Problem A
Triangles

Time Limit: 3 seconds

We want to create a long paper chain of triangles for the king of ACMland. This king requires that the various supply of grey coloured paper of triangles be connected together such that the shades of grey increase from light (white has value 0.0) to dark (black has value 1.0).

Each triangle can be fastened together with one or two other triangles along sides with equal length. The king is concerned only that triangles are correctly connected to their neighbours: he doesn't mind if triangles overlap each other when laid flat. To increase flair, each triangle of the chain must be a different shade of grey. For example, for the triangles shown in the figure below, we can connect together four of them to make a chain.

Input

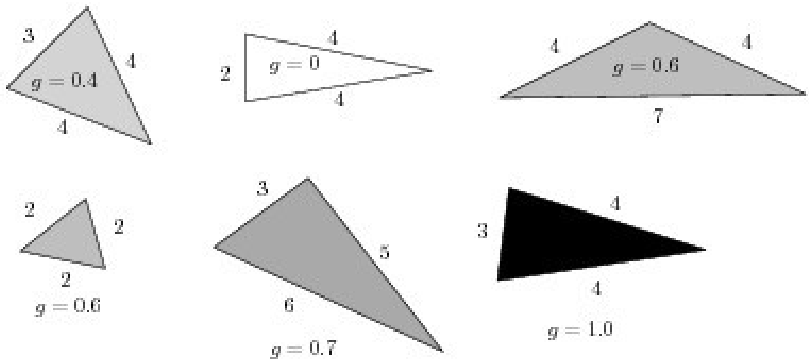
The first line in the input contains an integer, $1 \le n \le 1000$, which denotes the number of triangles available. Each of the following n lines contains three positive integers $0 < l_1, l_2, l_3 \le 10^6$ that denote the triangle side lengths and a real value $0.0 \le g \le 1.0$ with at most three digits after the decimal point that denotes its grey scale value.

Output

The output consists of a single positive integer that represents the longest chain of triangles that could be built and satisfy the criteria above.

Taken from the pictorial example below, we have the following sample input.

Sample Input	Output for the Sample Input
6 3 4 4 0.4 4 4 2 0.0 4 7 4 0.6 2 2 2 0.6 3 5 6 0.7 3 4 4 1.0	4


3
4
4
g = 0.4
4
2
4
g = 0
4
4
g = 0.6
7
2
2
2
g = 0.6
3
5
6
g = 0.7
4
3
4
g = 1.0

Problem B
Least Common Multiple

Time Limit: 3 seconds

Representing numbers in different bases can make certain problems much easier (it can also make the problem much harder). Here, we will be concerned with numbers in base 26. Let S be the set of all positive integers so that when they are represented in base 26, they only contain 0s and 1s. The first few elements of S (in base 10) are: 1, 26, 27, 676, 677, …

You will be given three numbers, a, b, and c, and you will need to find the least common multiple of 2^a, 3^b and 5^c that is in S and is also larger than a given value x.

Input

Input consists of multiple test cases (no more than 600). Each test case contains four integers, on a line by themselves, in the order a, b, c, x ($0 \le a < 50$, $0 \le b \le 3$, $0 \le c \le 2$, $0 \le x < 2^{63}$).

Output

For each test case, output the smallest number that is in S, is a multiple of 2^a, 3^b and 5^c and is larger than x. Output the number in base 26. If there is no number that satisfies the criteria above, output "No Solution" (without the quotes). See the sample output for the exact format.

Sample Input	Output for the Sample Input
1 1 0 5 1 0 1 20	Case 1: 110 Case 2: 111110

Problem C
Step by Step

Time Limit: 10 seconds

Bob wants to climb a mountain. The terrain is rough, however, so he can only place his feet in certain positions without falling down. Moreover Bob can move each of his feet only a certain distance from the other. Find out whether Bob can reach the summit.

Positions are given as three-dimensional Cartesian coordinates. Initially Bob's feet occupy two given positions. In each step he can move either his left foot or his right foot. While moving his left foot, his right foot stays fixed. The new position of the left foot must satisfy three criteria:

- It must differ from the position of the right foot in its *x*- or *y*-coordinate, or both.
- It must be at most a certain distance *d* away from the position of the right foot; the value of *d* depends on Bob's anatomy.
- It must not cross over the right foot. To be precise, let *P* be the plane that contains the position of the right foot and is perpendicular to the line between the position of the right foot and the point which has the *x*- and *y*-coordinates of the old position of the left foot and the *z*-coordinate (height) of the position of the right foot. Then the new position of the left foot must be on *P* or on the side of *P* that contains the old position of the left foot.

