Problem A: Astronauts

Introduction

The Bandulu Space Agency (BSA) has plans for the following three space missions:

- Mission A: Landing on Ganymede, the largest moon of Jupiter.
- Mission B: Landing on Callisto, the second largest moon of Jupiter.
- Mission C: Landing on Titan, the largest moon of Saturn.

Your task is to assign a crew for each mission. BSA has trained a number of excellent astronauts; every one of them can be assigned to any mission. However, if two astronauts hate each other, then it is not wise to put them on the same mission. Furthermore, Mission A is clearly more prestigious than Mission B; who would like to go to the second largest moon if there is also a mission to the largest one? Therefore, the assignments have to be done in such a way that only young, inexperienced astronauts go to Mission B, and only senior astronauts are assigned to Mission A. An astronaut is considered *young* if their age is less than the average age of the astronauts and an astronaut is *senior* if their age is at least the average age. Every astronaut can be assigned to Mission C, regardless of their age (but you must not assign two astronauts to the same mission if they hate each other).

Input

The input contains several blocks of test cases. Each case begins with a line containing two integers $1 \leq n \leq 100000$ and $1 \leq m \leq 100000$. The number n is the number of astronauts. The next n lines specify the age of the n astronauts; each line contains a single integer number between 0 and 200. The next m lines contains two integers each, separated by a space. A line containing i and j ($1 \leq i, j \leq n$) means that the i-th astronaut and the j-th astronaut hate each other.

The input is terminated by a block with $n = m = 0$.

Output

For each test case, you have to output n lines, each containing a single letter. This letter is either `'A'`, `'B'`, or `'C'`. The i-th line describes which mission the i-th astronaut is assigned to. Astronauts that hate each other should not be assigned to the same mission, only young astronauts should be assigned to Mission B and only senior astronauts should be assigned to Mission A. If there is no such assignment, then output the single line `'No solution.'` (without quotes).

Sample Input

```
16 20
21
22
23
24
25
26
27
28
101
102
103
104
105
106
107
108
1 2
3 4
5 6
7 8
9 10
11 12
13 14
15 16
1 10
2 9
3 12
4 11
5 14
6 13
7 16
8 15
1 12
1 13
3 16
6 15
0 0
```

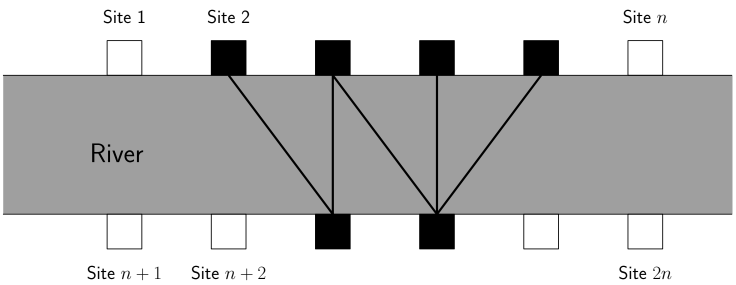
Sample Output

```
B
C
C
B
C
B
C
B
A
C
C
A
C
A
C
A
```

Problem B: Bridges

Introduction

The Widget Corporation has developed the greatest technological advance of the century: the megawidget. The megawidget uses revolutionary new technologies, thus completely new factories have to be built. The megawidget consists of n subunits; each subunit will be produced in a different factory. Widget production requires large amount of water, thus the new factories will be located on the banks of the Purple River. Widget Corporation located $2n$ suitable sites where factories can be built and there are n sites on each bank of the river (see the figure below). During the production of a widget, certain subunits should be moved very quickly from one factory to another. Unfortunately, it is impossible to build roads between the factories, since the terrain is too rough. Therefore, the only way you can move subunits between two factories if you build a bridge connecting them. By definition, a bridge has to cross the river, thus you cannot build a bridge that connects two factories on the same bank. Furthermore, due to technological limitations, two bridges cannot cross each other. The figure shows a possible layout of 6 factories and 5 bridges.



You are given a list specifying which pairs of factories should be connected by bridges, and your task is to find a location for each factory such that the bridges can be built without crossing. If this is not possible, then you can reduce the number of bridges by using helicopters between two factories (the two factories connected by helicopters can be on the same bank or they can be on opposite banks). Helicopters are very expensive, so we can afford to replace at most 2 bridges with helicopters. Therefore, your task is to replace exactly 2 bridges with helicopters, and build the remaining bridges in such a way that they do not cross.

