still multiple positions, choose the one with the smallest y. The value d is the sum of the distances from all attendees' houses to the point (x, y) .

Limits

$1 \leq T \leq 10$.

$|x_1|, |y_1|, |x_2|, |y_2| \leq 10^9$.

$x_1 \leq x_2, y_1 \leq y_2$.

The rectangular areas within a test case don't intersect.

Small dataset

$1 \leq B \leq 100$.

$1 \leq \text{Total number of people in each test case} \leq 1000$.

Large dataset

$1 \leq B \leq 1000$.

$1 \leq \text{Total number of people in each test case} \leq 1000000$.

Sample

Input	Output
2	Case #1: 1 1 12
1	Case #2: -1 2 6
0 0 2 2	
3	
-1 2 -1 2	
0 0 0 0	
1 3 1 3	

All problem statements, input data and contest analyses are licensed under the [Creative Commons Attribution License](#).

© 2008-2017 Google [Google Home](#) - [Terms and Conditions](#) - [Privacy Policies and Principles](#)

Powered by



Google Cloud Platform

Round B China New Grad Test 2014

[A. Sudoku Checker](#)[B. Meet and party](#)**C. Hex**[D. Dragon Maze](#)[E. Ignore all my comments](#)[Questions asked](#)

Submissions

Sudoku Checker

5pt Not attempted
1471/2010 users
correct (73%)9pt Not attempted
1146/1443 users
correct (79%)

Meet and party

9pt Not attempted
496/823 users
correct (60%)15pt Not attempted
47/409 users
correct (11%)

Hex

12pt Not attempted
19/260 users
correct (7%)13pt Not attempted
14/18 users correct
(78%)

Dragon Maze

8pt Not attempted
336/594 users
correct (57%)12pt Not attempted
229/330 users
correct (69%)

Ignore all my comments

17pt Not attempted
216/468 users
correct (46%)

0pt Not attempted

Top Scores

TankEngineer	100
Nekosyndrome	100
I521530	100
W.Junqiao	100
LTzycLT	100
iloahz	100
drazil	87
navi	85
wishstudio	85
redsniper	76

Problem C. Hex

This contest is open for practice. You can try every problem as many times as you like, though we won't keep track of which problems you solve. Read the [Quick-Start Guide](#) to get started.

Small input
12 points

Solve C-small

Large input
13 points

Solve C-large

Problem

This problem was inspired by a board game called Hex, designed independently by Piet Hein and John Nash. It has a similar idea, but does not assume you have played Hex.

This game is played on an $N \times N$ board, where each cell is a hexagon. There are two players: Red side (using red stones) and Blue side (using blue stones). The board starts empty, and the two players take turns placing a stone of their color on a single cell within the overall playing board. Each player can place their stone on any cell not occupied by another stone of any color. There is no requirement that a stone must be placed beside another stone of the same color. The player to start first is determined randomly (with equal probability among the two players).

The upper side and lower sides of the board are marked as red, and the other two sides are marked as blue. The goal of the game is to form a connected path of one player's stones connecting the two sides of the board that have that player's color. The first player to achieve this wins. Note that the four corners are considered connected to both colors.

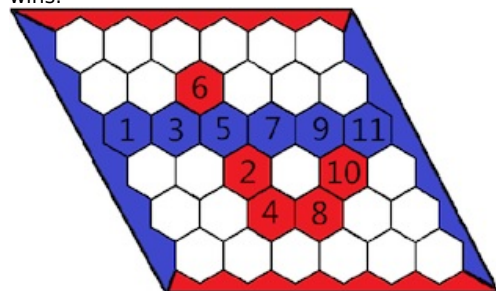
The game ends immediately when one player wins.

Given a game state, help someone new to the game determine the status of a game board. Say one of the following:

- **"Impossible"**: If it was impossible for two players to follow the rules and to have arrived at that game state.
- **"Red wins"**: If the player playing the red stones has won.
- **"Blue wins"**: If the player playing the blue stones has won.
- **"Nobody wins"**: If nobody has yet won the game. Note that a game of Hex can't end without a winner!