The constraints for moving the right foot are symmetrical: swap left and right in the above description. Bob needs to perform a sequence of such steps to put his feet into given target positions. Your task is to find the smallest distance *d* for which this is possible.

Input

The first line of the input contains an integer t specifying the number of test cases ($1 \le t \le 200$). Each test case starts with a line containing an integer n that specifies the number of positions on which Bob can place a foot during his climb ($4 \le n \le 100$).

Each of the following n lines contains the three integer coordinates x, y and z of a position ($-10^6 \le x, y, z \le 10^6$); the z-coordinate is its height. The first position is the initial position of Bob's left foot; the second is the initial position of his right foot. The next-to-last position is the target position of his left foot; the last is the target of his right foot.

Positions in a test case are distinct.

Output

For each test case, display the smallest integer that is greater than or equal to the smallest distance d for which Bob can reach the target positions. Note that d is at least as large as the distance between the initial positions of Bob's feet. Display "*impossible*" (without the quotes) if Bob cannot reach the target positions.

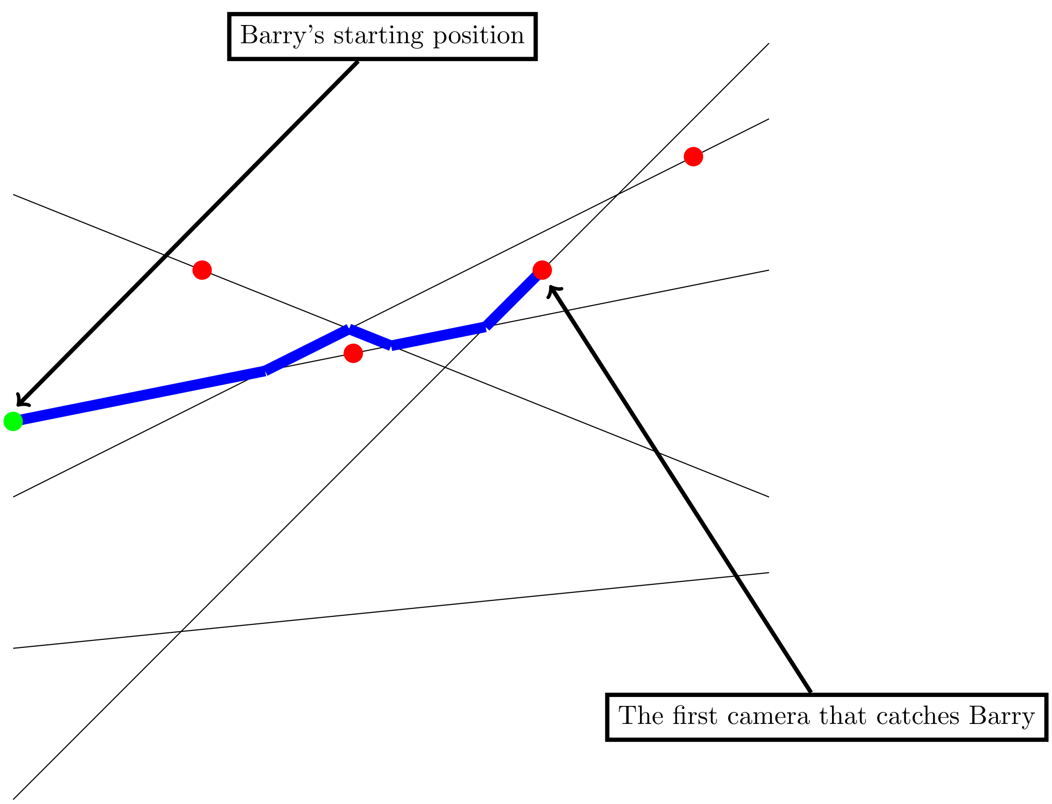
Sample Input	Output for the Sample Input
3 4 1 0 0 2 0 0 3 1 0 0 1 0 4 100 0 0 200 0 0 0 100 0 300 100 0 4 100 0 0 200 0 0 100 100 0 200 100 0	impossible 300 142

Problem D – Biker

Time Limit: 5 seconds

Barry the biker is planning a trip across the country on his motorcycle. His goal is to travel from his house and get to the east coast of the country. During the whole journey, Barry only travels East (never West). All of the roads are straight lines and there are no roads running north-south (that is, there are no vertical lines). The biker wants to see as many sites as possible, so whenever the biker reaches the intersection of two roads, he turns (i.e., he leaves the current road and starts traveling on the other road while keeping his direction to the East).

