Algorithms for Programming Contests
SS20 - Week 6

Chair for Foundations of Software Reliability and Theoretical Computer Science,
TU München

Mikhail Raskin, Tobias Meggendorfer, Stefan Jaax, A.R.Balasubramanian,
Christoph Welzel

Welcome to our practical course! This problem set is due by

Tuesday, 02.06.2020, 6:00 a.m.

This problem set differs from the usual one as there are seven problems, but only five of them are solvable in a few seconds of computation time. One of your tasks is therefore to figure out which problems to solve. The scoreboard will be randomized this week to avoid revealing the solvable problems. Submit your solutions at

`https://judge.in.tum.de/conpra/`

This week's problems are:

Six points will be awarded for each problem solved. Additionally, you will get three points for each problem you identified correctly as impossible. Make submissions for **at least** five problems to claim these points! If you submit for less problems you will not get any points for the impossible problems. You do not need to actually solve five problems, just submit something. These points are awarded for figuring out which problems are solvable and which problems cannot be handled practically. (The impossible problems will be edited to be worth three points after deadline)

If the judge does not accept your solution but you are sure you solved it correctly, use the “request clarification” option. In your request, include:

- the name of the problem (by selecting it in the subject field)
- a verbose description of your approach to solve the problem
- the time you submitted the solution we should judge

We will check your submission and award you half the points if there is only a minor flaw in your code.

If you have any questions please ask by using the judge’s clarification form.

Problem A
Alarm Clock

Lea is not a morning person. In fact, she hates getting up. Whenever she is in a particularly bad mood, she might even throw her alarm clock across the room and against the wall. She has been doing this for quite a while now and her clock is starting to malfunction.

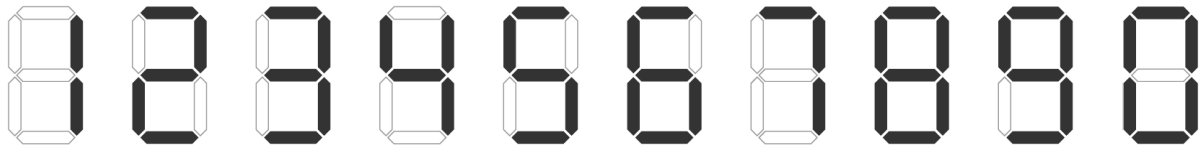

Figure A.1: The seven segments of Lea's clock

Lea's clock is a 24-hour digital clock with seven segments per number (see above). She suspects that some of the segments have been damaged and never light up while others still work fine, but Lea does now know which do work and which do not. Now she has a hard time figuring out how late it is. She has already watched the clock for some time, recording what it showed every minute, but she cannot figure out the time. Help her!

Input

The first line of the input contains an integer t. t test cases follow, each of them separated by a blank line.

Each test case starts with a line containing an integer n, n lines follow. Each of the following lines contains a string in the format "xx:xx" where each "x" is a digit from 0 to 9. The first line contains the time the clock showed initially, the second line the time one minute later and so on.

Output

For each test case, output one line containing "Case #i:" where i is its number, starting at 1. Output one more line formatted "xx:xx" for each possible time at which Lea could have started to watch the clock, sorted in ascending order. Under the assumption that some segments work and others never light up, the times must be consistent with all observations Lea made.

It might still be possible that the assumption is wrong and/or the clock itself is broken, and no time is consistent with all observations. In that case, output "none". It is theoretically possible that a broken clock shows a shape which does not correspond to any digit from 0 to 9, however Lea never observed such an event.

Each line of the output should end with a line break.