Note that in any impossible state, the only correct answer is "Impossible", even if red or blue has formed a connected path of stones linking the opposing sides of the board marked by his or her colors.

Here's a an example game on a 6x6 gameboard where blue won. Blue was the first player to move, and placed a blue stone at cell marked as 1. Then Red placed at cell 2, then blue at cell 3, etc. After the 11th stone is placed, blue wins.



Input

The first line of input gives the number of test cases, T . T test cases follow. Each test case start with the size of the side of the board, N . This is followed by a board of N rows and N columns consisting of only 'B', 'R' and '.' characters. 'B' indicates a cell occupied by blue stone, 'R' indicates a cell occupied by red stone, and '.' indicates an empty cell.

Output

For each test case, output one line containing "Case #x: y", where x is the case number (starting from 1) and y is the status of the game board. It can be "Impossible", "Blue wins", "Red wins" or "Nobody wins" (excluding the quotes). Note that the judge is case-sensitive, so answers of "impossible", "blue wins", "red wins" and "nobody wins" will be judged incorrect.

Limits

 $1 \leq T \leq 100.$

Small dataset

$1 \leq N \leq 10$.

Large dataset

$1 \leq N \leq 100$.

Sample

Input	Output
7	Case #1: Nobody wins
1	Case #2: Blue wins
.	Case #3: Red wins
1	Case #4: Impossible
B	Case #5: Blue wins
1	Case #6: Impossible
R	Case #7: Blue wins
2	
BR	
BB	
4	
BBBB	
BBB.	
RRR.	
RRRR	
4	
BBBB	
BBBB	
RRR.	
RRRR	
6	
.....	
..R...	
BBBBBB	
..R.R.	
..RR..	
.....	

All problem statements, input data and contest analyses are licensed under the [Creative Commons Attribution License](#).

© 2008-2017 Google [Google Home](#) - [Terms and Conditions](#) - [Privacy Policies and Principles](#)

Powered by



Google Cloud Platform

Round B China New Grad
Test 2014

[A. Sudoku Checker](#)

[B. Meet and party](#)

[C. Hex](#)

D. Dragon Maze

[E. Ignore all my comments](#)

[Questions asked](#)

Submissions

Sudoku Checker

5pt Not attempted
1471/2010 users
correct (73%)

9pt Not attempted
1146/1443 users
correct (79%)

Meet and party

9pt Not attempted
496/823 users
correct (60%)

15pt Not attempted
47/409 users
correct (11%)

Hex

12pt Not attempted
19/260 users
correct (7%)

13pt Not attempted
14/18 users correct
(78%)

Dragon Maze

8pt Not attempted
336/594 users
correct (57%)

12pt Not attempted
229/330 users
correct (69%)

Ignore all my comments

17pt Not attempted
216/468 users
correct (46%)

0pt Not attempted

Top Scores

TankEngineer	100
Nekosyndrome	100
I521530	100
W.Junqiao	100
LTzycLT	100
iloahz	100
drazil	87
navi	85
wishstudio	85
redsniper	76

Problem D. Dragon Maze

This contest is open for practice. You can try every problem as many times as you like, though we won't keep track of which problems you solve. Read the [Quick-Start Guide](#) to get started.

Small input
8 points

Solve D-small

Large input
12 points

Solve D-large

Problem

You are the prince of Dragon Kingdom and your kingdom is in danger of running out of power. You must find power to save your kingdom and its people. An old legend states that power comes from a place known as Dragon Maze. Dragon Maze appears randomly out of nowhere without notice and suddenly disappears without warning. You know where Dragon Maze is now, so it is important you retrieve some power before it disappears.