However, Barry loves to speed – and there are many speed cameras set up throughout the country. Given Barry's strategy, what is the first speed camera that Barry will pass by (if any)? In the following example, Barry would get caught by two cameras, but we only care about the first camera to catch him.



The roads are infinite lines and three roads will never meet at one location. The speed cameras and Barry's staring position will not be on an intersection.

Input

The first line of the input will contain a positive integer T ($1 \le T \le 20$) denoting the number of test cases to follow. Each test case will start with two integers n ($1 \le n \le 10^5$) and k ($1 \le k \le 100$) denoting the number of roads and cameras, respectively.

The next n lines will contain the two real numbers, m and b ($-10^8 \le m, b \le 10^8$), representing the slope and y-intercept of the road, respectively (that is, the equation of the line would be $y = mx + b$). The next k lines contain two numbers. The first is an integer i ($1 \le i \le n$) denoting which line this camera is on and the second is a real number representing the x-coordinate of the camera. The last line contains two numbers representing Barry's house (in the same format as the cameras).

Every coordinate given in the input will have absolute value no more than 10^8. All intersection points, Barry's house and cameras are all pairwise distinct.

Output

For each test case, output the number of the camera that Barry will get caught by first. The first camera in the input is camera 1, the second in the input is camera 2, etc. If Barry never gets caught by a camera, output -1.

Sample Input

```
1
5 4
2 0
5 -10
-4 30
10 -50
1 -30
3 2.5
1 4.5
4 7
2 9
1 0
```

Sample Output

```
3
```

Problem E – AFL Grand Final

Time Limit: 10 seconds

Tonight is the 2014 AFL (Australian Football League) Grand Final between the Hawthorn Hawks and the Sydney Swans. Hundreds of thousands of people across Australia will be watching as these two teams compete for bragging rights. Me and a group of my friends are wagering who will win the big game. There is a lot on the line (including my pride), so I have decided to put my emotions aside and have come up with a mathematical model of the game to determine who I should bet on. This model is very complex. In fact, it is too complex to describe during contest time. But I need your help for one of the steps in the process. I have collected statistics from the teams and I need to know which team is *closest to average*. Let me describe what I mean:

Let K be a multiset of n numbers. Each number has a weight associated with it. There are 2^n subsets of K and each has a corresponding weighted average. (By subset, here and in the following, we mean "sub-multiset".) For example, consider the multiset $K = [2, 6, 9, 13]$ with weights $[2, 1, 3, 1]$. Then the **weighted** averages are:

Subset	Average	Subset	Average	Subset	Average	Subset	Average
$\{\}$	Undefined	$\{2\}$	2	$\{6\}$	6	$\{2, 6\}$	$\frac{10}{3}$
$\{9\}$	9	$\{2, 9\}$	$\frac{31}{5}$	$\{6, 9\}$	$\frac{33}{4}$	$\{2, 6, 9\}$	$\frac{37}{6}$
$\{13\}$	13	$\{2, 13\}$	$\frac{17}{3}$	$\{6, 13\}$	$\frac{19}{2}$	$\{2, 6, 13\}$	$\frac{23}{4}$
$\{9, 13\}$	10	$\{2, 9, 13\}$	$\frac{22}{3}$	$\{6, 9, 13\}$	$\frac{46}{5}$	$\{2, 6, 9, 13\}$	$\frac{50}{7}$

Each team is given a score (H for the Hawks and S for the Swans). The *distance* between a score and the multiset K is the minimum difference (in terms of absolute value) between the score and the weighted average of any of the subsets of K. For example, if $H = 5$ and $S = 11$, then H is $\frac{2}{3}$ away from the subset $\{2, 13\}$ and S is 1 away from the subset $\{9, 13\}$. I need to know which team has the smaller distance from K, or in other words, which team is closer to the weighted average of a subset of K.

Input

The input will contain multiple test cases (no more than 20). Each test case will start with three integers n ($1 \leq n \leq 30$), H ($1 \leq H \leq 10^{12}$) and S ($1 \leq S \leq 10^{12}$). The next n lines will contain two integers, k_i ($1 \leq k_i \leq 10^{12}$) and w_i ($1 \leq w_i \leq 3$), denoting the ith element of K and its weight, respectively.

