

Round C APAC Test

A. Minesweeper

[B. Taking Metro](#)

[C. Broken Calculator](#)

[D. Tetris](#)

Questions asked 1

Submissions

Minesweeper

8pt Not attempted
748/1510 users
correct (50%)

14pt Not attempted
703/741 users
correct (95%)

Taking Metro

9pt Not attempted
64/319 users
correct (20%)

15pt Not attempted
56/62 users correct
(90%)

Broken Calculator

10pt Not attempted
570/1000 users
correct (57%)

16pt Not attempted
385/544 users
correct (71%)

Tetris

11pt Not attempted
29/163 users
correct (18%)

17pt Not attempted
27/29 users correct
(93%)

Top Scores

cebrusfs	100
LXZ	100
Kriiii	100
drazil	100
xing89qs	100
xhae	100
culaucon	100
whsb	100
jki14	100
zck921031	100

Problem A. Minesweeper

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 A-small

Large input
14 points

Solve A-large

Problem

Minesweeper is a computer game that became popular in the 1980s, and is still included in some versions of the Microsoft Windows operating system. This problem has a similar idea, but it does not assume you have played Minesweeper.

In this problem, you are playing a game on a grid of identical cells. The content of each cell is initially hidden. There are M mines hidden in M different cells of the grid. No other cells contain mines. You may click on any cell to reveal it. If the revealed cell contains a mine, then the game is over, and you lose. Otherwise, the revealed cell will contain a digit between 0 and 8, inclusive, which corresponds to the number of neighboring cells that contain mines. Two cells are neighbors if they share a corner or an edge. Additionally, if the revealed cell contains a 0, then all of the neighbors of the revealed cell are automatically revealed as well, recursively. When all the cells that don't contain mines have been revealed, the game ends, and you win.

For example, an initial configuration of the board may look like this ('*' denotes a mine, and 'c' is the first clicked cell):

```
*..*...**
...*....
...C*....
.....*
.....*
```

There are no mines adjacent to the clicked cell, so when it is revealed, it becomes a 0, and its 8 adjacent cells are revealed as well. This process continues, resulting in the following board:

```
*..*...**
1112*....
00012*...
00001111*
00000001..
```

At this point, there are still un-revealed cells that do not contain mines (denoted by '.' characters), so the player has to click again in order to continue the game.

You want to win the game as quickly as possible. You want to find the minimum number of clicks to win the game. Given the size of the board ($N \times N$), output such minimum number of clicks.

Input

The first line of the input gives the number of test cases, T . T test cases follow. First line of each test case contains one integer N . N lines strings with length N follows containing '*' and '.', denotes the Minesweeper initial board.

Output

For each test case, output one line containing "Case #x: y", where x is the test case number (starting from 1) and y is the minimum number of clicks to win.

Limits

$1 \leq T \leq 100$.

Small dataset

$1 \leq N \leq 50$.

Large dataset

$1 \leq N \leq 300$.

Sample

Input	Output
-------	--------

2	Case #1: 2
3	Case #2: 8

```
..*  
..*  
**.  
5  
..*..  
..*..  
.*.*  
.*..  
.*..  
.*..
```

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 C APAC Test

[A. Minesweeper](#)

B. Taking Metro

[C. Broken Calculator](#)

[D. Tetris](#)

Questions asked 1

Submissions

Minesweeper

8pt	Not attempted 748/1510 users correct (50%)
14pt	Not attempted 703/741 users correct (95%)

Taking Metro

9pt	Not attempted 64/319 users correct (20%)
15pt	Not attempted 56/62 users correct (90%)

Broken Calculator

10pt	Not attempted 570/1000 users correct (57%)
16pt	Not attempted 385/544 users correct (71%)

Tetris

11pt	Not attempted 29/163 users correct (18%)
17pt	Not attempted 27/29 users correct (93%)

Top Scores

cebrusfs	100
LXZ	100
Kriiii	100
drazil	100
xing89qs	100
xhae	100
culaucon	100
whsb	100
jki14	100
zck921031	100

Problem B. Taking Metro

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

Tom is taking metros in the city to go from station to station.