Input

The input contains several blocks of test cases. Each case begins with a line containing two integers: $1 \leq n \leq 10000$, the number of factories, $2 \leq m \leq 10000$, the number of pairs that have to be connected. The next m lines contains two integers each, separated by a space. A line containing i and j ($1 \leq i, j \leq n$) means that the i-th factory and the j-th factory have to be connected (either by a bridge or by a helicopter connection). It is not sufficient that j can be reached from i via a series of bridges, it is important to have a direct connection between the two factories.

The input is terminated by a block with $n = m = 0$.

Output

For each test case, if there is a solution, then you have to output 2 lines specifying the pairs of factories that are to be connected by helicopters. Each of these two lines contains a pair of integers, separated by space. If it is not possible to solve the problem with 2 helicopters, then output the single line `'No solution.'` (without quotes).

Sample Input

```
8 7
1 2
2 3
2 4
2 5
3 6
4 7
5 8
0 0
```

Sample Output

```
2 3
3 6
```

Problem C: Chimes

Introduction

In the monastery of Hotoshopp, the monks perform a mysterious ritual every midnight. The exact details of this centuries-old ritual is secret, but it is known that at the beginning of the ritual (exactly at midnight) several chimes are sounded at the same time. There are many chimes in the monastery, but there are only n different types of chimes. Every chime produces a perfect triangle wave (see below for details) and chimes of the same type produce exactly the same sound. A Type 1 chime produces a sound of frequency 10Hz and the frequency of a Type i chime is double the frequency of a Type $i-1$ chime.

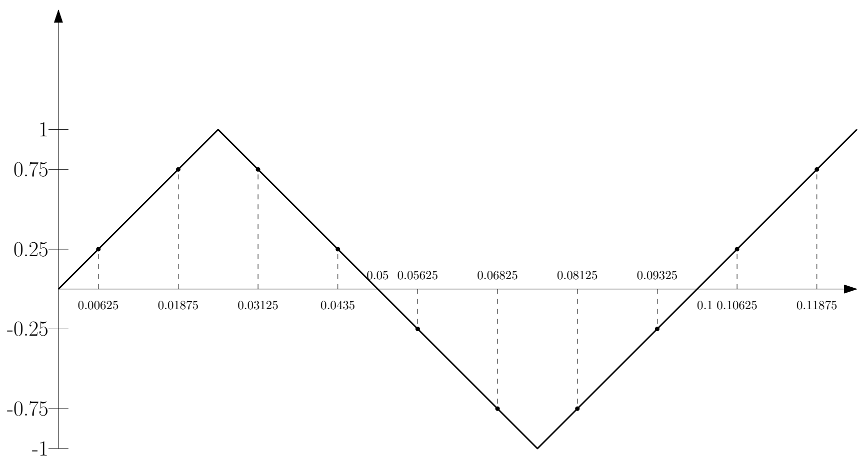
Usually not all the chimes are sounded during the ritual; there are very strict rules that determine which combination of chimes are sounded at midnight on a given day. Your task is to write a program that, given a digital recording of the sound, determines which chimes were used.

The input contains m sound samples which describe the sound, starting at midnight. The sampling rate is exactly twice the frequency of the chime with the highest frequency. For example, if there are $n = 4$ chimes, then the Type 4 chimes have frequency 80Hz, thus there are 160 samples per second. If the sampling frequency is f, then the i-th sample is taken

$$t(i) := \frac{i}{f} - \frac{1}{2f}$$

seconds after midnight ($i = 1, 2, \ldots, m$). That is, the samples are taken once every $1/f$ seconds, and the first sample is taken $1/(2f)$ seconds after midnight.

Each chime produces a sound wave that is a perfect triangle wave. The wave oscillates between 1 and -1 in each period. At midnight, the value of the sound wave is 0, and it is increasing. The sound of the chime will not change during the recording: it does not stop, it does not get quieter or louder. As an example, the figure below shows the triangle wave of a Type 1 chime. If the sampling rate is 80Hz, then the dashed vertical lines show the time points when a sample is taken.



