Problem A. Quake Live

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 1 5 points	Solve A-small-1
Small input 2 10 points	Solve A-small-2

Problem

Quake Live is a first-person shooter game that supports several types of matches. One of the most popular types is the 4-on-4 team deathmatch. Eight players enter. They are split into two teams of 4. The teams fight to the death. The team whose players all die first loses.

The Quake Live servers maintain a history of matches for each player, which is used to estimate each player's *skill level* -- an integer between 1 and 1000. To keep the game as fair as possible, whenever 8 players connect to the server, the server assigns players to teams to keep the skill balance as fair as possible. To do this, the server looks at the skill levels of the 8 players and finds a way to split the players into two teams of 4 in a way that minimizes the difference between the sum of skills on team A and the sum of skills on team B.

You think that something is fishy in this logic and would like to verify that the server is doing its job correctly. Given the skill levels of the players who enter, can you find the smallest possible difference between the total team skills? Note that the two teams must always have the same number of players.

Input

The first line of the input gives the number of test cases, **T**. **T** lines follow. Each line starts with the integer **N** -- the number of players who enter. The next **N** integers on the line are the skill levels of the players in no particular order.

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 smallest possible difference between the sum of skill levels of the players on team A and the sum of skill levels of the players on team B.

Limits

$1 \le \mathbf{T} \le 100$.

Small dataset

N = 8.

Large dataset

N could be 2, 4, 6 or 8.

Sample

Input

```
3
8 1 2 3 4 5 6 7 8
8 1 1 1 1 1 1 1 1000
2 13 17
```

Output

```
Case #1: 0
Case #2: 999
Case #3: 4
```

EuroPython 2012

A. Quake Live
B. Shoot the Target
C. Bejeweled Befuddlement
D. Technology Planning

Questions asked

Submissions

Quake Live
5pt Not attempted 30/48 users correct (63%)
10pt Not attempted 28/28 users correct (100%)

Shoot the Target
8pt Not attempted 1/4 users correct (25%)
12pt Not attempted 0/1 users correct (0%)

Bejeweled Befuddlement
10pt Not attempted 1/3 users correct (33%)
20pt Not attempted 0/1 users correct (0%)

Technology Planning
15pt Not attempted 13/16 users correct (81%)
20pt Not attempted 12/13 users correct (92%)

Top Scores

andreidid	50
tlotze	50
alexamici	50
errebepe	50
almost	50
pts	50
bucko	50
r3m0t	43
eseriva	35
mumino	35

EuroPython 2012

A. Quake Live
B. Shoot the Target
C. Bejeweled Befuddlement
D. Technology Planning

Questions asked

Submissions

Quake Live

5pt	Not attempted **30/48 users** correct (63%)
10pt	Not attempted **28/28 users** correct (100%)

Shoot the Target

8pt	Not attempted **1/4 users** correct (25%)
12pt	Not attempted **0/1 users** correct (0%)

Bejeweled Befuddlement

10pt	Not attempted **1/3 users** correct (33%)
20pt	Not attempted **0/1 users** correct (0%)

Technology Planning

15pt	Not attempted **13/16 users** correct (81%)
20pt	Not attempted **12/13 users** correct (92%)

Top Scores

andreidid	50
tlotze	50
alexamici	50
errebepe	50
almost	50
pts	50
bucko	50
r3m0t	43
eseriva	35
mumino	35

Problem B. Shoot the Target

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 1 8 points	Solve B-small-1
Small input 2 12 points	Solve B-small-2

Problem

Kazuki is playing Worms. There is a target above the ground which he has to hit. However, it is slanted so that it is not perpendicular to the ground. Kazuki is allowed to shoot from anywhere on the ground and he wants to find the best place to stand to maximize the visible angle of the target. Find this maximum visible angle in degrees.

The world is a 2D plane, with the y-axis pointing up towards the sky. The ground can be considered as the x-axis. You will be given the coordinates of the two endpoints of the target -- **(X1, Y1)** and **(X2, Y2)**. The target is a line segment between those two points.

