

ACM-ICPC 2012 Asia Regional Dhaka Site Online Preliminary Contest A - Phone No Mnemonics

Time Limit: 2000 ms

Memory Limit: 32768 KB

We the human beings or the homo sapiens tend to remember words (Meaningful or not so meaningful) better than numbers. That is why we all have names (Not serial numbers) and we use telephone directories as we don't recall a person's telephone number when we meet him and for similar reasons in programming languages we have identifiers. It would have been nice to convert our phone numbers to a meaningful words or a sequence of meaningful words so that our friends can remember it easily. In fact there are some websites like [phonespell](#) to convert your phone number to a sequence of meaningful words (There may be some digits in between or at the beginning or at the end).



Conventional Keypad



Modified Keypad

So now we need to be specific on how this conversion happens. If you look at a conventional keypad (Shown on the left) you will find that more than one character maps to a digit, eg: A, B and C maps to 2, D, E and F maps to 3 etc. So each Sequence of letters maps to an unique number but each number maps to many sequence of letters of which only some may contain meaningful words. For example the number 8674869 can be written as **TOPGUN9**. Given a phone number (Not containing more than 15 digits) your job is to find whether it can be expressed as a word, sequence of words and digits. You may notice that in the conventional keypad, digit 0 and 1 does not map to any character which is demoralizing for people who have

these two digits in their phone number more than once. So in this problem we will use a modified keypad (Shown on the right). As a result digit 7 maps to P or Q or R instead of P or Q or R or S. Similarly the mapping of 8 and 9 is also different. In the original keypad 0 and 1 did not match to any letter but in the modified keypad 1 maps to S and 0 maps to T.

Input

The input file contains only one set of input. This set starts with two integer N ($0 < N < 130000$) which denotes the total number of words in the dictionary and Q ($0 < Q < 8001$) which denotes the total no of phone numbers to convert. Each of the next N lines contain contains an English word (All uppercase letters and at most **10** characters). These N words makes the dictionary for this problem. Each of the next Q lines contain a telephone number (Containing at most **15** digits). Please note that all words of the dictionary are actually taken from a reasonably standard English dictionary and words found as a part of the phone-number must have at least **3** letters. So you have to ignore words like NO, GO, TO, IF etc from the dictionary.

Output

For each query produce one line of output. This line contains a string where some digits are from the original phone number while the other sequence of digits are converted to meaningful words (having at least 3 letters) from the given dictionary. If there is more than one possible way report the one that converts most digits as part of meaningful words. If there is still a tie report the lexicographically smallest one.

Sample Input Output for Sample Input

```
3 1
ABACUS
GUN          TOPGUN9
TOP
0674869
```

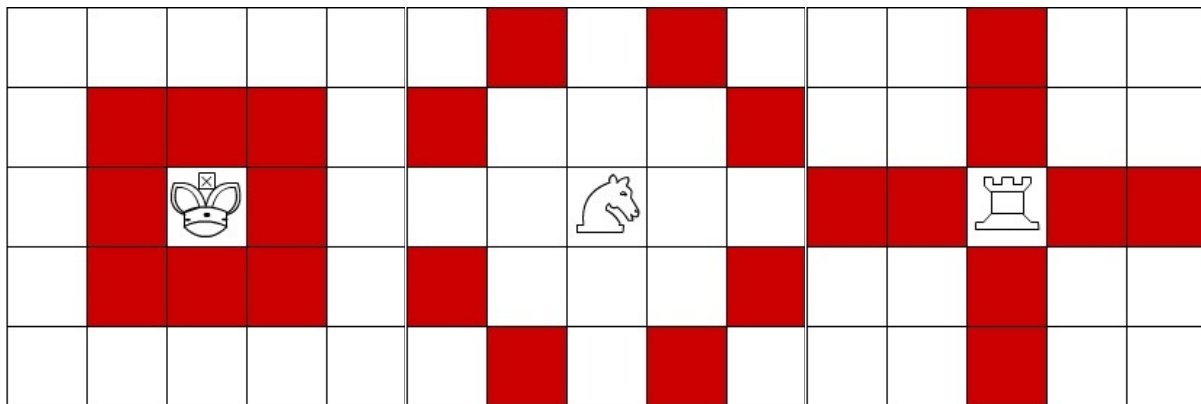
Problem Setter: Shahriar Manzoor
Courtesy to: Md. Mahbubul Hasan

ACM-ICPC 2012 Asia Regional Dhaka Site Online Preliminary Contest B - Playing with Chess Pieces

Time Limit: 3000 ms











Memory Limit: 262144 KB

Chess requires high intelligence to play. But don't worry, we will not ask you to play chess against us! Those who do not know chess, Chess is played in a 8*8 board with 32 pieces in total. In this problem we will only consider Rook, Knight and King. For those who are not familiar with these pieces:



- King: A king can attack all the eight adjacent cells as shown in the leftmost picture.
- Knight: A knight can attack 8 cells as in middle diagram.
- Rook: A rook can attack all the cells in its row and column. In the right most picture we have shown a rook and the cells it can attack.

