



Anupam Bhattacharjee Memorial Contest, 24th October, 2010

[You get 17 pages, 9 problems and 300 minutes]



Anupam Bhattacharjee was a Member of UVa Online Judge Algorithmic Panel

(Anupam Bhattacharjee (November 16, 1983 - September 6, 2010), a PhD student at the department of Computer Science in Wayne State University, went missing near Ludington Lighthouse on September 6, 2010 when a high wave swept him off the pier in an unfavorable weather. His body was recovered three days later.)

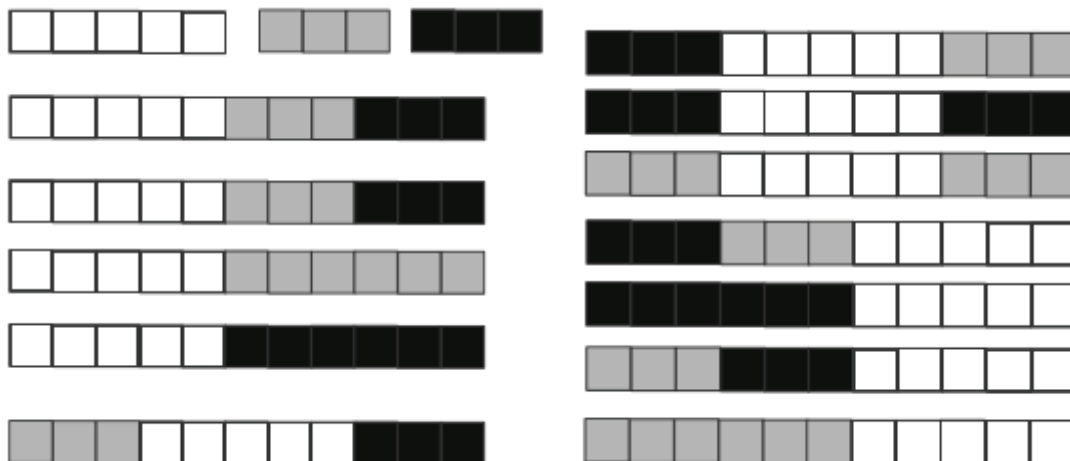


Anupam Bhattacharjee Memorial Contest, 24th October, 2010

<h1>A</h1>	<h2>Game of Blocks</h2>	
	Input	Standard Input
	Output	Standard Output

John decided to buy his son Johnny some mathematical toys. One of his most favorite toy is blocks of different colors. John has decided to buy blocks of c different colors. For each color he will buy googol (10^{100}) blocks. All blocks of same color are of same length. But blocks of different color may vary in length.

Johnny has decided to use these blocks to make a large $1 \times n$ block. He wonders how many ways he can do this. Two ways are considered different if there is a position where the color differs. The example shows a red block of size 5, blue block of size 3 and green block of size 3. It shows there are 12 ways of making a large block of length 11.



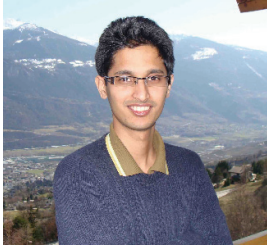
Input

Input starts with a positive integer $T \leq 25$. T test cases follow.

Each test case starts with an integer $1 \leq c \leq 100$. Next line consists a integer. i th integer $1 \leq len_i \leq 750$ denotes length of i th color. Next line is positive integer $n \leq 10^{15}$.

Output

For each case output case number followed by the number of ways Johnny can make the desired block modulo 100000007 (a prime number). See sample output for exact format.



