Problem A
Sudoku Checker

A Sudoku puzzle of size S consists of S^2 non-overlapping "S x S" squares organized into a grid of "S^2 x S^2" numbers. Columns are labeled from left to right starting with number "1", and rows are labeled from top to bottom starting with number "1". Squares are labeled from top to bottom, starting with number "1", such that squares spanning the same rows are assigned consecutive labels. For a puzzle of size 3, the S^2 non-overlapping "S x S" squares are labeled as follows:

1	2	3
4	5	6
7	8	9

A puzzle is given as a partially filled "S^2 x S^2" grid, and a solver attempts to fill each grid location with numbers from 1 to S^2 such that each number appears in:

- Each of the S^2 vertical columns,
- Each of the S^2 horizontal rows, and
- Each of the S^2 non-overlapping "S x S" squares.

An example of a valid solution for a puzzle of size 4 is:

```
14 12  8 10 16  7  4  3  2  9 15  6 11  1  5 13
 7 11 16  9  6  1 12  5 13 10  3  8 15  2 14  4
 5 15  4  3 14 13 10  2  1 12 11 16  6  9  7  8
 6 13  2  1  8 11 15  9  4  5  7 14 16  3 12 10
 2  1  5 13  3 15  9  4  8 16  6  7 14 11 10 12
11  4 10  6 13 12  5 16  3  2 14  9  7  8 15  1
 3 16 12 14 10  8  7 11  5 15  1  4 13  6  2  9
15  9  7  8  2 14  1  6 12 13 10 11  4  5  3 16
 1  5 13  7 12  4  2 10 11  6 16  3  8 15  9 14
10  3 15 16  7  6  8 14  9  4 13  5  1 12 11  2
 8  6 11  2  9 16 13  1 10 14 12 15  3  7  4  5
 4 14  9 12 11  5  3 15  7  8  2  1 10 13 16  6
12  7  6  4 15  2 14  8 16 11  9 13  5 10  1  3
16 10 14  5  1  9 11 13 15  3  8 12  2  4  6  7
 9  8  3 15  5 10  6  7 14  1  4  2 12 16 13 11
13  2  1 11  4  3 16 12  6  7  5 10  9 14  8 15
```

Your task is to write a program to check the validity of solutions for Sudoku puzzles.

Input

Input starts with an integer N, $1 \leq N \leq 50$, on a separate line that represents the number of puzzles to be checked. The specifications of each puzzle starts with an integer on a separate line S ($2 \leq S \leq 12$) that represents the size of the puzzle. Each of the next S^2 lines lists one line in the Sudoku puzzle's solution, where each line consists of S^2 integers separated by single spaces.

Output

For each puzzle, print the puzzle number (starting with 1, and using the format in the sample) followed by either:

- a single line with the statement "The solution is valid.", or
- a number of lines that list all the errors followed by a separate line with the statement "The solution is invalid." First, errors in squares are listed in increasing order of their labels, followed by errors in rows and finally errors in columns.

For each line all the missing numbers are listed first, in increasing order, followed by the numbers that appear multiple times, again in increasing order, using the format shown in the sample below.

Sample Input	Output for the Sample Input
3 2 1 2 3 4 3 4 1 2 2 1 4 3 4 3 2 1 2 1 2 3 1 3 4 1 2 2 1 4 3 2 3 2 1 3 3 1 6 7 4 9 8 5 2 7 5 9 2 8 6 1 3 4 4 8 1 5 1 3 6 7 9 1 2 3 3 7 4 5 6 8 8 9 7 6 4 6 2 6 3 6 4 5 4 2 8 9 1 7 5 7 4 1 9 2 3 8 6 2 3 1 8 6 7 6 9 5 9 6 8 4 3 5 7 2 1	Puzzle 1 The solution is valid. Puzzle 2 Square 2 has no 4s and 2 1s Square 3 has no 4s and 2 2s Row 1 has no 4s and 2 1s Row 4 has no 4s and 2 2s Column 1 has no 4s and 2 2s Column 4 has no 4s and 2 1s The solution is invalid. Puzzle 3 Square 1 has no 2s and 2 1s Square 5 has no 1s and no 5s and no 9s and 3 4s and 2 6s Square 6 has no 4s and 2 6s Square 9 has no 4s and 2 6s Row 3 has no 2s and 2 1s Row 4 has no 9s and 2 3s Row 5 has no 1s and no 5s and 3 6s Row 6 has no 3s and 2 4s Row 8 has no 4s and 2 6s Column 3 has no 2s and 2 1s Column 4 has no 9s and 2 4s Column 5 has no 5s and 2 4s Column 6 has no 1s and 2 6s Column 7 has no 4s and 2 6s Column 8 has no 4s and 2 6s The solution is invalid.