The following table shows the values of the first 10 samples:

Sample number	Sample time	Sample value
1	0.00625	0.25
2	0.01875	0.75
3	0.03125	0.75
4	0.0435	0.25
5	0.05625	-0.25
6	0.06825	-0.75
7	0.08125	-0.75
8	0.09325	-0.25
9	0.10625	0.25
10	0.11875	0.75

If two chimes are sounded at the same time, then the two sound waves are simply added together; each sound sample is the sum of the corresponding two sound samples.

Input

The input contains several blocks of test cases. Each test case begins with a line containing an integer $1 \leq n \leq 10$, the number of chimes, and an integer $1 \leq m \leq 100000$, the number of sound samples. The number of samples is always a power of 2, and it is sufficiently large such that the input contains at least one full period from the sound of each chime. The next m lines contain the m sound samples: each line contains a real number.

The input is terminated by a block with $n = m = 0$.

Output

For each test case, your program should output n lines, each line containing a single integer. The integer in the i-th line should be the number Type i chimes that were sounded. It can be assumed that there are at most 1000 chimes of each type.

Sample Input

```
4 32
13.5
30.5
39
27
18.5
-6.5
-11
-7
7
11
6.5
-18.5
-27
-39
-30.5
-13.5
13.5
30.5
39
27
18.5
-6.5
-11
-7
7
11
6.5
-18.5
-27
-39
-30.5
-13.5
0 0
```

Sample Output

```
26
29
2
2
```

Problem D: DNA Regions

Introduction

A DNA sequence or genetic sequence is a succession of letters representing the primary structure of a real or hypothetical DNA molecule or strand, with the capacity to carry information. The possible letters are A, C, G, and T, representing the four nucleotide subunits of a DNA strand: adenine, cytosine, guanine and thymine bases covalently linked to phospho-backbone.

DNA sequences undergo mutations during the evolution of species, which means that some letters are randomly replaced with others. Therefore, the DNA sequences of two closely related species are very similar, and the difference increases as the distance between the species increases. The mutations do not occur with uniform frequency throughout the sequence; typically there are fewer mutations at the biologically important parts, since even a single mutation can be lethal at such a place. On the other hand, if a part of the sequence does not carry any biologically relevant information, then mutations on this part have no effect. It follows that if we compare the DNA sequences of two species and a particular region of the sequence contains fewer than the average number of mutations, then most probably this part of the sequence plays an important biological role. Therefore, it is of crucial importance to identify such regions. More precisely, a *conserved region* is a consecutive interval of the DNA sequence such that in this region at most p percent of the letters are different in the two sequences. Your task is to write a program that, given two DNA sequences, finds the longest conserved region.

Input

The input contains several blocks of test cases. Each case begins with a line containing two integers: $1 \leq n \leq 150000$, the length of the genetic sequences and $1 \leq p \leq 99$, the maximum percentage of mutated letters allowed in a conserved region. This is followed by two lines, each containing a DNA sequence of length n. The sequence contains only the letters `'A'`, `'C'`, `'G'`, and `'T'`.

The input is terminated by a test case with $n = 0$.

Output

For each test case, you have to output a line containing a single integer: the length of the longest conserved region between the two sequences. If there are no conserved regions in the input, then output `'No solution.'` (without quotes).

Sample Input

```
14 25
ACCGGTAACGTGAA
ACTGGATACGTAAA
14 24
ACCGGTAACGTGAA
ACTGGATACGTAAA
8 1
AAAAAAAA
CCCCCCCC
8 33
AAACAAAA
CCCCCCCC
0 0
```

Sample Output

```
8
7
No solution.
1
```

Problem E: Education Radio Programme

Introduction

After the very exciting 2006 ICPC finals in San Antonio (Texas), ACM has decided to start a new radio station that will broadcast news, music and useful programming tricks to the farmers in Texas. The radio station itself will not be in Texas; it will be located at Saratov State University (Russia), whose team won the 2006 finals. Who else could give better programming advice than the winners?

In order to reach every ranch in Texas, powerful antennas have to be installed at the radio station. These antennas are very expensive, and they can broadcast only at a limited angle. More precisely, the price of an antenna is proportional to the *square* of the angle. The cheapest antenna can broadcast across a 1-degree range and costs 1 unit of money; the second cheapest broadcasts across a 2-degree range and costs 4 units, etc. For example, if the radio station is at (0,0) and there are two ranches at $(0, 2)$ and $(2, 2)$, then a 45-degree antenna (costing 2025) can broadcast to both ranches at the same time, but a 44-degree antenna (costing 1936) can broadcast to at most one of the ranches. The angle of the antenna is always an integer, thus there are 360 different types of antennas.