Anupam Bhattacharjee Memorial Contest, 24th October, 2010

Sample Input	Sample Output
4 3 3 3 5 11 3 3 5 3 11111111111111 4 1 1 100 100 1000000 3 1 1 1 5	Case 1: 12 Case 2: 20634244 Case 3: 94126777 Case 4: 243

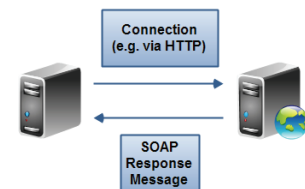
Problemsetter: Tanaeem M Moosa, Special Thanks: Md. Arifuzzaman Arif



Anupam Bhattacharjee Memorial Contest, 24th October, 2010

B	<h2>SOAP Response</h2>	
	Input	Standard Input
	Output	Standard Output

With increasing availability of broadband internet and significant progress in Web technology, Web Services are playing a crucial role in application development. The basic idea behind Web Service is, the server hosting the services can be queried and it will send back the response based on the query. Similar to all other forms of communication, a protocol must exist on how the messages should be communicated. One popular protocol is SOAP(Simple Object Access Protocol). As the name suggests, it deals with sending objects across client and server machine. Since objects are represented differently in different languages, the objects are sent in a common text based format known as XML. In this problem, we will be analyzing a simplified version of SOAP messages in XML format and evaluate queries based on the message.



The XML messages will be formatted as below:

- i) Each line consists of either an opening tag or a closing tag.
- ii) Each tag will have a name.
- iii) Each opening tag can have zero or more properties. The properties will have a value assigned to them. These names and values will have at least one character. The property names in individual tags will be unique. See examples below for exact formatting.
- iv) Closing tags will not have any properties.
- v) Tags can be nested inside another tag.

Example XML message and description:

```

<one valid="true" active="alive">
  <anestedtag>
</anestedtag>
</one>
  
```

In the above example, we have a tag named ‘one’ which has two properties, namely ‘valid’ and ‘active’. The property named ‘valid’ has a value of ‘true’ and ‘active’ has a value of ‘alive’. Each property information is separated from each other by a single space and there is also a space after tag name if the tag has any property. No other space occurs in the XML text other than those mentioned earlier. There is a tag named ‘anestedtag’ which is nested inside the tag named ‘one’. Note that, this tag doesn’t have any properties defined. Closing tags contain the same name of the corresponding opening tag, but is preceded by the ‘/’ character.



Anupam Bhattacharjee Memorial Contest, 24th October, 2010

You will be given a simplified soap message, containing only start and end tags with property information embedded in the start tag as the above format. Then you will have some query with some property name under some tag. You need output the value of the property under the specified tags or report such property is undefined.

Input

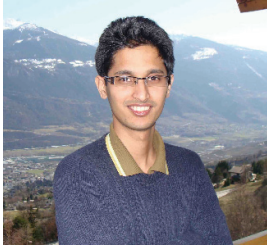
The first line of input will be a positive integer $\mathbf{T} \leq 10$, where \mathbf{T} denotes the number of test cases. Each case starts with a positive integer $\mathbf{n} \leq 1000$ and \mathbf{n} will always be even. The next \mathbf{n} lines each describes an opening or a closing tag. These lines will have at most **1000** characters. The last line of the input will always be the closing tag for the tag opened in the first line. No opening tag name will occur more than once. All input will be syntactically valid, that means all opening tags will also contain a corresponding closing tags in the appropriate line. The next line will contain a positive integer $\mathbf{q} \leq 1000$. Each of the next \mathbf{q} lines will describe a query. The query will consist of tag names separated by a dot('.') followed by few characters in the format `["<prop name>"]`, where `<prop name>` is one or more character forming a property name. See sample input for exact formatting.

All *names* and *values* in the input will consist of alphabets only.

Output

Each case of output will contain the case number in one line. Then there will be \mathbf{q} lines, each corresponding to the property value given in the query. If that particular property does not exist, the output will be 'Undefined' without the quotes. Note that, the input queries will always be syntactically valid, but they may refer to tags and properties that are not defined. If two tags are separated by a '.', it indicates that the tag to the right should be contained within the left tag. See sample **input/output** for further clarification. In output, there should not be any leading or trailing spaces in any of the line.