Problem B
Time to Balance

When the finances of companies fall into disarray, the big honcho in the land of B likes to see deficits reduced as soon as possible to a manageable level. The preferred approach requires such companies to:

1. reduce the current yearly deficit in half, if the reduced deficit remains an integer and is not less than its maximum manageable level, otherwise
2. reduce the current yearly deficit by 1, if the reduced deficit is not less than its maximum manageable level.

Each company in the land of B must declare its maximum manageable deficit level, for tax purposes, as a positive integer .

Your task is to write a program to read the companies' data and to calculate the minimum number of years each company needs to get into maximum manageable deficit levels. If the current deficit of a company is below or equal to its maximum manageable level, your program should print zero.

Input

Input consists of the maximum manageable and current deficits for several companies. The data for each company consists of two positive integers, on a line by themselves: the first integer represents the maximum manageable deficit and the second integer represents the current deficit. The two integers are separated by a single space, and are less than 10000 . A line with two zeros, separated by a single space, denotes the end of the input.

Output

For each company, print the number years it needs to return to manageable deficit level as an integer on a separate line, in the order of their data's presence in the input.

Sample Input	Output for the Sample Input
4 7 1 9 4 8 4 10 10 4 4 16 0 0	3 4 1 2 0 2

Problem C
Golfing with the Stars

Writers, actors and actresses, sport stars, and some famous geeks have donated their time to support the ACM *"Golfing with the Stars"* global charity event. Each star promised to play as many rounds of golf as needed so that each donor to the charity can play one, and only one, round of golf with one star from her/his nominated set of favorite stars. This problem is about a volunteer role for you to support this event.

Your task is to write a program that reads a listing of the names of the stars donating their time to an event, a listing of the donors along with their favorite stars, and then assigns stars to one or more donors, for a round of golf, in a way that minimizes the maximum number of golf rounds any star has to play.

Input

Input begins with an integer G on a line by itself, $1 \leq G \leq 20$, that represents the number of scenarios where each scenario describes a charity event. Each event description begins with two integers N and M, separated by a single space, on a line by themselves. The integer N represents the number of stars and M represents the number of donors. Each of the next N lines contains the name of a star. Each of the following M lines, one line per donor, starts with the name of a donor followed by the names of her/his favourite stars, if any. Names are separated by a single space, and the name of a star cannot appear more than once in each line.

Each name (star or donor) consists of a single word; that is, a string with no white spaces. The names of stars and donors are all distinct from each other. $0 < N \leq 50$ and $0 < M \leq 2000$.

Output

For each charity event, print the event number (starting with 1, and using the format in the sample) followed by a single space and then an integer describing the maximum number of golf rounds any star has to play.

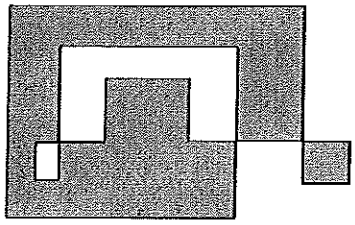
Sample Input	Output for the Sample Input
2 3 7 PrincessLeia DarthVader R2D2 grumpy DarthVader sneezy PrincessLeia R2D2 dopey DarthVader PrincessLeia R2D2 doc DarthVader R2D2 sleepy DarthVader bashful happy PrincessLeia 3 6 CaptainKirk MrSpock Xindi grumpy MrSpock sneezy MrSpock dopey MrSpock doc MrSpock sleepy MrSpock happy CaptainKirk	Event 1: 2 Event 2: 5