Note that the visible angle of the target that Kazuki wants to maximize is the angle formed by the points **(X1, Y1)**, **(X, 0)** and **(X2, Y2)**. The optimal **X** will not necessarily be an integer.

Input

The first line of the input gives the number of test cases, **T**. **T** lines follow. Each lines consists of 4 integers **X1**, **Y1**, **X2** and **Y2** separated by spaces, where **(X1, Y1)** and **(X2, Y2)** are the coordinates of the two ends of the target.

Output

For each test case, output one line containing "Case #t: Z", where t is the case number (starting from 1) and **Z** is the maximum visible angle of the target in degrees that Kazuki can achieve. Answers accurate to within an absolute or relative error of 10^{-4} will be accepted.

Limits

1 ≤ **T** ≤ 100.

X1 ≠ X2

Small dataset

1 ≤ **Y1, Y2** ≤ 10

-10 ≤ **X1, X2** ≤ 10

Large dataset

1 ≤ **Y1, Y2** ≤ 100000

-100000 ≤ **X1, X2** ≤ 100000

Sample

Input	Output
3 1 1 3 1 -2 1 -1 2 0 2 3 3	Case #1: 90 Case #2: 45 Case #3: 63.8059

Submissions

Quake Live

5pt Not attempted
30/48 users correct (63%)

10pt Not attempted
28/28 users correct (100%)

Shoot the Target

8pt Not attempted
1/4 users correct (25%)

12pt Not attempted
0/1 users correct (0%)

Bejeweled Befuddlement

10pt Not attempted
1/3 users correct (33%)

20pt Not attempted
0/1 users correct (0%)

Technology Planning

15pt Not attempted
13/16 users correct (81%)

20pt Not attempted
12/13 users correct (92%)

Top Scores

andreidid	50
tlotze	50
alexamici	50
errebepe	50
almost	50
pts	50
bucko	50
r3m0t	43
eseriva	35
mumino	35

Problem C. **Bejeweled Befuddlement**

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 1 10 points	Solve C-small-1
Small input 2 20 points	Solve C-small-2

Problem

John has been playing a lot Bejeweled lately, but he's becoming frustrated at his inability to get high scores. He would like a program to help him play, and you can help!

In Bejeweled, you have a grid of gems of various colors, such as the one below:

```
BGR
GRR
BGY
```

You make a move by switching two gems that are horizontally or vertically connected to each other. So, for example, you could switch the first two gems in the second row, changing the board to this:

```
BGR
RGR
BGY
```

If when you do this, any three gems in a row are the same color, they all disappear at once. So in this example, the three green gems would disappear.

```
B.R
R.R
B.Y
```

It can get a little more complex, though. Additionally, multiple sets of three (or more) can be formed at once. For example:

```
RGOY
RGOG
OOYO
RGOG
YBOB
```

In this case, swapping the right-most orange in the middle row with the yellow next to it...

```
RGOY
RGOG
OOOY
RGOG
YBOB
```

... makes a row of 3 and a column of 5, which all disappear...

```
RG.Y
RG.G
...Y
RG.G
YB.B
```

... and then, any gems that are above an open space fall down to fill the empty spaces...

```
...Y
RG.G
RG.Y
RG.G
YB.B
```

... and then any newly formed rows of three or more will also disappear:

```
...Y
...G
...Y
...G
YB.B
```

This continues until there are no more rows or columns of three or more similar colored gems formed. So, more generally, the process is as follows:

- Switch two gems
- While there are any rows or columns of three or more gems of the same color
  - Remove those gems
  - Move any gems that are above empty spaces down until there are no gems above empty spaces

Note: In the actual game, gems fall in from the top to replace removed gems. You do not need to take this into account.

Input

The first line of input will contain the number of test cases, **T**. Each test case will start with a line containing two integers, **N** and **M**, separated by a space. The next **N** lines will each contain exactly **M** uppercase letters, each representing one gem. Gems of the same letter are considered to be of the same color.