Sample Input	Sample Output
<pre>1 8 <c contest="running"> <d ballon="no"> </d> </c> 4 a.b.c.d["ballon"] a.c["contest"] a.b.c["contest"] c["contest"]</pre>	<pre>Case 1: no Undefined running Undefined</pre>



Anupam Bhattacharjee Memorial Contest, 24th October, 2010

<h1>C</h1>	<h1>Antonyms</h1>	
	Input	Standard Input
	Output	Standard Output

Little Rushel is preparing for his English test. Part of the test includes matching pairs of words as antonyms or synonyms. A pair of words are called synonyms of each other if both words mean the same thing. For example, “big” and “large” are synonyms. A pair of words are called antonyms of each other if each is the opposite of the other. For example, “big” and “small” are antonyms.

For his test, Rushel needs to memorize a huge list of words. This is really tedious and he would rather spend the time learning to program. So, he comes up with a clever scheme. He notices that if one word is a synonym of another, it is also the antonym of all words that are antonyms of that word. This means that he needs only remember that the words “big” and “large” are synonyms and that “big” and “small” are antonyms, and from that he can infer that “large” and “small” are also antonyms. Thus he does not need to memorize the relationship between all pairs of words, only enough for him to be able to deduce the answer for the rest. Of course, not all words are related (e.g. “big” and “cold”), but the examiner is a sensible person and will not ask meaningless questions like that.

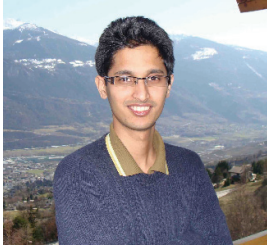
After a little more thought, Rushel observed that for a set of words that are related, the following rules apply:

1. If A and B are synonyms and B and C are synonyms, A and C are synonyms.
2. If A and B are synonyms and B and C are antonyms, A and C are antonyms.
3. If A and B are antonyms and B and C are synonyms, A and C are antonyms.
4. If A and B are antonyms and B and C are antonyms, A and C are synonyms.

Unfortunately for Rushel, his friends pointed out that the rules are too simple for a real human language like English. For example, if fed the data that “ice” and “water” are antonyms, and that “fire” and “water” are also antonyms, it would lead one to infer that “ice” and “fire” are synonyms (by Rule 4), which is obviously wrong. Rushel thinks that this is not really a good example (after all, it doesn't make a lot of sense), but he is still concerned that there might be equally nonsensical relationships in the words that he is learning. He has already prepared lists of synonyms and antonyms to learn, he thinks it will be enough to just check if any of them contain any contradictions.

A set of relationships is said to contain a contradiction if some of the relationships imply or state that a certain pair of words are synonyms while some of the other relationships imply or state that the same pair are antonyms.

Please help Rushel, who is too young to be allowed into this contest, by writing a program that takes as its input sets of pairs of synonyms and antonyms and verifies that there are no contradictions in them.



Anupam Bhattacharjee Memorial Contest, 24th October, 2010

Input

The first line of the input will contain a single integer **T** all by itself, $T \leq 100$. T test cases follow.

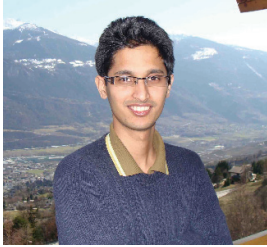
Each test case starts with two numbers **N** and **M** ($1 \leq N, M \leq 1000$). **N** lines follow, each containing a pair of words separated by one or more spaces. Then another **M** lines follow, also containing pairs of words separated by one or more spaces. Words will be made up of lowercase English characters only ('a' to 'z'), and will never be more than 20 characters long. The first set (the first **N** pairs) is the set of synonyms and the second set (the latter **M** pairs) is the set of antonyms.

Output

For each case output one line of the form “**Case X: Y**”, where **X** is the test case (starting from 1) and **Y** is either “**YES**” (if the test case contains no contradictions) or “**NO**” (if the input contains contradictions).