Dragon Maze is a rectangular maze, an $N \times M$ grid of cells. The top left corner cell of the maze is (0,0) and the bottom right corner is ($N-1$, $M-1$). Each cell making up the maze can be either a dangerous place which you never escape after entering, or a safe place that contains a certain amount of power. The power in a safe cell is automatically gathered once you enter that cell, and can only be gathered once. Starting from a cell, you can walk up/down/left/right to adjacent cells with a single step.

Now you know where the entrance and exit cells are, that they are different, and that they are both safe cells. In order to get out of Dragon Maze before it disappears, you must walk from the entrance cell to the exit cell taking as few steps as possible. If there are multiple choices for the path you could take, you must choose the one on which you collect as much power as possible in order to save your kingdom.

Input

The first line of the input gives the number of test cases, T . T test cases follow.

Each test case starts with a line containing two integers N and M , which give the size of Dragon Maze as described above. The second line of each test case contains four integers en_x , en_y , ex_x , ex_y , describing the position of entrance cell (en_x , en_y) and exit cell (ex_x , ex_y). Then N lines follow and each line has M numbers, separated by spaces, describing the $N \times M$ cells of Dragon Maze from top to bottom. Each number for a cell is either -1, which indicates a cell is dangerous, or a positive integer, which indicates a safe cell containing a certain amount of power.

Output

For each test case, output one line containing "Case #x: y", where x is the case number (starting from 1). If it's possible for you to walk from the entrance to the exit, y should be the maximum total amount of power you can collect by taking the fewest steps possible. If you cannot walk from the entrance to the exit, y should be the string "Mission Impossible." (quotes for clarity). Please note that the judge requires an exact match, so any other output like "mission impossible." or "Mission Impossible" (which is missing the trailing period) will be judged incorrect.

Limits

The amount of power contained in each cell will not exceed 10,000.

$1 \leq T \leq 30$.

$0 \leq en_x, ex_x < N$.

$0 \leq en_y, ex_y < M$.

Small dataset

$1 \leq N, M \leq 10$.

Large dataset

$1 \leq N, M \leq 100$.

Sample

Input	Output
2	Case #1: Mission Impossible.
2 3	Case #2: 7
0 2 1 0	
2 -1 5	

```
3 -1 6
4 4
0 2 3 2
-1 1 1 2
1 1 1 1
2 -1 -1 1
1 1 1 1
```

All problem statements, input data and contest analyses are licensed under the [Creative Commons Attribution License](#).

© 2008-2017 Google [Google Home](#) - [Terms and Conditions](#) - [Privacy Policies and Principles](#)

Powered by



Google Cloud Platform

Round B China New Grad Test 2014

[A. Sudoku Checker](#)

[B. Meet and party](#)

[C. Hex](#)

[D. Dragon Maze](#)

E. Ignore all my comments

[Questions asked](#)

Submissions

Sudoku Checker

5pt	Not attempted 1471/2010 users correct (73%)
9pt	Not attempted 1146/1443 users correct (79%)

Meet and party

9pt	Not attempted 496/823 users correct (60%)
15pt	Not attempted 47/409 users correct (11%)

Hex

12pt	Not attempted 19/260 users correct (7%)
13pt	Not attempted 14/18 users correct (78%)

Dragon Maze

8pt	Not attempted 336/594 users correct (57%)
12pt	Not attempted 229/330 users correct (69%)

Ignore all my comments

17pt	Not attempted 216/468 users correct (46%)
0pt	Not attempted

Top Scores

TankEngineer	100
Nekosyndrome	100
I521530	100
WJunqiao	100
LTzycLT	100
iloahz	100
drazil	87
navi	85
wishstudio	85
redsniper	76

Problem E. Ignore all my comments

This contest is open for practice. You can try every problem as many times as you like, though we won't keep track of which problems you solve. Read the [Quick-Start Guide](#) to get started.

Small input
17 points

Solve E-small

Large input
0 points

Solve E-large

Problem

Good programmers write fabulous comments. Igor is a programmer and he likes the old C-style comments in `/* ... */` blocks. For him, it would be ideal if he could use this style as a uniform comment format for all programming languages or even documents, for example Python, Haskell or HTML/XML documents.