Constraints

- $1 \leq t \leq 20$
- $1 \leq n \leq 20$

Sample Input 1	Sample Output 1
7 1 23:49 1 88:88 10 11:24 11:25 11:26 11:27 11:28 11:29 11:30 11:31 11:32 11:33 4 12:73 12:74 12:75 12:75 5 16:11 16:12 16:13 16:14 16:15 7 22:77 22:78 22:79 22:10 22:11 22:12 22:13 3 18:70 18:71 18:72	Case #1: 23:48 23:49 Case #2: none Case #3: 00:24 01:24 03:24 04:24 07:24 08:24 09:24 10:24 11:24 13:24 14:24 17:24 18:24 19:24 Case #4: 02:03 02:33 08:03 08:33 12:03 12:33 18:03 18:33 Case #5: 06:01 06:11 06:31 06:41 08:01 08:11 08:31 08:41 16:01 16:11 16:31 16:41 18:01 18:11 18:31 18:41 Case #6: 22:07 22:37 Case #7: 08:00 08:30 18:00 18:30

Problem B
Muffin Queen

Lea is passionate about baking muffins. She has a group of friends who bake together with her and sometimes they even participate in baking competitions. The next contest is the famous "World Muffin Championship" scheduled for tomorrow and Lea's team is working on its list of recipes. All winners will be awarded the title "Muffin Queen" respectively "Muffin King" for the following year. For each type of muffins they know which team member bakes it best and only that person should bake it. Since all of them have a similar baking level, it happened that each team member appears exactly two times on this list. Each member of the group may bring at most one of his or her two types of muffins to the competition, so they have to decide for each baker which one of the two recipes to include.

There will also be judges at the competition. Surprisingly, the judges do not care that much about whether the muffins fit together, but they want to see some special muffins they really like. Each of the judges has a list with some muffins he or she wants to taste. Since the judges do not know which baker is going to create which type of muffins it may happen that several or no recipes assigned to one baker appear on their lists.

Lea's team wants to make all judges happy for obvious reasons. Therefore, their plan is to bring muffins in a way that each judge sees at least one of the recipes on his list. But is this even possible?

Input

The first line of the input contains an integer t. t test cases follow, each of them separated by a blank line.

Each test case begins with a line consisting of two integers m, the number of bakers, numbered from 1 to m, and n, the number of judges. n lines follow describing the judges. The i-th line contains the space-separated list of recipes judge i likes most. Each of these lines contains several integers, where a positive integer a means the first recipe of baker a and a negative integer $-b$ means the second recipe of baker b. The judge's lines end with 0.

Output

For each test case, output one line containing "Case #i: x" where i is its number, starting at 1, and x is "yes" if there is an assignment of recipes such that each judge can try at least on type of muffin from his list or "no" otherwise. Each line of the output should end with a line break.

Constraints

- $1 \leq t \leq 20$
- $1 \leq m \leq 20$
- $1 \leq n \leq 100$
- Each judge has at most m entries on his list.

Sample Input 1

```
2
2 2
1 -2 0
1 2 0

2 3
1 0
-1 2 0
-2 -1 0
```

Sample Output 1

```
Case #1: yes
Case #2: no
```

Sample Input 2

```
7
2 4
1 2 0
-2 0
-2 0
2 0

5 4
2 -3 0
-2 3 4 5 0
1 3 -4 0
-1 2 -3 -4 0

2 6
-1 0
-2 0
-2 0
-1 -2 0
1 2 0
-1 -2 0

7 1
-2 3 5 -7 0

3 4
-1 -3 0
1 -3 0
-2 3 0
2 3 0

7 3
1 -2 3 -4 5 0
1 2 6 0
-2 5 0

5 1
1 -3 4 -5 0
```

Sample Output 2

```
Case #1: no
Case #2: yes
Case #3: no
Case #4: yes
Case #5: no
Case #6: yes
Case #7: yes
```

Problem C
Planetarium Problem

As every year, the local planetarium runs a series of practical courses and internships which are free for the public. However, as the number of interested people was at a record high last year and is expected to rise yet again this time, the planetarium has decided to have trials in order to find out which people are actually taking part. Lea has always had a great interest in astronomy and astrophysics, so she likes to take part in some courses to learn more about the great mysteries of the universe. The task she is assigned is fairly simple: she is given a number of astronomical objects and has to decide in how many clusters this vast amount of objects can be divided. The restriction for the division is that all clusters must be of equal size and that no cluster can be subdivided any further into smaller clusters of equal size, except for trivial clusters of size 1. Lea has thought about the problem a lot, but cannot quite get to a point where she can decide how to do it efficiently. As usual, she asks you for help. What is your answer?