The metro system in the city works like this:

- There are **N** metro lines in the city: line 1, line 2, ..., line **N**.
- For each metro **i**, there are **SN_i** stations. Let's assume they are **S_{i,1}**, **S_{i,2}**, ..., **S_{i,SN_i}**. These stations are ordered from one end point to the other end point. The metro is running in both directions. In other words, the metro is going from **S_{i,1}** -> **S_{i,2}** -> ... -> **S_{i,SN_i}**, and **S_{i,SN_i}** -> **S_{i,SN_i-1}** -> ... -> **S_{i,1}**. You can take the metro from any station and get off at any station. It takes a certain time to travel from one station to the next station. It takes **Time_{i,1}** minutes to travel from **S_{i,1}** to **S_{i,2}**, **Time_{i,2}** minutes to travel from **S_{i,2}** to **S_{i,3}**, etc. It takes the same time in the other direction.
- There are **M** transfer tunnels. Each transfer tunnel connects two stations of different metro lines. It takes a certain amount of time to travel through a tunnel in either direction. You can get off the metro at one end of the tunnel and walk through the tunnel to the station at the another end.
- When you arrive at a metro station of line **i**, you need to wait **W_i** minutes for the next metro.

Now, you are going to travel from one station to another. Find out the shortest time you need.

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 number of metro lines. **N** metros descriptions follow. Each metro description starts with two integers **SN_i** and **W_i**, the number of stations and the expected waiting time in minutes. The next line consists of **SN_i-1** integers, **Time_{i,1}**, **Time_{i,2}**, ..., **Time_{i,SN_i-1}**, describing the travel time between stations.

After the metro descriptions, there is an integer **M**, the number of tunnels. **M** lines follow to describe the tunnels. Each tunnel description consists of 5 integers, **m1_i**, **s1_i**, **m2_i**, **s2_i**, **t_i** which means the tunnel is connecting stations **S_{m1_i,s1_i}** and station **S_{m2_i,s2_i}**. The walking time of the tunnel is **t_i**.

The next line contains an integer **Q**, the number of queries. Each of the next **Q** lines consists of 4 integers, **x1**, **y1**, **x2**, **y2**, which mean you are going to travel from station **S_{x1,y1}** to station **S_{x2,y2}**.

Output

For each test case, output one line containing "Case #x:", where x is the test case number (starting from 1), then followed by **Q** lines, each line containing an integer y which is the shortest time you need for that query. If it's impossible, output -1 for that query instead.

Limits

- $1 \leq T \leq 100$.
- $1 \leq W_i \leq 100$.
- $1 \leq \text{Time}_{i,j} \leq 100$.
- $1 \leq m1_i \leq N$.
- $1 \leq s1_i \leq SN_{m1_i}$.
- $1 \leq m2_i \leq N$.
- $1 \leq s2_i \leq SN_{m2_i}$.
- m1_i** and **m2_i** will be different.
- $1 \leq t_i \leq 100$.
- $1 \leq Q \leq 10$.
- $1 \leq x1 \leq N$.
- $1 \leq y1 \leq SN_{x1}$.

$1 \leq x_2 \leq N$.
 $1 \leq y_2 \leq SN_{y_2}$.
 Station S_{x_1, y_1} and station S_{x_2, y_2} will be different.

Small dataset

$1 \leq N \leq 10$.
 $0 \leq M \leq 10$.
 $2 \leq SN_i \leq 100$.
 The total number of stations in each case is at most 100.

Large dataset

$1 \leq N \leq 100$.
 $0 \leq M \leq 100$.
 $2 \leq SN_i \leq 1000$.
 The total number of stations in each case is at most 1000.

Sample

Input	Output
2	Case #1:
	11
2	Case #2:
5 3	18
3 5 7 3	
4 2	
1 1 1	
1	
1 2 2 2 1	
1	
1 1 2 4	
2	
5 3	
3 5 7 3	
4 2	
1 1 1	
2	
1 2 2 2 1	
2 4 1 4 1	
1	
1 1 1 5	

In the first case, you are going to travel from station 1 of metro line 1 to station 4 of metro line 2. The best way is:

- wait 3 minutes for line 1 and get on it.
- take it for 3 minutes and get off at station 2.
- take the tunnel and walk for 1 minute to station 2 of line 2.
- wait 2 minutes for line 2 and get on it.
- take it for 2 minutes and get off at station 4.

The total time is: $3+3+1+2+2=11$.