Making this happen doesn't seem too difficult to Igor. What he will need is a comment pre-processor that removes all the comment blocks in `/*`, followed by comment text, and by another `*/`. Then the processed text can be handed over to the compiler/document renderer to which it belongs—whatever it is.

Igor's pre-processor isn't quite that simple, though. Here are some cool things it does:

- The comments the pre-processor reads can be nested the same way brackets are nested in most programming languages. It's possible to have comments inside comments. For example, the following code block has an outer level of comments that should be removed by the comment pre-processor. The block contains two inner comments.

```
printf("Hello /* a comment /* a comment inside comment */
        inside /* another comment inside comment */
        string */ world");
```

After the pre-process step, it becomes:

```
printf("Hello world");
```

- Igor recognizes comments can appear anywhere in the text, including inside a string `"/ * . . . */"`, a constant number `12 / * . . . */ 34` or even in a character escape `\\ * . . . */ \n`

Or more formally:

```
text:
  text-piece
  text-piece remaining-text
text-piece:
  char-sequence-without-/*
  empty-string
remaining-text:
  comment-block text

comment-block:
  /* comment-content */
comment-content:
  comment-piece
  comment-piece remaining-comment
comment-piece:
  char-sequence-without-/*-or-*/
  empty-string
remaining-comment:
  comment-block comment-content

char:
  letters
  digits
  punctuations
  whitespaces
```

Our pre-processor, given a **text**, removes all **comment-block** instances as specified.

Notes

- Igor only needs to remove the comment in one pass. He doesn't remove additional comment blocks created as a result of the removal of any comment

block. For example:

```
/**no recursion*/ file header */
```

should generate:

```
/* file header */
```

- The `*` character in any `/*` or `*/` cannot be re-used in another `/*` or `*/`. For example the following does **NOT** form a proper comment block

```
/*/
```

Input

A text document with comment blocks in `/*` and `*/`. The input file is valid. It follows the specification of **text** in the problem statement. The input file always terminates with a newline symbol.

Output

We only have one test case for this problem. First we need to output the following line.

```
Case #1:
```

Then, print the document with all comments removed, in the way specified in the problem statements. Don't remove any spaces or empty lines outside comments.

Limits

The input program contains only:

- Letters: a-z, A-Z,
- Digits: 0-9
- Punctuation: ~ ! @ # % ^ & * () - + = : ; " ' < > , . ? | / \ { } [] _
- Whitespace characters: space, newline

Small dataset

The small input contains a program of less than 2k bytes.

Large dataset

The large input contains a program of less than 100k bytes.

Sample

Input

```
/**no recursion*/ file header
*****
* Sample input program *
*****
*/
int spawn_workers(int worker_count) {
    /* The block below is supposed to spawn 100 workers.
       But it creates many more.
       Commented until I figure out why.
       for (int i = 0; i < worker_count; ++i) {
           if(!fork()) {
               /* This is the worker. Start working. */
               do_work();
           }
       }
    */
    return 0; /* successfully spawned 100 workers */
}

int main() {
    printf("Hello /*a comment inside string*/ world");
    int worker_count = 0/*octal number*/144;
    if (spawn_workers(worker_count) != 0) {
        exit(-1);
    }
    return 0;
}
```

Output

```
Case #1:
/* file header
*****
*/
int spawn_workers(int worker_count) {
    return 0;
}

int main() {
    printf("Hello world");
    int worker_count = 0144;
    if (spawn_workers(worker_count) != 0) {
        exit(-1);
    }
    return 0;
}
```

All problem statements, input data and contest analyses are licensed under the [Creative Commons Attribution License](#).

© 2008-2017 Google [Google Home](#) - [Terms and Conditions](#) - [Privacy Policies and Principles](#)

Powered by



Google Cloud Platform