Your task is to optimize the type and position of the antennas such that every ranch is reached and the total cost is minimized.

Input

The input contains several blocks of test cases. Each block begins with a line containing two integers: $1 \leq n \leq 5000$ is the number of ranches and $1 \leq m \leq 40$ is the maximum number of antennas that you can install. The next n lines contain 2 integers each; they describe the coordinates of the n ranches (these numbers are between 0 and 100000). The location of the radio station is always $(0, 0)$.

The input is terminated by a block with $n = m = 0$.

Output

For each test case, you have to output a single line containing an integer, the minimum cost of broadcasting to all the ranches using at most m antennas.

Sample Input

```
3 2
0 2
2 0
2 2
3 3
0 2
2 0
2 2
0 0
```

Sample Output

```
2026
3
```

Problem F: Fast Track

Introduction

Travelling in the Infinite Desert is very difficult, dangerous, and boring. Without roads and railway tracks, camel riding is the only form of travel that is available between the cities of the desert. The Desert Development Committee has decided to build a super fast train to connect the cities. The train is so fast that it can go only on a perfectly straight track; even the slightest turn would result in a catastrophe. Therefore, the train track that will be built should be one straight line segment, which means that not every city will be reached by the train. Your task is to design the track in such a way that the number of people who can use the train is maximized. The citizens of a city can only use the train line if the track goes 1 km or closer to the city centre.

Input

The input contains several blocks of test cases. Each case begins with a line containing a single integer $1 \leq n \leq 1000$, the number of cities. Each of the next n lines contain two real numbers $0 \leq x, y \leq 100000$, the coordinates of the city (in km) and an integer $1 \leq p \leq 100000$, the population of the city. It can be assumed that the distance between two cities is always more than 2 km.

The input is terminated by a test case with $n = 0$.

Output

For each test case, you have to output a line containing a single integer: the maximum number of people that the train can reach.

Sample Input

```
4
10 10.4 100
11.0 20 250
9 30 120
20.5 20 200
4
10 10.4 100
11.0 20 250
9 30 120
20.5 20 250
0
```

Sample Output

```
470
500
```

Problem G: Grass is Green

Introduction

This year exactly n people bought land in Squareville—including you. When someone buys land, then the first thing they do is to plant grass on the land; everyone wants to make sure that their grass is greener than the neighbour's. Depending on the type of grass, planting it has a certain cost. The next thing to do is to build a fence around the land; the cost of the fence depends of the type of the fence (green or white, with or without barbed wire, electric or not, etc.) Everyone has a very particular idea about the type of grass and the type of fence they want. In fact, everyone firstly decided upon the type of grass and fence, and then bought the largest piece of land they could afford (i.e., they could buy the land, the grass, and the fence). In Squareville, you can buy only square-shaped land, but you can buy any size you want. We assume that everyone plants grass across the whole area of land and everyone builds a fence around the full perimeter, i.e., on all four sides.

Having a larger garden means that you are more respected in Squareville. Therefore, you would like to know how many of the n people will have larger land than you. Your task is to write a program that calculates this number.

Input

The input contains several blocks of test cases. Each case begins with a line containing an integer $1 \leq n \leq 10000$, the number of people buying land and a real number $0 < c < 100$, the cost of a unit area of land. The next n lines describe the n people; the first of these lines describes you. Each line contains three real numbers: the amount $1 \leq m \leq 100000$ of money this person has, the cost $1 \leq g \leq 100$ of a unit amount of grass this person plants, and the cost $1 \leq f \leq 100$ of a unit length of fence this person builds.

The input is terminated by a block with $n = c = 0$.

Output

For each test case, you have to output a single integer: how many people have larger land than you. Thus, if you have the largest land, then output 0; if you have the smallest land, then output $n - 1$.