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 C APAC Test

[A. Minesweeper](#)

[B. Taking Metro](#)

C. Broken Calculator

[D. Tetris](#)

Questions asked 1

Submissions

Minesweeper

8pt	Not attempted 748/1510 users correct (50%)
14pt	Not attempted 703/741 users correct (95%)

Taking Metro

9pt	Not attempted 64/319 users correct (20%)
15pt	Not attempted 56/62 users correct (90%)

Broken Calculator

10pt	Not attempted 570/1000 users correct (57%)
16pt	Not attempted 385/544 users correct (71%)

Tetris

11pt	Not attempted 29/163 users correct (18%)
17pt	Not attempted 27/29 users correct (93%)

Top Scores

cebrusfs	100
LXZ	100
Kriiii	100
drazil	100
xing89qs	100
xhae	100
culaucon	100
whsb	100
jki14	100
zck921031	100

Problem C. Broken Calculator

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
10 points

Solve C-small

Large input
16 points

Solve C-large

Problem

Alice is a smart student who is very good at math. She is attending a math class. In this class, the teacher is teaching the students how to use a calculator. The teacher will tell an integer to all of the students, and the students must type that exact number into their calculators. If someone fails to type the number, he or she will be punished for failing such an easy task!

Unfortunately, at the beginning of the class, Alice finds that her calculator is broken! She finds that some of the number buttons are totally broken, and only the "multiply" and "equals" operator buttons are available to use. So she can only use these buttons to get the number quickly.

For instance, the teacher may say the number "60", while Alice's calculator can only type "1", "2" and "5". She could push the following buttons:

- Button "15" (2 clicks)
- Button "multiply" (1 click)
- Button "2" (1 click)
- Button "multiply" (1 click)
- Button "2" (1 click)
- Button "equals" (1 click)

This method requires 7 button clicks. However, if Alice uses "12*5=", only 5 clicks are needed. Of course Alice wants to get the integer as fast as possible, so she wants to minimize the number of button clicks. Your task is to help her find a way to get the required number quickly.

Input

The first line of the input gives a number **T**, the number of integers the teacher says. **T** test cases follow.

Each case contains two lines. The first line contains ten numbers each of which is only 0 or 1. the *i*th number (starting from 0) is "1" if the number *i* can be clicked, or "0" if it is broken. The second line contains only one number **X**, the integer the teacher tells everyone.

Output

For each test case, output one line containing "Case #x: y", where x is the test case number (starting from 1) and y is the minimum number of button clicks needed, or "Impossible" if it is not possible to produce the number.

Limits

$1 \leq T \leq 100$.

Small dataset

$1 \leq X \leq 100$.

Large dataset

$1 \leq X \leq 10^6$.

Sample

Input	Output
3	Case #1: 5
0 1 1 0 0 1 0 0 0 0	Case #2: 4
60	Case #3: Impossible
1 1 1 1 1 1 1 1 1 1	
128	
0 1 0 1 0 1 0 1 0 1	
128	

The first sample case is explained in problem statement.

In the second case, all digits are available, so Alice can just press "1", "2", "8" and then "equals" to get the result. Please note that she still needs to press "equals" in the last step, even though there are no calculations.

For the last case, it's impossible since Alice cannot input any even numbers.

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 C APAC Test

[A. Minesweeper](#)

[B. Taking Metro](#)

[C. Broken Calculator](#)

D. Tetris

Questions asked 1

Submissions

Minesweeper

8pt Not attempted
748/1510 users
correct (50%)

14pt Not attempted
703/741 users
correct (95%)

Taking Metro

9pt Not attempted
64/319 users
correct (20%)

15pt Not attempted
56/62 users correct
(90%)

Broken Calculator

10pt Not attempted
570/1000 users
correct (57%)

16pt Not attempted
385/544 users
correct (71%)

Tetris

11pt Not attempted
29/163 users
correct (18%)

17pt Not attempted
27/29 users correct
(93%)

Top Scores

cebrusfs	100
LXZ	100
Kriiii	100
drazil	100
xing89qs	100
xhae	100
culaucon	100
whsb	100
jki14	100
zck921031	100

Problem D. Tetris

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
11 points

Solve D-small

Large input
17 points

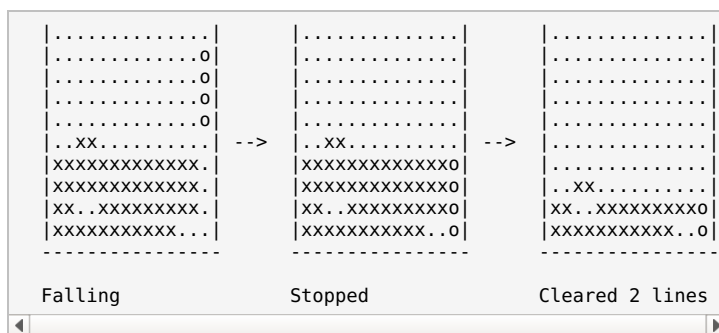
Solve D-large

Problem

Tetris is a famous video game that almost everyone has played it. In this problem, you need to simulate a simplified version of it.