Problem D
Farmland Redistribution

The world of D is a large rectangular piece of land near the mouth of a long river that is regularly flooded. While a flood is good for the land, it destroys all markings that define land holdings. After a flood the head honcho redistributes the land amongst the farmers according to the ancient rules of D:

1. Following each flood, all the land of D must be distributed as rectangular lots of land such that each farmer receives one rectangular lot equal in area to his land holdings before the flooding.
2. A farmer may sell all his land. He may also sell a rectangular part as long as it is not completely surrounded by the remaining portion and that his land holdings remain connected.
3. A farmer may buy a rectangular lot of land as long as all his land holdings remain connected.

The buying and selling of land can result in land holdings with rather complex shapes. For example:



It is important to note that the border line of a land holding is not part of its interior, and that a land holding is considered connected if it is possible to walk around its whole border without stepping into the interior of any land holding, including her/his own.

This problem is about a small role for you in the development of a game about life in the world of D. Your task is to write a program that is capable of reading a map of land holding of a single farmer and calculate its area.

Input

Input consists of several scenarios, where each scenario describes a map. Each map description begins with an integer N, on a line by itself, that represents the number of sides of the border of a land holding. The next N lines list the sides in order of their appearance on the border as one walks with the land on his right-hand side.

Each of the N lines, which gives a description of a single side, contains four integers separated by a single space. The first pair of integers represents the x- and y-coordinates of one end-point of the side, and the second pair represents the x- and y-coordinates of the other. All x- and y-coordinates are given as positive integers.

$0 \leq N \leq 10000$, and a value of N less than four (4) denotes the end of the input.

Output

For each map, print the map number (starting with 1, and using the format in the sample) followed by a single space and then an integer describing the value of the land area.

Sample Input	Output for the Sample Input
4 1 1 1 2 1 2 3 2 3 2 3 1 3 1 1 1 8 1 1 1 2 1 2 3 2 3 2 3 5 3 5 5 5 5 5 5 2 5 2 3 2 3 2 3 1 3 1 1 1 6 1 1 1 2 1 2 3 2 3 2 3 5 3 5 5 5 5 5 5 1 5 1 1 1 2	Map 1: 2 Map 2: 8 Map 3: 10

Problem E
Cheap Tours

A tour operator thought he had finally struck gold when he signed a long term binding contract with a local bus company to drive his customers between sites of his tours. According to the terms of the contract, the bus company will charge a flat fee per mile. Unfortunately, the Global Financial Crisis was not kind to him as the large number of expected customers turned to be a mirage, and his competitors slashed the prices of similar tours.

After some research he realized that he can retain his customers by reducing the cost of each tour. Such cost reduction can be achieved by reorganizing the order in which sites are visited to minimize the total distance traveled in each tour. The bad news is that reorganizing a tour to visit the same sites, possibly in a different order, to achieve the smallest possible distance is very hard task that may take months to perform. As a compromise, he settled on looking for reorganizations of his tours with guarantees that they are not much worse than the shortest possible.

Your task is to write a program that processes the given geographic locations, described by x- and y-coordinates, of sites to be visited in a tour, and plan a good tour to visit all of them. The tour's length must not be longer than twice the length of the shortest possible tour.

Input

Input consists of several scenarios, where each scenario describes a tour. Each tour description begins with a single integer N, on a line by itself, which represents the number of sites to be visited in the tour. Each of the next N lines contains two numbers, separated by a single space, that represent the x- and y-coordinates of the geographic location of a site. Each number is larger than or equal to zero and may be given in floating point or in integer format. $0 \leq N \leq 10000$, and a value of N equal to zero, on a line by itself, indicates the end of the input and should not be processed.

Output

The output for each tour consists of a single line that contains a number denoting the length of the proposed tour rounded to four decimal places in floating point format, on a line by itself, as shown in the sample output. It is important to note that there are multiple reorganizations for a given tour, and that any reorganization whose length is not longer than twice the length of the shortest possible tour is valid.