Sample Input

```
5 1
32.0 5.0 1.0
16.0 1.0 1.0
63.0 2.0 3.0
68.0 10.0 3.0
88.0 1.0 10.0
0 0
```

Sample Output

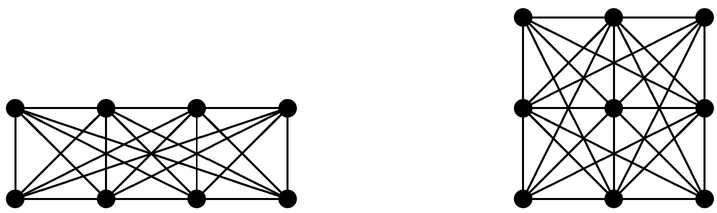
```
1
```

Problem H: Highways

Introduction

Hackerland is a happy democratic country with $m \times n$ cities, arranged in a rectangular m by n grid and connected by m roads in the east-west direction and n roads in the north-south direction.

By public demand, this orthogonal road system is to be supplemented by a system of highways in such a way that there will be a direct connection between any pair of cities. Each highway is a straight line going through two or more cities. If two cities lie on the same highway, then they are directly connected. If two cities are in the same row or column, then they are already connected by the existing orthogonal road system (each east-west road connects all the m cities in that row and each north-south road connects all the n cities in that column), thus no new highway is needed to connect them. Your task is to count the number of highway that has to be built (a highway that goes through several cities on a straight line is counted as a single highway).



Input

The input contains several blocks of test cases. Each test case consists of a single line containing two integers $1 \leq n, m \leq 300$, specifying the number of cities.

The input is terminated by a test case with $n = m = 0$.

Output

For each test case, output one line containing a single integer, the number of highways that must be built.

Sample Input

```
2 4
3 3
0 0
```

Sample Output

```
12
14
```

Problem I: Islands

Introduction

The ideal holiday for many people involves being on the white sand of a sunny beach on a tropical island, drinking cocktails under the palm trees and swimming, snorkelling and diving in the clear blue waves. The Algorithmic Archipelago is the ideal place for such a holiday as it consists of countless beautiful islands. Since many tourists spend their holidays (and money) on these islands, you decided to enter the tourism industry and buy an island. The cost of an island is proportional to its area, but the number of tourists that visit an island is proportional to the length of its beaches, i.e., the perimeter of the island. Your task is to select the island where the profit per investment ratio is the highest: where the ratio of the perimeter and the area is maximum. Out of curiosity, you would like the determine the minimum of this ratio as well.

In the input you are given the map of the archipelago as an $n \times n$ matrix. Each cell contains either water or land. An island is a connected area of land.

- Two cells sharing only a corner are not considered to be neighbours. (Therefore, there are 3 islands in the sample input).
- It can happen that an island contains one or more lakes. The perimeter of these lakes also contribute to the perimeter of the island.
- It can happen that the lake on an island contains further islands. These are considered to be separate islands, and the area of these islands does not contribute to the area of the island containing them.
- We assume that there is water in the area outside the map.

Input

The input contains several blocks of test cases. Each case begins with a line containing a single integer $1 \leq n \leq 400$, the size of the map. This is followed by n lines, each containing n characters. Character `'.'` means water and character `'X'` means land. It can be assumed that the map contains not only water.

The input is terminated by a test case with $n = 0$.

Output

For each test case, you have to output a line containing two real numbers, separated by a space: the maximum and the minimum of the ratio of the perimeter and the area. To avoid rounding problems, we accept solutions with a maximum of ± 0.001 error.

Sample Input

```
8
....XXXX
.XXX.XXX
XX.XX...
XX...XX.
X.XXX.XX
X.XXX.XX
X....XX.
XXXXXX..
0
```

Sample Output

```
1.923 1.666
```

Problem J: Jewel-eating Monsters

Introduction

According to ancient legends, deep in the Forest of Walking Trees lies a mystical cave called the Cave of Infinite Wealth. In the middle of a huge cavern there is a lake surrounded by stalactites and stalagmites. The dark, cold water of the lake has magical properties; if anyone tosses a gold coin into the lake at midnight and spends the night in the cave, then he will find that he has more money in the morning than he had in the evening. More precisely, if he had x gold coins (after tossing one into the water), then he will have αx gold coins in the morning.

The famous adventurer Gorgon Rattlesnake spent many years finding the cave. As you can imagine, the exact location is a well-guarded secret. Finally, after studying countless old maps, consulting with many famous sages, and making heavy use of GPS technology, he managed to arrive at the entrance of the cave.

Gorgon Rattlesnake spent n nights in the cavern, tossing a coin into the lake each night. After n nights, he travelled to the bustling city of Tse'padub. In Tse'padub, he visited the jewellery shop, and bought diamond rings to make transporting his valuables easier. Each diamond ring costs c gold coins, and he bought as many diamond rings as he could.

