Submissions

Ambiguous Cipher

7pt Not attempted **813/966 users** correct (84%)

12pt Not attempted **683/755 users** correct (90%)

X Squared

9pt Not attempted **377/706 users** correct (53%)

14pt Not attempted **319/358 users** correct (89%)

Magical Thinking

6pt Not attempted **570/621 users** correct (92%)

19pt Not attempted **149/325 users** correct (46%)

The 4M Corporation

11pt Not attempted **109/194 users** correct (56%)

22pt Not attempted **60/78 users** correct (77%)

Top Scores

ACMonster	100
subscriber	100
Kasugano.Sora	100
spnautilus	100
1717374	100
Benq	100
LeeSin	100
yubowenok	100
praran26	100
cephian	100

Problem A. Ambiguous Cipher

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 7 points	Solve A-small
Large input 12 points	Solve A-large

Problem

Susie and Calvin are classmates. Calvin would like to be able to pass notes to Susie in class without their teacher or other classmates knowing what they are talking about, just in case the notes fall into the wrong hands. Calvin has devised the a system to encrypt his messages.

Calvin only passes one word to Susie each time, and that word consists of only uppercase letters, because Calvin is so excited to talk to Susie. Each word is encrypted as follows:

- Calvin assigns a number to each letter based on the letter's position in the alphabet, where A = 0, B = 1, ..., Z = 25.
- For every letter in the word, Calvin determines the encrypted value of the letter by summing the values of the 1 or 2 letter(s) that are adjacent to that letter in the word. He takes that sum modulo 26, and this is the new value of the letter. Calvin then converts the value back to an uppercase letter based on positions in the alphabet, as before.
- The encrypted word is determined by encrypting every letter in the word using this method. Each letter's encryption is based only on the letters from the original unencrypted message, and not on any letters that have already been encrypted

Let's take a look at one of the notes Calvin is writing for Susie. Since Calvin is always hungry, he wants to let Susie know that he wants to eat again. Calvin encrypts the word `SOUP` as follows:

- S = 18, O = 14, U = 20, and P = 15.
- Calvin encrypts each letter based on the values of its neighbor(s):
  - First letter: 14 mod 26 = 14.
  - Second letter: (18 + 20) mod 26 = 12.
  - Third letter: (14 + 15) mod 26 = 3.
  - Fourth letter: 20 mod 26 = 20.
- The values 14 12 3 20 correspond to the letters `OMDU`, and this is the encrypted word that Calvin will write on the note for Susie.

It is guaranteed that Calvin will not send Susie any words that cannot be decrypted at all. For example, Calvin would not send Susie the word `APE`, since it does not have any valid decryptions. (That is, there is no word that Calvin could have encrypted to `APE`.)

However, Calvin's system is not perfect, and some of the words he sends Susie can actually be decrypted to multiple words, creating ambiguity! For example, `BCB` can be decrypted to `ABC` or `CBA`, among other possibilities.

Susie pulled another all-nighter yesterday to finish school projects, and she is too tired to decrypt Calvin's messages. She needs your help!

Input

The first line of the input gives the number of test cases, **T**. **T** test cases follow. Each case is a single line that contains a string **W** of uppercase letters: an encrypted word that Calvin has sent.

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 decrypted word, or `AMBIGUOUS` if it is impossible to uniquely determine the decrypted word.

Limits

1 ≤ **T** ≤ 100.
W consists of only of uppercase English letters.
W is decryptable to one or more words. (That is, **W** is the result of an encryption of some word.)
W does not decrypt to the word `AMBIGUOUS`. (You will only output that when the decryption is ambiguous.)

Small dataset

2 ≤ the length of **W** ≤ 4.

Large dataset

2 ≤ the length of **W** ≤ 50.

Sample

Input	Output

`3`
`OMDU`
`BCB`
`AOAAAN`
	`Case #1: SOUP`
	`Case #2: AMBIGUOUS`
	`Case #3: BANANA`

Note that the last sample case would not appear in the Small dataset.

Sample Cases #1 & #2 were explained in the problem statement.

In Sample Case #3, `BANANA` is the only word that encrypts to `AOAAAN`.