Output

For each test case, you need to determine which team is closer to the average. If they are the same distance from K, output `Tie`, otherwise output either `Hawks` or `Swans`.

Sample Input

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

Sample Output

```
Hawks
Swans
Tie
```

Problem F – Going Postal

Time Limit: 20 seconds

Pat the Postman is employed to collect the mail from all of the mail boxes in the city each night. He is accompanied on his collections by his white and black cat. The mail boxes are situated on certain street corners and the streets of the city are laid out as a rectangular grid. All streets are open to traffic in both directions and there are no restrictions on which way traffic can turn, so there is always a path along the streets that Pat the Postman can take to get to the next mail box.

The Postal Service guarantees that all mail posted by 6:00 pm will be picked up each night. Along with a regular mail service, the Postal Service provide an express mail service with a later time for pick up. Express mail is always delivered the next day, rain, hail or shine. There is only one express mail box in the city and it sits next to a regular mail box. This pair of mail boxes is situated at the closest point to the Mail Sorting Centre.

Pat the Postman starts by picking up the mail from the regular mail box closest to the Mail Sorting Centre and finishes with the express mail box. Pat times his travel so that he arrives at the first mail box at exactly 6:00 pm.

Given the locations of the mail boxes, the size of each city block, the speed at which Pat drives and the amount of time it takes to clear each mail box, what is the latest time someone could post express mail to ensure that it arrives at its destination the next day?

Input

The first line of input contains a single integer T ($T \leq 50$) being the number of test cases. For each test case the first line of input contains a single integer P ($1 \leq P \leq 20$), which is the number of locations where postboxes are situated. The next line consists of four integers: W, H, S and D. W and H ($1 \leq W, H \leq 100$) are the width and height in distance units of each city block. S ($1 \leq S \leq 100$) is the constant average speed in distance units per minute that Pat the Postman drives at. D ($1 \leq D \leq 5$) is the number of minutes it takes to pick up the mail from each mail box.

This line is followed by P lines specifying the mail box locations on the grid of streets. Each location is specified as the number of blocks east and north of the Mail Sorting Centre x_p, y_p ($0 \leq x_p, y_p \leq 100$). The first location is the location of the express mail box.

Output

For each test case output the latest time that express mail can be posted to guarantee it will be picked up. The time must be output in 24 hour format HH:MM ($00 \leq$ HH ≤ 23 and $00 \leq$ MM ≤ 59). If the mail is not in the express mail box by the time Pat the Postman arrives it will not be picked up, though it will be picked up if it is being posted at exactly the same time that Pat arrives.

Sample Input

```
2
4
1 1 10 1
0 0
10 10
10 0
0 10
4
2 1 1 2
0 5
6 0
10 5
5 10
```

Sample Output

```
18:08
19:08
```

Problem G
Gridstown Trip Planner

Time Limit: 5 seconds

The city of Gridstown is built on a 10 km by 10 km square axis-aligned grid with a 100 m grid-line spacing. Footpaths run along all grid lines. All addresses in Gridstown are given by non-negative integer x, y coordinates in the 100 m grid. The residents of Gridstown move around the city by a combination of walking on the footpaths and taking public transport. All residents walk at a rate of 100 m/minute.

Public transport consists of three train services: one perimeter service and two diagonal services. On all three services a train leaves from the starting point every hour, on the hour, 24 hours a day, seven days a week. The perimeter service runs anticlockwise around the perimeter of the city starting from the southwest corner of the city, (0, 0). One diagonal service starts from the southwest corner, (0, 0), and goes to the northeast corner and back. The other starts from the southeast corner and goes to the northwest corner and back. All trains stop on every 1 km grid point and take 2 minutes between stops. When they have returned to their starting point, their run is complete.

For example the perimeter service leaving from (0, 0) at 10:00 arrives at (10, 0) at 10:02, at (20, 0) at 10:04 and eventually returns to (0, 0) after 80 minutes. The time difference between a train's arrival at a station and its departure can be neglected but if two trains arrive at a station at the same time a transfer from one train to another is possible.

You have been asked by the city fathers to write a trip planner for Gridstown residents, which works out the latest time that one can leave a given start point and arrive at a given destination point at or before a specified time.

A route can involve at most a single train-ride section. Two or more consecutive train-rides, with no walks between them, are counted as a single section of the route.

The only information to be printed is the departure time.