Sample Input	Sample Output
3 2 2 big large small tiny big small tiny huge 2 1 xyz abc abc def def xyz 1 3 fire flame fire ice ice water water fire	Case 1: YES Case 2: NO Case 3: NO

Problemsetter: Muntasir Azam Khan, Special Thanks: Sohel Hafiz



Anupam Bhattacharjee Memorial Contest, 24th October, 2010

D	New Land	
	Input	Standard Input
	Output	Standard Output

Once upon a time there lived an old wise king. He had only one son. The prince had the idea that he (the prince) was about to be the future king; that's why he became too lazy day by day. This made the wise king a bit worried because his lazy son couldn't be the correct man for the throne. He couldn't sleep; his kingdom was in need of a perfect king, not the lazy king.

The old king grew older. One day, while doing his regular works, he found an excellent idea. The next day he called his son. He said the prince that he wanted the prince to have his own kingdom. The prince became excited with joy. The king continued that the prince can have a land from the king's kingdom, but he should start walking after sunrise and cover a rectangular area before sunset. Then the land would be given to the prince.

The lazy prince thought, "It would be an easy task!" That's why he wanted to find the maximum rectangular area in king's land. But there were some rocks in the kingdom, and the prince didn't want any rocks in his new land. He would rather take nothing, but no rocks.

The king's kingdom can be thought of an $m \times n$ grid, where m is the number of rows and n is the number of columns. Each cell in the grid is a small rectangular land whose area is 1. If a land contains rock it will be denoted by 1, otherwise it will be denoted by 0. The prince can walk in the sides of the cells and he can either take a cell (land) or ignore it, but he can't take a part of a cell. Now your task is to find the maximum rectangular area the prince could cover.

Input

Input starts with in integer T (≤ 50) denoting number of cases.

Each case starts with a blank line. Next line contains two integers m and n ($1 \leq m, n \leq 2000$) denoting the rows and columns of the kingdom respectively.

Each of the next m lines will contain some integers in the following form

$k \ c \ p_1 \ p_2 \ p_3 \ \dots \ p_k$

meaning that in that row, the first p_1 columns are of type c , the next p_2 columns are of type 'opposite of c ', the next p_3 columns are of type c , the next p_4 columns are of type 'opposite of c ' and so on. You can assume that c will be either 0 or 1, and opposite of 0 is 1 and vice versa. You can also assume that $p_1 + p_2 + p_3 + \dots + p_k = n$ and p_i is positive.



Anupam Bhattacharjee Memorial Contest, 24th October, 2010

Output

For each case, print the case number and the maximum area the prince could cover.

Sample Input	Sample Output
2 5 7 5 0 1 2 1 2 1 3 0 5 1 1 3 1 1 5 1 4 0 1 1 4 1 4 1 2 3 1 1 3 3 2 0 2 1 2 1 1 2 3 1 1 1 1	Case 1: 12 Case 2: 3

Problemsetter: Jane Alam Jan, Special Thanks: Md. Towhidul Islam Talukder

Notes

- 1) For the first case the land covered by the prince is

```
0110110
0000010
1000001
0100001
1100010
```

- 2) For the second case the land covered by the prince is

```
001
100
101
```

Warning: Input file can be very large, use faster input, e.g. scanf.



Anupam Bhattacharjee Memorial Contest, 24th October, 2010

<h1>E</h1>	<h2>Where to Run</h2>	
	Input	Standard Input
	Output	Standard Output

Last night you robbed a bank but couldn't escape and when you just got outside today, the police started chasing you. The city, where you live in, consists of some junctions which are connected by some bidirectional roads.

Now since police is behind, you have nothing to do but to run. You don't know whether you would get caught or not, but if it is so, you want to run as long as you can. But the major problem is that if you leave a junction, next time you can't come to this junction, because a group of police wait there for you as soon as you left it, while some other keep chasing you.

That's why you have made a plan to fool the police as longer time as possible. The plan is, from your current junction, you first find the number of junctions which are safe (no police are there) and if you go to one of them; you are still able to visit all the safe junctions (in any order) maintaining the above restrictions. You named them 'Elected Junction' or EJ. If there is no such junction; you stop running, because you lose your mind thinking what to do, and the police catch you immediately.