Input

The first line of the input contains an integer t. t test cases follow.

Each test case consists of only one integer n, the number of astronomical objects considered.

Output

For each test case, output one line containing "Case #i:" where i is its number, starting at 1, and a sequence of all possible cluster sizes, in increasing order, separated by spaces.

Constraints

- $1 \leq t \leq 20$
- $1 \leq n \leq 10^{85}$

Sample Input 1

```
2
16445
357
```

Sample Output 1

```
Case #1: 1 5 11 13 23
Case #2: 1 3 7 17
```

Sample Input 2

```
7
563233
41578
15789543
44155578
1574682
852636
19
```

Sample Output 2

```
Case #1: 1 11 51203
Case #2: 1 2 20789
Case #3: 1 3 7 11 29 2357
Case #4: 1 2 3 37 198899
Case #5: 1 2 3 19 727
Case #6: 1 2 3 41 1733
Case #7: 1 19
```

This page is intentionally left blank.

Problem D
Queens Problem

Queens are quite picky persons. Lea currently has to deal with an especially nasty one. Her task is to create a painting for the Queen of Templonia, but of course the queen has certain preferences: She wants the painting to be a quadratic grid of $n \times n$ cells, where n is a natural number. Furthermore, n of the squares in the grid should be filled, the rest should be left empty. Any two filled squares should not be in the same row, in the same column, or on the same diagonal through the grid. (E.g.: (1,2) and (3,4) are on the same diagonal, (3,4) and (4,3) are on the same diagonal, but (3,3) is not on the same diagonal as any of the others.)

Lea has prepared multiple paintings. For each she has already decided on a number n, drawn the grid and in some cases even filled out some squares. Can you help her decide which of her paintings can be finished according to the constraints above?

Input

The first line of the input contains an integer t. t test cases follow, each of them separated by a blank line.

Each test case starts with an integer n indicating that the grid has dimension $n \times n$. n lines follow, each with exactly n characters. The i-th line describes row i in the grid, the j-th character describes square j. A "." denotes an empty square, an "x" denotes a filled square.

Output

For each test case, output one line containing "Case #i:" where i is its number, starting at 1. If there is a possibility to fill the grid according to the constraints above, output n more lines containing the solution in the same format as the input. If there are multiple solutions, any will be accepted. If it is not possible to complete the grid, output the line "impossible". Each line of the output should end with a line break.

Constraints

- $1 \leq t \leq 20$
- $1 \leq n \leq 15$

Sample Input 1

```
8
4
..x.
....
....
....

4
..x.
....
x...
....

4
..x.
x...
...x
.x..

2
..
..

1
x

7
...x...
.......
.......
.x.....
.......
.......
......x

5
.x...
...x.
x...x
x....
.....

6
......
......
.....x
...x..
......
......
```

Sample Output 1

```
Case #1:
..x.
x...
...x
.x..
Case #2:
impossible
Case #3:
..x.
x...
...x
.x..
Case #4:
impossible
Case #5:
x
Case #6:
...x...
x......
....x..
.x.....
.....x.
..x....
......x
Case #7:
impossible
Case #8:
impossible
```

Problem E
Story Time

On another boring afternoon a few months ago, Lea decided to write a fantasy book which she would call "Game of Kings", an epic tale about kings and queens, bishops and knights. Oh, and magic.

She has spent many nights imagining what would happen to all the awesome characters she would describe. Now, she just needs to compile the list of events into one book. For every character, she wrote a list of chapters which are centered around him or her. Every chapter is centered around exactly one character, as she felt it would be confusing otherwise. These have to happen in a fixed order.