Sample Input	Output for the Sample Input
5 0 0 0 6 0 10 6 8 11 10 8 0 0 0 3 4 0 8 0 12 0 4 2.99 8 2.99 12 3 0	36.5758 37.3493

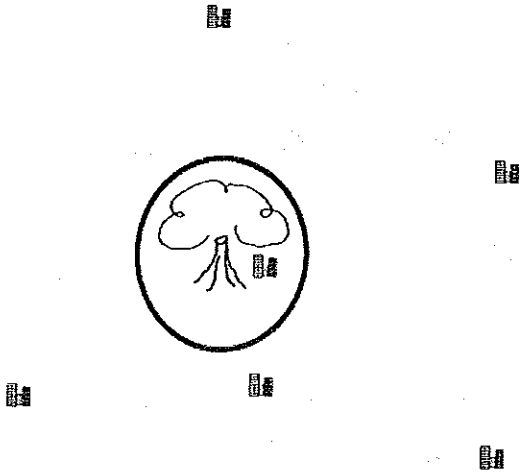
Note: 36 and a touch over 30 are the shortest lengths of the first tour and the second tour shown in the table, respectively.

Reminder: Rounding a positive number R.xxxxy to four decimal places

1. If the fifth decimal place is less than 5,
   then the rounded value equals R.xxxx
2. Otherwise, the rounded value equals R.xxxx + 0.0001

Problem F
Mt Convenient

A volcano has erupted, casting a circular cloud of ash over a northwest district of Flatland, where the world is absolutely flat. There are a number of cities in the area, and each pair of them is connected by a road, which is always a straight line. In the interests of safety the district's council closed all the roads that intersect or touch the ash cloud.



The ensuing travel disruption left a lot of travelers stranded in bus terminals around the volcano. Javier Alfonso, a young and enthusiastic journalist, plans to travel in a closed loop that contains the volcano's exclusion zone by bus, in an anti-clockwise direction, from a bus terminal in one city to another. He wants to gather information about the situation in the whole area as quickly as possible to complete his report.

Your task is to write a program to calculate the minimum distance for a trip around the volcano. The trip does not have to visit all the cities, but it must satisfy the following two conditions:

1. All travels must be on roads between the cities, and
2. No segment of Javier Alfonso's travel can be in a clockwise direction with respect to the volcano's center.

Input

Input consists of G scenarios, $1 \leq G \leq 30$, where each scenario describes a volcanic event. Each event description begins with an integer N, $0 \leq N < 200$, on a line by itself that represent the number of cities. A value of N equal to zero, on a line by itself, indicates the end of the input and should not be processed. The next line contains three integer values X, Y and R that represent the location of the volcano and its radius. Each of the next N lines contains two integer values Xc and Yc that represent the location of a city. $1 \leq X,Y,R, Xc, Yc < 1000000$.

Output

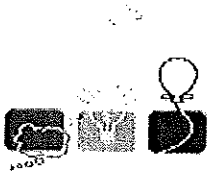
For each event, print the event number (starting with 1, using the format in the sample) followed by a single space and then a value that represents the minimum possible total length for a trip around the volcano rounded to four decimal places. You may assume that no ambiguity due to rounding errors is present in all test cases.

If a trip is not possible, then print "fail" (without double quotation) instead.

Sample Input	Output for the Sample Input
6 48 53 15 54 55 55 74 12 75 97 40 46 16 95 85 4 3 3 1 1 1 1 5 5 1 5 5 1 1 1 1 1 1 0	1: 221.5091 2: 16.0000 3: fail

Reminder: Rounding a positive number R.xxxxy to four decimal places

1. If the fifth decimal place is less than 5,
   then the rounded value equals R.xxxx
2. Otherwise, the rounded value equals R.xxxx + 0.0001



Problem G
Drink Orders

Each year a group of acquaintances gets together to discuss programming competitions. After the discussions it is time for drinks. The group come from different cities and drinks of a particular volume in one city are often known by other names in other cities. For example a Pot in Brisbane is the same as a Middie in Woolloomooloo and as a Schooner in Adelaide.