But if there is at least one EJ, you can either fool around the police by staying in the current junction for 5 minutes (actually you just hide there, so the police lose your track thinking which road you might have taken), or you can choose to go to any EJ. The probability of choosing to stay in the current junction or to go to each of the EJ is equal. You can fool the police (by hiding) multiple times in a city, but of course the above conditions should be satisfied. And you have decided not to stop in the middle of any road, because you have the fear that, if you stop in the middle of any road, then the police would surround you from both ends.

Now, given the map of the city and the required time for you to travel each road of the map; you have to find the expected time for the police to catch you.



Input

Input starts with in integer T (≤ 100) denoting number of cases.

Each case starts with a blank line. Next line contains two integers n ($1 \leq n \leq 15$) denoting the



Anupam Bhattacharjee Memorial Contest, 24th October, 2010

number of junctions and m , denoting the number of roads in the city. The junctions are numbered from 0 to $n - 1$.

Each of the next m lines contains three integers $u v w$ ($0 \leq u, v < n$ and $0 < w \leq 100$ and $u \neq v$) meaning that there is a road between junction u and v and you need w minutes to travel the road. Your home is in junction 0 and you are initially in your home. And you may safely assume that there can be at most one road between a pair of junctions.

Output

For each case, print the case number and the expected time in minutes. Errors less than 10^{-5} will be ignored.

Sample Input	Sample Output
3 3 2 0 1 3 1 2 3 4 6 0 1 75 0 2 86 0 3 4 1 2 1 1 3 53 2 3 10 5 5 0 1 10 1 2 20 2 3 30 1 3 20 3 4 10	Case 1: 16 Case 2: 106.833333333333 Case 3: 90

Problemsetter: Jane Alam Jan, Special Thanks: Sohel Hafiz



Anupam Bhattacharjee Memorial Contest, 24th October, 2010

F	<h2>Gift Packing</h2>	
	Input	Standard Input
	Output	Standard Output

Alfred is a researcher and he also likes to give people the largest gift with smallest possible budget. Being busy in research all the time he cannot go and deliver his gifts in most cases. Now that his closest friend John is getting married, so he plans to send him the largest gift box just to surprise John and others as usual.

Alfred has spent 7 long days to buy gifts for his friend and packed all the gifts in nine circular boxes of equal radius. He now plans to put all of them in a large square shaped box. He can put them in smallest possible box as shown in Figure 1 but in that case his gift box would not be as large as he would like it to be. But he cannot put them in an arbitrary large box because then the circular boxes will move freely and be damaged while in transport. So now Alfred tries to discover a way to put nine circular boxes in a largest Possible Square shaped box so that none of the boxes can move (all are in rigid position). He comes to know from his researcher friend Erich the way to put nine circular boxes in largest possible square shaped box so that none of the circular boxes can move. This is shown in figure 3.

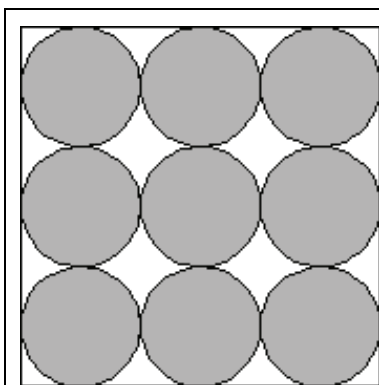


Figure 1: Nine circles of equal radius in rigid packing in smallest possible square shaped box.

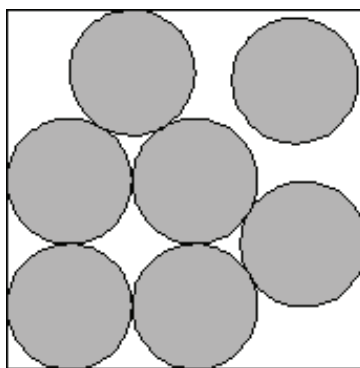


Figure 2: Seven circles of equal radius in non-rigid packing in smallest possible square shaped box.