Input

The first line of the input contains an integer *m* specifying the number of route queries ($1 \leq m \leq 1500$). Each of the following *m* line contains four integers $0 \leq x1, y1, x2, y2 \leq 100$ followed by the arrival time *t_dest*. Single spaces separate all entries. Time is given in 24 hour format with a 1 or 2 digit hour from 0 to 23, a colon and then a 2 digit number of minutes from 00 to 59.

Output

For each route query the output consists of the latest possible departure time from (*x1*, *y1*) that allows a traveller to arrive at (*x2*, *y2*) on or before time *t_dest* within a 24 hour window.

Sample Input	Output for the Sample Input
2 33 34 35 35 0:01 23 20 25 80 11:00	23:58 10:01

Explanation:

In the first example, the quickest route is to walk on the grid from (33, 34) to (35, 35), a distance of 300 m, which takes 3 minutes. So the departure time is 2 minutes before midnight.

In the second case the best option is to leave from (23, 20) at 10:01, walk for 3 minutes to (20, 20) in time to catch the first diagonal train to (50, 50) arriving at 10:10. Then catch the other diagonal train departing from (50, 50) at 10:10 and get off at (20, 80) at time 10:16. Walk 5 minutes to (25, 80), arriving at 10:21. Variants where one walks further, having alighted from the second train at an earlier or later stop, are also possible but only the departure time is required, which is the same for all such variants.

Problem H – Superstitious Socks

Time Limit: 10 seconds

Trudy is a very superstitious person. She owns n different socks all with different lengths. Once she wears two socks on one day, she will never wear that exact pair again. After $\binom{n}{2}$ days, she will throw the n socks away and buy more. She has come up with a plan to ensure that she will not wear the same pairs of socks on different days: Each day, she will look at all pairs that she has not yet worn and choose the pair whose lengths are as close as possible, breaking ties by selecting the pair with the smallest sock.

For example, if you have five socks of lengths 1, 2, 4, 6 and 15. The first five pairs of socks will be

Day	1	2	3	4	5
Socks	$(1,2)$	$(2,4)$	$(4,6)$	$(1,4)$	$(2,6)$
Difference	1	2	2	3	4

But Trudy didn't realize how hard it was going to be to track which pairs to wear, and now she needs your help. Given the lengths of the socks and the current day, can you tell Trudy which socks she should wear today?

Input

The input will contain multiple test cases (no more than 20). Each test case will start with two integers n ($2 \le n \le 100000$) and k ($1 \le k \le 100000$ and $k \le \binom{n}{2}$). The next line will contain n distinct integers denoting the lengths of Trudy's socks. Each length with be between 1 and 10^9, inclusive.

The output will end when $n = k = 0$.

Output

For each test case, output two integers, the lengths of the two socks that Trudy should wear on day k. Output the smaller sock first then the larger sock.

Sample Input

```
5 4
1 2 4 6 15
4 6
2 8 13 120
0 0
```

Sample Output

```
1 4
2 120
```

Problem I
Chaos Management

Time Limit: 3 seconds

Sydney is a great city, when they finish building it! In the meantime there will always be construction sites around the city and the locals will suffer the accompanying traffic chaos around each site.

The "Australian Chaos Management" ACM company, which specialises in supplying traffic controllers for construction sites, has a booming business and also has a policy of employing casuals only. Casual employees help the financial bottom line but they are difficult to manage because they are not always available for work.

After many years of using phones to call on an employee for a job, the ACM company decided to switch to a more efficient IT system for assigning employees to traffic control jobs. Employees are asked to nominate the time they are willing to work and the system will allocate them to jobs as needed. Each job can be assigned to at most one employee and each employee can be assigned to at most one job.

Your task is to program a small part of the system whose purpose is to report the maximum number of jobs for which the ACM company can provide traffic controllers.

Input

The first line contains an integer $0 < n \leq 100$ which denotes the number of cases to be considered. Each case starts with two integers, on a line by themselves, which are the number of casual employees N and the number of jobs M, $0 \leq N,M \leq 200$. Each of the next N lines contains the available time interval of one employee. Each of the following M lines contains the time interval of one job. All time intervals are described by a starting and an end time separated by a single space. The start time and the end time both use the *hh:mm* format ($00 \leq hh \leq 99$, $0 \leq mm \leq 59$). The start time is strictly before the end time.

Output

For each case, the output consists of a single line that starts with a single number that is the maximum number of jobs for which the ACM company can provide traffic controllers.

Sample Input	Output for the Sample Input
1 3 3 00:00 05:00 20:00 28:00 07:00 35:00 06:00 10:00 25:00 30:00 22:59 27:09	2