Your task is to write a program to help translate the drinks orders for the host.

Input

The input consists of a series of scenarios. Each scenario starts with a positive integer N, on a line by itself, which is the number of cities that people come from ($2 \leq N \leq 100$). Each of the following N lines contains the name of a city.

The next line contains a positive integer D, which is the number of sizes of drinks that can be ordered ($2 \leq D \leq 100$). Each of the following D lines contains the size of a drink and the names by which it is known in each of the N cities listed in the same order as the order of cities in the input.

The next line contains a positive integer P, which is the number of people ordering drinks ($2 \leq P \leq 100$). Each of the following P lines contains the name of a person followed by her/his city name. Each person's name is unique.

The next line contains the name of the city where the meeting is held.

The next line contains a positive integer R, which is the number of rounds ($1 \leq R \leq 5$) where each round consists of P orders for a drink, one from each person at the meeting, as they are known in their own city. There will be $P \times R$ pairs of names and drinks before input for the next scenario begins.

Each name of a city, a person or a drink consists of a single word, and input is terminated by a scenario with a value of N equals 0.

Output

The output for each scenario begins with the scenario number and the name of the city in which the meeting is being held (starting with 1, and using the format in the sample), followed by a line that contains the round number (starting with 1, and using the format in the sample). The following lines contain the drinks to be ordered for that round before output for the next round, or scenario, begins.

Each line starts with a positive integer representing the number of drinks of a certain size followed by the name of the drink. The drink names in the output are those used in the city where the meeting is held, and are to be listed in increasing order of their sizes. If the number of drinks is greater than 1, the name of the drink should be made plural by appending the character 's'.

Sample Input	Output for the Sample Input
4 Adelaide Brisbane Wangaratta Sydney 3 140 Pony Pony Five Five 200 Butcher Beer Seven Seven 285 Schooner Pot Middie Middie 4 Alan Adelaide Bruce Brisbane Steve Wangaratta Max Brisbane Sydney 2 Alan Butcher Bruce Pony Steve Middie Max Pony Steve Seven Alan Butcher Bruce Beer Max Beer 0	Scenario 1 – Sydney Round 1 2 Fives 1 Seven 1 Middie Round 2 4 Sevens

Problem H
to Sir, with Love

The 2010 graduating students of a college want to present their teacher with a special present in appreciation for his long hours of work with the poorly performing students of the class. After dispensing with the traditional, they settled on the idea of a specially designed computer program. The program is meant to reduce his workload by making assignments of buddies, from the pool of good students, to assist poorly performing students with their difficulties. A tuple of three integers (A_g, A_n, A_l) denotes a student's mark in the three categories of general abilities, numeracy, and literacy. The program is meant to use students' marks and applies the following rules for the assignment of buddies:

1. A student X is assigned as a buddy to a student Y if:
   (a) X has better or equal marks than Y in all categories, but not equal in all three categories, and
   (b) The function f(X,Y), defined as $|X_g - Y_g| + |X_n - Y_n| + |X_l - Y_l|$, has the largest value among all possible values f(s,Y), where s is a member of the set of students, and
   (c) Where f(X,Y) and f(Z,Y) have equal values, X gets the assignment instead of Z if
      i. $|X_g - Y_g| > |Z_g - Y_g|$, or
      ii. $|X_g - Y_g| = |Z_g - Y_g|$ and $|X_n - Y_n| > |Z_n - Y_n|$.

2. Where more than one good student has the same marks in all three categories, potential buddies are divided between them such that the number of their assignments differs by no more than one. The 2nd group in the "sample input" demonstrates such situation.

Your task is to write a program to read the students' marks in the three categories and calculate the maximum number of poorly performing students to be assigned to a single buddy.