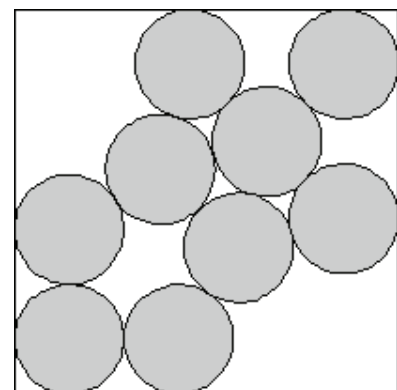


Figure 3: Nine circles of equal radius in rigid packing in largest possible square shaped box.

Now given the radius of the circular boxes your job is to find out the length of the sides of the square shaped box. But it is still not as simple as you think because of his other requirements. Being a researcher Alfred has also discovered that if the radius of the circular boxes is an integer then the



Anupam Bhattacharjee Memorial Contest, 24th October, 2010

length of the sides L can be expressed in a expression like: $\frac{p + \sqrt{q} + \sqrt{r} + \sqrt{s + \sqrt{t}}}{2}$, where p, q, r, s, t are all integer values and $q < r$. But as he is busy doing his other research so you have to help him solving this problem. So given the radius R of the circular boxes your job is to find out such integer value of p, q, r, s and t such that $L = \frac{p + \sqrt{q} + \sqrt{r} + \sqrt{s + \sqrt{t}}}{2}$

Input

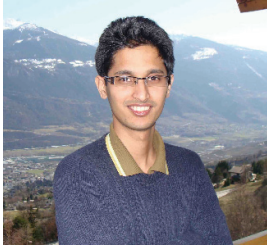
The input file contains around **1000** lines of input. Each line contains an integer R ($0 < R < 1501$) that denotes the radius of all the circular boxes. A line containing a single zero terminates input. This line should not be processed.

Output

For each line of input produce one line of output. This line contains the serial of output followed by **5** integer numbers. These five numbers denote the value of p, q, r, s and t respectively (Here $q < r$). You can safely assume that values of p, q, r, s and t will safely fit in a 64-bit signed integer. **Note that the values shown in the sample output is not correct.** The values are there just to show the correct output formatting.

Sample Input	Sample Output
1	Case 1: 78 456 1100 356 233456
2	Case 2: 89 2134 3800 8765 18374621
0	

Problemsetter: Shahriar Manzoor, Special Thanks: Derek Kisman



Anupam Bhattacharjee Memorial Contest, 24th October, 2010

<h1>G</h1>	<h2>Travel Company</h2>	
	Input	Standard Input
	Output	Standard Output

A travel company is planning to launch their bus service in a new route. So they conducted a survey and made a list of all possible roads connecting different cities. Each of the roads has a certain amount of income based on current fare. But at the same time, each road has some expenses too (this includes fuel and maintenance cost, staff payments, taxes and tribute to labor union which is recently approved by the Government). The travel company is looking for a cyclic route. That is, the bus will start from any city, then visit one or more other cities each exactly once and return to the starting city. The company is also concerned with the profit on the route. In fact the directors of the company have a strict requirement of a profit ratio strictly greater than P . Otherwise they will not launch the service. A profit ratio for a route is the ratio between the total income to the total expenses for that route. One of your friends works in that company and he asks for a little help from you. All you have to do is to determine if there exists such route, so that the company has a profit ratio of P .

Input

The first line of input is a positive integer T ($T \leq 100$), the number of test cases. Then T test cases will follow. Each of the test cases will start with 3 integers. N, R, P ($2 \leq N \leq 100, 0 \leq R \leq 9900, 1 \leq P \leq 100$). N, R and P represents number of cities, number of road links and the expected profit ratio. Then R lines will follow. Each line will contain 4 integers A_i, B_i, I_i, E_i ($0 \leq A_i, B_i < N, 0 \leq I_i \leq 5000, 1 \leq E_i \leq 5000$). (A_i, B_i) represents directed road link from city A_i to B_i . I_i and E_i are the income and expenses of the road link respectively. Each test case will be followed by a blank line. You may assume that $(A_i, B_i) \neq (A_j, B_j)$, if $i \neq j$ and $A_i \neq B_i$ for any i .