So far he was lucky. But shortly after he left the jewellery shop, he was attacked by jewel-eating monsters. These monsters did not harm him, but they ate all the diamond rings. For this reason, Gorgon Rattlesnake was left with no other valuables than the remaining gold coins. How many gold coins does he have?

Input

The input contains several blocks of test cases. Each case consists of four integers separated by spaces:

- The number $2 \leq x \leq 100000000$ of coins Gorgon Rattlesnake had when he arrived to the cave.
- The number $2 \leq \alpha \leq 100000000$ that the gold coins are multiplied by each night.
- The number $1 \leq n \leq 100000000$ of nights Gorgon Rattlesnake spent in the cave.
- The cost $2 \leq c \leq 100000000$ of a diamond rings.

It can be assumed (for mysterious reasons) that c is always a prime number.

The input is terminated by a block with $x = \alpha = n = c = 0$.

Output

For each test case, you have to output a line containing a single integer: the number of gold coins Gorgon Rattlesnake has.

Sample Input

```
52 7 1 691
27 4 56 379
94 7 63 103
38 2 88 557
62 11 86 433
0 0 0 0
```

Sample Output

```
357
198
82
177
394
```

Problem K: Knowledge Transfer

Introduction

In recent years it has become very fashionable to buy widgets and the demand for widgets increased above all expectations. In fact, the supply cannot keep up with the demand: every widgeteer is busy building widgets all day. The Association for Widget Production realized the problem and decided that new widgeteers have to be trained. Training widgeteers is a long process: a beginner's course on widgeteering lasts n days with a 2-hour training session on each day. Due to the lack of qualified teachers, only two group of students will be admitted to the course. To accommodate the different lifestyles of the students, the training sessions of the two groups take place at different times of the day. Each student is available only for a certain period of time on each day. Therefore, the assignment of the students into groups and the scheduling of the training sessions should be done in such a way that every student can attend every session.

Input

The input contains several blocks of test cases. Each block begins with a line containing two integers: the number $1 \le d \le 50$ of days the training lasts and the number $1 \le n \le 8000$ of students. The next n lines describe when the students are available. Each line contains d time periods, describing when the given student can attend a training session on the d days. A time period is given by a start time and end time separated by a dash. The time periods for the different days are separated by spaces. For example, a line beginning with `'8:00-19:30 16:45-20:00'` means that on the first day the student is available only between 8:00 and 19:30, while on the second day the student is available only between 16:45 and 20:00. Therefore, on the second day, the student can participate in a 2-hour training session that starts at 18:00, but it cannot participate in it if it starts at 18:01.

The input is terminated by a block with $d = n = 0$.

Output

For each test case, you have to output a way of assigning the students into two groups and a scheduling of the training sessions. Each of the first n lines of the output should contain the integer 1 or 2, depending on which group the given student is assigned to. The next d lines describe the scheduling of the training sessions. Each line contains two starting times (separated by a space): the starting times of the two training session on the given day. (Recall that each training session lasts exactly 120 minutes; the two training sessions can overlap). The starting time should be formatted like this: 9:45. A session cannot start later than 22:00 (i.e., it will end on the same day). If it is not possible to find a schedule that satisfies the requirements, then output `'No solution.'` (without quotes).

Sample Input

```
5 4
8:00-13:30 7:00-16:30 9:20-18:20 16:00-19:20 9:10-17:45
12:00-16:10 14:22-18:30 9:11-17:45 10:10-18:10 10:30-14:20
9:10-17:00 12:30-15:40 10:30-13:00 9:10-18:00 9:00-13:00
8:00-18:20 9:30-17:30 11:01-19:20 9:31-12:20 9:30-14:00
5 4
8:00-13:30 7:00-16:30 9:20-18:20 16:00-19:20 11:10-17:45
12:00-16:10 14:22-18:30 9:11-17:45 10:10-18:10 10:30-14:20
9:10-17:00 12:30-15:40 10:30-13:00 9:10-18:00 9:00-13:00
8:00-18:20 9:30-17:30 11:01-19:20 9:31-12:20 9:30-14:00
0 0
```

Sample Output

```
1
2
1
2
9:10 12:00
12:30 14:22
10:30 11:01
16:00 10:10
9:10 10:30
No solution.
```