Also, some chapters have a specific ordering. For example, the scene in which Queen Lena, nicknamed the "Black Queen", has a nervous breakdown over a letter detailing the abduction of her brother, a famous knight, has to happen after the description of the abduction from her brothers point of view. (The story would go on to detail how her knight would be brainwashed into a fearsome "White Knight" that would be difficult to keep in check.) Lastly, two sequential chapters should not be centered around the same character - that would just be boring.

Can you tell her in how many possible ways she can arrange all the scenes she imagined?

Input

The first line of the input contains an integer t. t test cases follow, each of them separated by a blank line.

Each test case consists of two integers n and m, where n is the number of characters (indexed from 1 to n) and m is the number of dependencies between chapters. n lines follow. The i-th line consists of a single integer a_i, the amount of chapters that are centered around character i. m lines follow. The j-th line consists of four integers c_j, p_j, d_j, and q_j and means that the p_j-th chapter centered around character c_j has to happen before the q_j-th chapter centered around d_j.

Output

For each test case, output one line containing "Case #i: x" where i is its number, starting at 1, and x is the amount of possible orderings of the chapters such that:

- Sequential chapters are about different characters.
- Chapters about a single character are in a fixed order.
- All m inter-chapter-dependencies are fulfilled.

Each line of the output should end with a line break.

Constraints

- $1 \leq t \leq 20$
- $1 \leq n \leq 6$
- $0 \leq m \leq 15$
- $1 \leq p_i, q_i \leq a_i \leq 13$
- $c_j \neq d_j$
- $1 \leq c_i, d_i \leq n$
- There are at most 13 chapters.

Sample Input 1

```
2
3 0
2
2
2

4 3
3
3
2
3
1 1 2 3
1 3 2 2
3 1 2 1
```

Sample Output 1

```
Case #1: 30
Case #2: 570
```

Sample Input 2

```
5
2 2
3
3
2 1 1 2
1 3 2 3

3 1
1
2
2
2 1 3 2

3 1
2
1
1
3 1 1 2

2 2
3
2
1 1 2 2
2 2 1 3

3 2
2
2
2
1 2 3 2
1 1 3 1
```

Sample Output 2

```
Case #1: 1
Case #2: 12
Case #3: 5
Case #4: 1
Case #5: 11
```

Problem F
Hockey Champion

One of Lea's great interests are sports. She likes to get active and go for a run, a bike ride, or play some team sports with other sport enthusiasts. What fascinates Lea as well is the whole science behind every sport, there is so much more than just playing a game, or riding a bike. Lately, Lea has taken a particular interest in hockey. And there are virtually uncountable statistics about everything you can and cannot imagine, such as goals scored, distance covered, longest winning streaks, average left foot sizes of players with jersey number 4, highest victory, and so on. These are all gathered and stored (and fortunately for Lea, publicly available) in a huge database called **S**tatistics for **H**ockey **I**nstitutions and **T**eams. But still, Lea faces some difficulties. The hockey league has recently changed their format so that the teams that play against each other in the next match are picked randomly among all teams. Lea, angry about the decision of the league managers to make the sport "more fun due to chaos", is now not able to tell which team has been the best one throughout the last season. She decides to search for the largest subset of teams that have all played against each other, so that she at least knows where to find her personal best team. Knowing that you are as excited about statistics as she is, Lea sends you the data to help her in the matter. Can you help her find the maximal subset of teams that have all played against each other?

Input

The first line of the input contains an integer t. t test cases follow, each of them separated by a blank line.

Each test case begins with two integers n and m. n is the number of teams in total (indexed from 1 to n), and m is the number of matches played throughout the last season. m lines follow. The i-th line consists of two integers a_i and b_i, indicating that team a_i has played a match against team b_i.

Output

For each test case, output one line containing "Case #i: x" where i is its number, starting at 1, and x is a space-separated list of all teams that are part of a largest subset of teams that have all played against each other. If there are multiple such subsets, any of them will be accepted.