Kickstart Round C 2017

A. Ambiguous Cipher
B. X Squared
C. Magical Thinking
D. The 4M Corporation

Contest Analysis

Questions asked 2

Submissions

Ambiguous Cipher
7pt Not attempted **813/966 users** correct (84%)
12pt Not attempted **683/755 users** correct (90%)

X Squared
9pt Not attempted **377/706 users** correct (53%)
14pt Not attempted **319/358 users** correct (89%)

Magical Thinking
6pt Not attempted **570/621 users** correct (92%)
19pt Not attempted **149/325 users** correct (46%)

The 4M Corporation
11pt Not attempted **109/194 users** correct (56%)
22pt Not attempted **60/78 users** correct (77%)

Top Scores

ACMonster	100
subscriber	100
Kasugano.Sora	100
spnautilus	100
1717374	100
Benq	100
LeeSin	100
yubowenok	100
praran26	100
cephian	100

Problem B. X Squared

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 14 points	Solve B-large

Problem

The hot new toy for this year is called "X Squared". It consists of a square **N** by **N** grid of tiles, where **N** is odd. Exactly 2 × **N** - 1 of the tiles are labeled with an X, and the rest are blank (which we will represent with the `.` character). In each move of the game, the player can either choose and exchange two rows of tiles, or choose and exchange two columns of tiles. The goal of the game is to get all of the X tiles to be on the two main diagonals of the grid, forming a larger X shape, as in the following example for **N** = 5:

```
X...X
.X.X.
..X..
.X.X.
X...X
```

You are about to play with your X Squared toy, which is not yet in the goal state. You suspect that your devious younger sibling might have moved some of the tiles around in a way that has broken the game. Given the current configuration of the grid, can you determine whether it is possible to win or not?

Input

The first line of the input gives the number of test cases, **T**. **T** test cases follow. Each one begins with one line with an integer **N**, the size of the grid. **N** more lines with **N** characters each follow; the j-th character on the i-th of these lines is X if the tile in the i-th row and j-th column of the grid has an X, or `.` if that tile is blank.

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 `POSSIBLE` if it is possible to win, and `IMPOSSIBLE` otherwise.

Limits

1 ≤ **T** ≤ 100.
N mod 2 = 1. (**N** is odd.)
The grid has exactly 2 × **N** - 1 X tiles and exactly $\mathbf{N}^2$ - 2 × **N** + 1 `.` tiles.
The grid is not already in the goal state, as described in the problem statement.

Small dataset

3 ≤ **N** ≤ 5.

Large dataset

3 ≤ **N** ≤ 55.

Sample

Input	Output
2 3 ..X XX. XX. 3 ... XXX XX.	Case #1: POSSIBLE Case #2: IMPOSSIBLE

In Sample Case #1, one winning strategy is:

1. Swap the top row with the middle row.
2. Swap the rightmost column with the middle column.

```
..X    XX.    X.X
XX. -> ..X -> .X.
XX.    XX.    X.X
```

In Sample Case #2, no sequence of moves can turn the grid into the desired final configuration.

Problem C. Magical Thinking

Small input 6 points	Solve C-small
Large input 19 points	Solve C-large

Problem

You and **N** of your friends just took the B.A.T. (Binary Answer Test) to try to get into wizard school. The B.A.T. has **Q** true-false questions, and each one is worth 1 point. You have no wizard powers, so you just picked arbitrary answers and hoped for the best.

The results of the test have already been sent out by quail mail, but the quail with your results has not arrived yet. However, each of your friends has told you their list of answers and their total score. You also remember your own list of answers. You are an optimist and you think that you probably did well!

Given that there is one correct list of answers (but you do not know what those answers are), and given your friends' answers and scores, what is the highest score that you possibly could have achieved?

Input

The first line of the input gives the number of test cases, **T**. **T** test cases follow. Each begins with one line with two integers **N** and **Q**. Then, **N**+1 lines follow; the i-th of these lines represents the i-th examinee's list of answers $\mathbf{A_i}$, and has **Q** characters, each of which is either T or F (representing True or False). $\mathbf{A_{N+1}}$ is your own list of answers. Finally, one line with **N** integers follows; the i-th of these integers, $\mathbf{S_i}$, represents the i-th examinee's score. (Note that your own score is not in this list, because it is unknown.)

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 highest score that you possibly could have achieved that is consistent with the given information.

Limits

$1 \le \mathbf{T} \le 100$.
The length of $\mathbf{A_i}$ is **Q**, for all i.
Each character of $\mathbf{A_i}$ is either T or F, for all i.
$0 \le \mathbf{S_i} \le \mathbf{Q}$.
It is guaranteed that there is at least one possible list of correct answers that is consistent with all of the friends' answers and scores.

Small dataset

$\mathbf{N} = 1$.
$1 \le \mathbf{Q} \le 10$.

Large dataset

$1 \le \mathbf{N} \le 2$.
$1 \le \mathbf{Q} \le 50$.