Input

Input consists of the students' marks in the three categories for several student groups. Input starts with an integer M, $0 < M \leq 20$, on a separate line that represents the number of groups to be processed. The data for each group starts with a single integer N, on a line by itself, that represents the number of students in the group. $0 < N \leq 100000$. Each of the following N lines contains three integers, separated by single spaces, that represent the three marks for general abilities, numeracy, and literacy.

Output

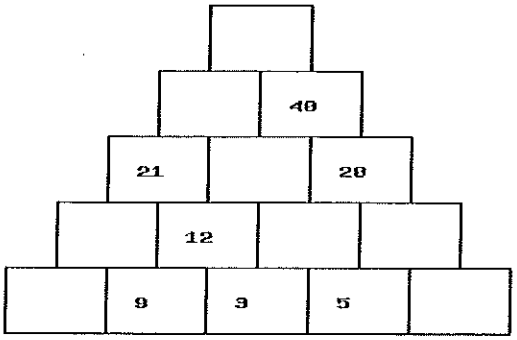
For each group of students, print group number (starting with 1, and using the format in the sample) followed, after a single space, by the maximum number of poorly performing students to be assigned to a single buddy.

Sample Input	Output for the Sample Input
3 4 10 5 0 5 10 0 5 5 0 7 7 0 7 10 10 10 10 10 10 1 1 1 2 2 2 3 3 2 4 4 4 5 5 6 8 11 9 10 10 10 10 9 11 10 1 1 1 2 2 2 3 3 3 4 4 4 5 5 0	Group 1: 1 Group 2: 3 Group 3: 5

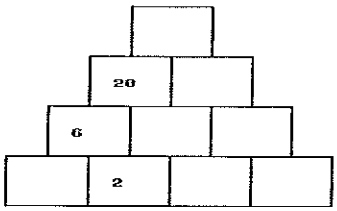
Problem I
*e*Door

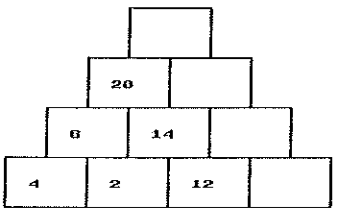
Bob the builder is a small business man with grand ideas, and the Building Education Revolution (BER) was an opportunity not to be missed. For each school hall that he has a contract to build, Bob plans to install a door that displays an educational puzzle and will only open if a correct solution is entered. The puzzle consists of a step pyramid with N rows and has a base of N boxes, as illustrated below with N equal to five (5). Each box is either empty or has a single non-negative integer whose value is:

1. larger than integers that may be present in the two boxes directly beneath it, and
2. equal to the sum of the integers, if both present, in the two boxes directly beneath it.



A puzzle is solved by making the maximum possible assignment of unique non-negative integers, which satisfy the above mentioned rules, to empty boxes. If more than one value is possible for an empty box, then it must be left empty, as shown in the following example:





Your task is to write a program to solve such puzzles and thus verify the entered answers.

Input

Input consists of several wall puzzles. Each puzzle description begins with an integer N, on a line by itself, that represents the number of rows in the puzzle. Each of the following N lines lists the entries in one of the rows starting with the top row. The top line has one entry, and each of the following N-1 lines has one more entry than the one preceding it. The entries on each line are separated by single spaces, and each entry is either the character "*", which represents an empty box, or an integer with a value in the range of 0 to 2000000000, inclusive. $0 \leq N \leq 50$, and a value of N equal to zero denotes the end of the input.

Output

For each puzzle, print the puzzle number (starting with 1, and using the format in the sample) followed by n lines showing the puzzle with integer values in all the boxes that can be resolved. Indicate boxes that cannot be uniquely resolved with an asterisk (*).

Sample Input	Output for the Sample Input
6 * * 100 34 * * * * * 33 * * 10 * * * 3 * * * 9 6 * 111 * 55 * * * 28 * 24 * 15 13 * 9 * * 5 * * * 0	Puzzle 1 * * 100 34 * * * * * 33 * * 10 * * * 3 * * * 9 Puzzle 2 219 111 108 55 56 52 27 28 28 24 12 15 13 15 9 2 10 5 8 7 2