Constraints

- $1 \leq t \leq 20$
- $1 \leq n \leq 1000$
- $0 \leq m \leq 100000$
- $1 \leq a_i, b_i \leq n$ for all $1 \leq i \leq m$
- $a_i \neq b_i$ for all $1 \leq i \leq m$

Sample Input 1

```
2
5 5
1 2
1 5
2 3
2 4
3 4

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

Sample Output 1

```
Case #1: 2 3 4
Case #2: 1 6 7 8
```

Sample Input 2

```
2
8 12
8 4
4 6
1 2
2 8
7 4
4 2
5 1
7 6
3 6
8 1
6 5
7 8

8 15
3 4
1 7
5 4
3 6
2 8
1 8
7 3
2 7
4 8
6 3
6 4
2 6
4 5
2 7
5 3
```

Sample Output 2

```
Case #1: 1 2 8
Case #2: 3 4 5
```

Problem G
Trapped

Over the last few weeks, Lea was hiking almost every day. She loves walking around in the nature and enjoys the feeling of freedom and the silence on top of the mountains. She knows most of the mountains close to her home quite good, but recently, she found a way she did not know before. Following the unknown trail, Lea suddenly stood in front of a big cave. Following her curiosity, Lea walked into the cave when she suddenly heard a loud noise. Stones were falling down and crashing everything in their way.

Luckily Lea was safe inside the cave, but the entry was blocked by a huge rock. Having no other possibilities, Lea started to explore the cave and found some tools lying around. If she can find all the tools in the cave, she would be able to move the rock far enough to slip out of her prison. Since she needs to remember where she came from, Lea uses the following technique: She divides the cave into squares of 1x1 meter each and walks only from each field to one of its four neighbouring fields. To remember the position of the cave's entry, she uses a piece of chalk and draws an arrow on each field pointing to the direction she came from. She does not want to walk a field twice since two arrows on one field would mean that Lea might take the wrong direction when going back. Also, she is not able to erase the arrows she painted.

Is there such a way through the cave, that starts at Lea's position and visits all tools, but does not use a field twice? After collecting the tools she will follow her arrows back to the entry of the cave, open it and enjoy her regained freedom.

Input

The first line of the input contains an integer t. t test cases follow, each of them separated by a blank line.

Each test case (better to say test cave in this case) starts with a line containing two integers w and h, the cave's width and depth, respectively, in meters.

h lines follow describing the cave, each of them containing w characters. The i-th character in the j-th line describes the field (i, j) where the characters could mean the following:

- "#": a non-walkable field or wall. All other fields are walkable.
- "_": a normal walkable field
- "L": Lea's starting position
- "T": a position of a tool

Output

For each test case, output one line containing "Case #i: x" where i is its number, starting at 1, and x is "yes" if there is such a way through the cave or "no" otherwise. Each line of the output should end with a line break.

Constraints

- $1 \leq t \leq 20$
- $1 \leq w, h \leq 10$
- There will be exactly one "L" and at least one "T" per test case.
- There will be at most 25 walkable fields per test case.

Sample Input 1

```
2
4 3
T___
____
T__L

5 3
T___T
_____
L____
```

Sample Output 1

```
Case #1: yes
Case #2: yes
```

Sample Input 2

```
5
7 7
###___#
_##_#T#
_##___#
_###_##
__L#__#
_####_#
_______#

8 6
__T#####
TT_##__#
LT###T_#
_______##
_####_##
_####TTT

9 6
#######_#
##_____#_T
#####_____
__L__#_#_
####_#_##
####_____#

9 8
T_____T_T#
_#L###TT#
T########
T########
_########
_########
T__##T###
##_____T##

6 6
_L_TT#
##_T_#
_#T__#
_#_T_#
_#__T#
T___T#
```

Sample Output 2

```
Case #1: yes
Case #2: no
Case #3: yes
Case #4: no
Case #5: yes
```