Output

For each test case, output a line of the form "Case #**C**: **D**", where **C** is the number of the test case, starting from 1, and **D** is the maximum number of gems that can be removed as a result of a single swap.

Limits

All gem characters will be upper-case letters.
The input grid will not contain a row or a column of 3 or more gems of the same color appearing consecutively.

Small dataset

1 ≤ **T** ≤ 20.
3 ≤ **number of rows** ≤ 10.
3 ≤ **number of columns** ≤ 10.

Large dataset

1 ≤ **T** ≤ 100.
1 ≤ **number of rows** ≤ 50.
1 ≤ **number of columns** ≤ 50.

Sample

Input

```
3
3 3
BGR
GRR
BGY
5 4
RGOY
RGOG
OOYO
RGOG
YBOB
3 3
ABC
DEF
GHI
```

Output

```
Case #1: 3
Case #2: 13
Case #3: 0
```

Submissions

Quake Live

5pt Not attempted **30/48 users** correct (63%)
10pt Not attempted **28/28 users** correct (100%)

Shoot the Target

8pt Not attempted **1/4 users** correct (25%)
12pt Not attempted **0/1 users** correct (0%)

Bejeweled Befuddlement

10pt Not attempted **1/3 users** correct (33%)
20pt Not attempted **0/1 users** correct (0%)

Technology Planning

15pt Not attempted **13/16 users** correct (81%)
20pt Not attempted **12/13 users** correct (92%)

Top Scores

andreidid	50
tlotze	50
alexamici	50
errebepe	50
almost	50
pts	50
bucko	50
r3m0t	43
eseriva	35
mumino	35

Problem D. Technology Planning

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 1
15 points
Solve D-small-1

Small input 2
20 points
Solve D-small-2

Problem

You are playing a culture simulation game, in which your culture can develop various technologies. Some technologies depend upon others; if technology A depends upon technology B, you cannot develop B until you have developed A. Your culture can work on only one technology at a time.

In the game, you have goals which require particular technologies. You have decided to write a program to help you by planning out the order in which to develop technologies.

Input

The input to your program starts with the number of test cases, **T**, on a line by itself. **T** test cases follow. Each one consists of:

- One line with an integer, **M**, which is the number of technological dependencies.
- **M** lines, each one containing two technology names, separated by a colon (':'). The first technology depends on the second one.
- One line with an integer, **Q**, which is the number of goal technologies.
- **Q** lines, each naming a goal technology.

Each technology name is a sequence of alphanumeric characters (letters or numbers). Technology names are case-sensitive.

Output

The output for each case should start with a line of the form "Case #**C**: **D**", where **C** is the case number, starting from 1, and **D** is the smallest possible number of technologies that have to be researched. The next **D** lines should each contain one technology, in the order that they need to be researched.

If there are several possible correct orderings, any one of them is acceptable.

Limits

T ≤ 25
M ≥ 1
Q ≥ 1
There will be no cycles in the dependency graph.

Small dataset

M ≤ 10
Q ≤ 10.

Large dataset

M ≤ 100
Q ≤ 100.

Sample

Input

```
4
9
Metalworking:Fire
Pottery:Fire
Iron:Metalworking
Steel:Iron
Battleships:Steel
Alchemy:Fire
Chemistry:Alchemy
Explosives:Chemistry
Battleships:Explosives
1
Battleships

9
Metalworking:Fire
```

Output

```
Case #1: 8
Fire
Metalworking
Iron
Steel
Alchemy
Chemistry
Explosives
Battleships
Case #2: 3
Fire
Pottery
Metalworking

Case #3: 1
d
```

```
Pottery:Fire
Iron:Metalworking
Steel:Iron
Battleships:Steel
Alchemy:Fire
Chemistry:Alchemy
Explosives:Chemistry
Battleships:Explosives
2
Metalworking
Pottery
2
b:a
c:b
1
d
1
A:a
1
A
```

```
Case #4: 2
a
A
```