Sample

```
Input      Output

3          Case #1: 2
1 2        Case #2: 1
TF         Case #3: 2
FF
1
1 3
TTT
TTF
0
2 3
TTF
FTF
TTT

1 2
```

Note that the last sample case would not appear in the Small dataset.

In sample case #1, your friend answered TF and you answered FF, and exactly one of your friend's answers was right. If your friend was wrong on question 1 and right on question 2, then the real set of answers is FF and you got both questions right. It is impossible to do better than this!

In sample case #2, your friend answered all Ts and got all of the questions wrong, so the real set of answers must be all Fs, which means that you got only question 3 right.

In sample case #3, the only possible real lists of answers that are consistent with the given information are FTT and FFF. (For example, the real answer list cannot be TFT; the first friend's answers and score would be consistent with that, but the second friend would have scored 0 instead of 2.) Of these two possibilities, FTT is more favorable to you and would give you a score of 2.

Kickstart Round C 2017

A. Ambiguous Cipher

B. X Squared

C. Magical Thinking

D. The 4M Corporation

Contest Analysis

Questions asked 2

Submissions

Ambiguous Cipher

7pt Not attempted **813/966 users** correct (84%)

12pt Not attempted **683/755 users** correct (90%)

X Squared

9pt Not attempted **377/706 users** correct (53%)

14pt Not attempted **319/358 users** correct (89%)

Magical Thinking

6pt Not attempted **570/621 users** correct (92%)

19pt Not attempted **149/325 users** correct (46%)

The 4M Corporation

11pt Not attempted **109/194 users** correct (56%)

22pt Not attempted **60/78 users** correct (77%)

Top Scores

ACMonster	100
subscriber	100
Kasugano.Sora	100
spnautilus	100
1717374	100
Benq	100
LeeSin	100
yubowenok	100
praran26	100
cephian	100

Problem D. The 4M Corporation

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 22 points — Solve D-large

Problem

The 4M Corporation has hired you to organize their departments and allocate headcount. You will create at least one department, and each department will receive some positive integer number of employees. It will not be easy, though — you have four different bosses, and each has given you a different instruction:

1. The department with the fewest employees must have exactly **MINIMUM** employees.
2. The department with the most employees must have exactly **MAXIMUM** employees.
3. The average number of employees across all departments must be exactly **MEAN**.
4. The median of the number of employees across all departments must be exactly **MEDIAN**. As a reminder, the median of a list is the value that, when the list is sorted in nondecreasing order, is in the center (for a list of odd length) or is the average of the two values in the center (for a list of even length).

Moreover, for the sake of efficiency, it is best to avoid creating too many departments. What is the smallest number of departments that you can create, if it is possible to satisfy your bosses' requests?

Input

The first line of the input gives the number of test cases, **T**. **T** test cases follow. Each consists of four integers: **MINIMUM**, **MAXIMUM**, **MEAN**, and **MEDIAN**, in that order.

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 either the minimum possible number of departments, or `IMPOSSIBLE` if it is impossible to satisfy all four bosses' requests.

Limits

1 ≤ **T** ≤ 100.

Small dataset

1 ≤ **MINIMUM** ≤ 8.
1 ≤ **MAXIMUM** ≤ 8.
1 ≤ **MEAN** ≤ 8.
1 ≤ **MEDIAN** ≤ 8.
The constraints for the Small dataset guarantee that the answer is either `IMPOSSIBLE` or is less than 14.

Large dataset

1 ≤ **MINIMUM** ≤ 10000.
1 ≤ **MAXIMUM** ≤ 10000.
1 ≤ **MEAN** ≤ 10000.
1 ≤ **MEDIAN** ≤ 10000.

Sample

Input	Output
5	Case #1: IMPOSSIBLE
6 4 5 1	Case #2: IMPOSSIBLE
7 7 8 8	Case #3: 1
2 2 2 2	Case #4: 2
3 7 5 5	Case #5: 3
1 4 3 4	

Sample Case #1 is `IMPOSSIBLE` because the maximum value cannot be smaller than the minimum value.

Sample Case #2 is `IMPOSSIBLE` because the mean and median cannot be larger than the maximum value.

In Sample Case #3, you can create a single department with 2 employees. This satisfies all four bosses: the department with the fewest employees has exactly 2, the department with the most employees has exactly 2, and the mean and median are both 2.

In Sample Case #4, you can create one department with 3 employees and another department with 7 employees. Note that it would **not** suffice to create only one department with 5 employees, because then the department with the fewest employees would not have exactly 3 and the department with the most employees would not have exactly 7.

For Sample Case #5, you can create one department with 1 employee and two more departments with 4 employees each.