Output

For each test case, output one line in the format “**Case k: s**”. Here k is the case number starting from 1 and s is a string either “**YES**” if there is a cyclic route for which the profit ratio is greater than P or “**NO**”, if there is no such route. See the sample input output for details.



Anupam Bhattacharjee Memorial Contest, 24th October, 2010

Sample Input	Sample Output
3 5 8 3 0 1 17 8 1 0 10 5 1 2 11 5 1 4 5 3 2 3 13 7 3 1 9 4 4 3 11 1 3 0 11 6 5 8 3 0 1 17 8 1 0 10 5 1 2 11 5 1 4 5 3 2 3 13 7 3 1 9 4 4 3 11 2 3 0 11 6 5 8 2 0 1 17 8 1 0 10 5 1 2 11 5 1 4 5 3 2 3 13 7 3 1 9 4 4 3 11 5 3 0 11 6	Case 1: YES Case 2: NO Case 3: YES

Problemsetter: Md. Towhidul Islam Talukder, Special Thanks: Jane Alam Jan

Note: For the first case, consider the cycle 1 -> 4 -> 3 -> 1. Total income is 25 (5 + 11 + 9) and total expense is 8 (3 + 1 + 4). Hence the profit ratio is $25 / 8 > 3$.



Anupam Bhattacharjee Memorial Contest, 24th October, 2010

<h1>H</h1>	<h2>Brick Game</h2>	
	Input	Standard Input
	Output	Standard Output

There is a village in Bangladesh, where brick game is very popular. Brick game is a team game. Each team consists of odd number of players. Number of players must be greater than 1 but cannot be greater than 10. Age of each player must be within 11 and 20. No two players can have the same age. There is a captain for each team. The communication gap between two players depends on their age difference, i.e. the communication gap is larger if the age difference is larger. Hence they select the captain of a team in such a way so that the number of players in the team who are younger than that captain is equal to the number of players who are older than that captain.

Ages of all members of the team are provided. You have to determine the age of the captain.

Input

Input starts with an integer T ($T \leq 100$), the number of test cases.

Each of the next T lines will start with an integer N ($1 < N < 11$), number of team members followed by N space separated integers representing ages of all of the members of a team. Each of these N integers will be between 11 and 20 (inclusive). Note that, ages will be given in strictly increasing order or strictly decreasing order. We will not mention which one is increasing and which one is decreasing, you have to be careful enough to handle both situations.

Output

For each test case, output one line in the format "Case x : a " (quotes for clarity), where x is the case number and a is the age of the captain.

Sample Input	Sample Output
2 5 19 17 16 14 12 5 12 14 16 17 18	Case 1: 16 Case 2: 16

Problemsetter: Md. Arifuzzaman Arif, Special Thanks: Shamim Hafiz



Anupam Bhattacharjee Memorial Contest, 24th October, 2010

I	$N + \text{NOD}(N)$	
	Input	Standard Input
	Output	Standard Output

Consider an integer sequence N where,

$$N_0 = 1$$

$$N_i = N_{i-1} + \text{NOD}(N_{i-1}) \text{ for } i > 0$$

Here, $\text{NOD}(x)$ = number of divisors of x .

So the first few terms of this sequence are $1\ 2\ 4\ 7\ 9\ 12\ 18\dots$

Given two integers A and B , find out the number of integers in the above sequence that lies within the range $[A, B]$.

Input

The first line of input is an integer T ($T < 100000$), that indicates the number of test cases. Each case contains two integers, A followed by B ($1 \leq A \leq B \leq 1000000$).

Output

For each case, output the case number first followed by the required result.

100 → 1 2 4 5 10
 100 50 25 20

 9 DIVISORS

Sample Input	Sample Output
3 1 18 1 100 3000 4000	Case 1: 7 Case 2: 20 Case 3: 87

Problemsetter: Sohel Hafiz, Special Thanks: Shamim Hafiz