In our version, the game is played in a **W** by **H** field with gravity. At the beginning, the field is empty. Then the tetrominos start to fall from above top of the field to bottom of the field, one by one. Each tetromino will stop as soon as it touches some other tetrominos or bottom of the field.

One interesting feature of the game is called "line clearing". A line will be cleared as soon as it is filled by tetrominos. More than one line may be cleared at a time. For example:



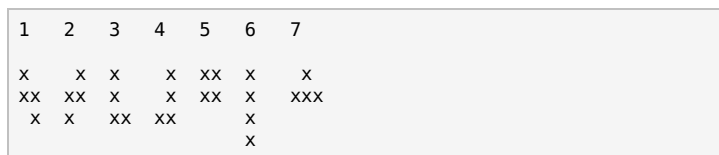
Note that in this simplified version, the "floating" tetromino blocks won't continue to fall after lines are cleared. This is why the top-most two squares will keep in such position. Consequently, cascade clearing won't happen, even though it would happen in the original version of Tetris.

The game ends when all the given tetrominos are placed, or the current tetromino cannot be placed due to the height limit of the field is reached.

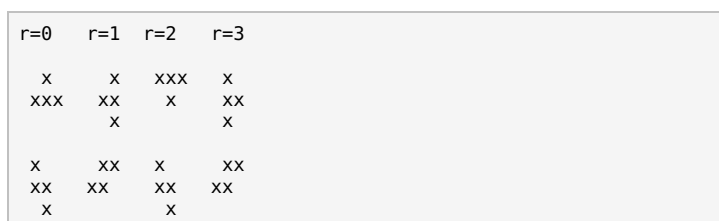
In this problem, each tetromino will has its type, rotation and falling position told by the input. They will start to fall from the **above** of the field. Your goal is to simulate and get the final result of each play.

Input

We have 7 types of tetromino:



Rotation of a tetromino is represented by a number **r**. **r** can be 0, 1, 2 or 3. Rotation is counterclockwise. For example:



The horizontal falling position is represented by a number **x**. It is the coordinate of the lower left square of a tetromino's bounding box. Here **x** starts from 0.

The first line of the input gives the number of test cases, **T**. For each test case, the first line of input has 3 integers, **W**, **H**, **N**. **W** is the width, **H** is the height

and **N** is the number of blocks that are going to fall.

Then **N** lines below, each line has 3 integers, **t_i**, **r_i**, **x_i**. **t_i** tells the tetromino type. **r_i** is the rotation of this tetromino. **x_i** is the horizontal falling position of this tetromino. It is guaranteed that **x_i** will make the tetromino inside the field, horizontally.

Output

For each test case, first output one line containing "Case #i:", where **i** is the test case number (starting from 1). And then, if the game ends before the **N** blocks, output "Game Over!"(without quotes). Otherwise, output the game field's final state, which should have **H** lines, each has **W** characters. Each character can be '.' or 'x'.

Limits

1 ≤ **T** ≤ 100

1 ≤ **t_i** ≤ 7

0 ≤ **r_i** < 4

Small dataset

4 ≤ **W** ≤ 20

1 ≤ **H** ≤ 20

0 ≤ **N** ≤ 100

Large dataset

4 ≤ **W** ≤ 100

1 ≤ **H** ≤ 100

0 ≤ **N** ≤ 5000

Sample

Input	Output
5	Case #1:
8 6 1
1 0 0
5 4 1
1 1 1	x.....
5 6 3	xx.....
5 0 0	.x.....
5 0 2	Case #2:
3 2 3
6 4 3
6 2 0	..xx.
6 2 0	.xx..
6 2 0	Case #3:
6 4 2
6 0 0
6 0 1

	...xx
	Case #4:
	Game Over!
	Case #5:
	xx....
	xx....
	xx....
	xx....

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