Our task is, given a $4 \times n$ board ($1 \leq n \leq 6$) and countless number of Rooks, Knights and Kings. How many ways are there to place rooks, knights and kings so that no one attacks another. For example, in the left diagram, the upper knight attacks a king, so it can not be a valid placement. However, in the right diagram no one attacks other pieces. So this one is a valid placement. Note, even an empty board should be considered as valid placement.

Input

First line of the test file contains number of test cases, **T** which is at most **50**.

Then **T** cases follow. Each case starts with two integers **n** ($1 \leq n \leq 6$). Hence follows **4** lines of **n** characters describing the chess board. Characters will be either '.' Or 'x'. A cell containing an 'x' means you can not place any chess piece there.

Output

For each case, output case number followed by number of valid configurations. Since the answer can be quite large, output the answer modulo **3851919**. If you are wondering why the mod is so peculiar, it is the numerical form of 'chess'.

Sample Input Output for Sample Input

```

2
1
.
.
.
.
Case 1: 35
3
.
.
Case 2: 11
.x.
xxx
xxx
xxx

```

Problem Setter: Md. Mahbubul Hasan
 Courtesy to: Jane Alam Jan, Sohel Hafiz

ACM-ICPC 2012 Asia Regional Dhaka Site Online Preliminary Contest C - The Queue Tree

Time Limit: 2000 ms

Memory Limit: 32768 KB

Professor X is very excited with his new idea. He is trying to explain a queue with a branching process. For this to happen he must derive the generation tree out of a queue. So he went to the nearby burger shop early in the morning and started to collect data of customer arrival. He collected a lot of data and now ready to process them. The data consist of the arriving and service time of each customer. Now he wants to process the data and build a tree out of them. But as there is a lot of data, he asked for your help. He describes the idea to you:

The data contains the arriving time and the service time of each customer. While a customer is receiving service the other customer needs to wait in a queue. The customer who comes earlier gets service earlier. Consider the very first customer to be of **generation 0**. All the customers arrive the shop during he gets service are considered to be his child and will form **generation 1**. While any customer of **generation 1** is getting service, the customers arrive are the child of the customer who is getting service and will form **generation 2**. In general, when a customer arrives, if there is a customer who is getting service, he will be a child of that customer and will be a member of **generation (n + 1)** where **n** is the generation of the customer who is getting service. If there is no customer getting service at the time of arrival of the new customer, he will start a new tree as **generation 0** again. You need to do a simple thing. Just compute the average size of the generations. To calculate this you need to find the average size of each of the generations (generation 0, generation 1 etc) over the separate trees and then find the average of these values. If any tree does not have any generation it will have a size of 0 for that generation. You only need to consider generations that appear in at least one tree.

Input

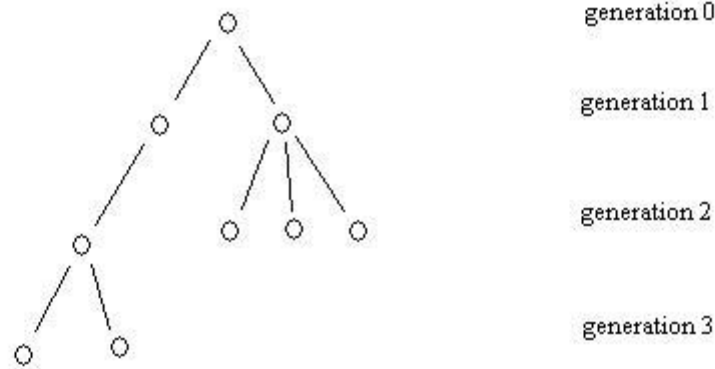
The first line of input will denote the number of test case **T** ($T \leq 50$). Then there will be **T** test cases to follow. Each of the test case will start with an integer **N** ($N \leq 10000$), the number of customers. Then there will be **N** lines each containing two integers **A_i** and **S_i** ($0 < A_i, S_i \leq 100000$) where **A_i** is the arrival time and **S_i** is the service time for the *i*-th customer. The customer information will be sorted on their arrival time in non-decreasing order. If the arrival times of two customers are same, the customer who is mentioned earlier will get service earlier.

Output

For each input, output the case number and the average size of generations. See sample for exact format. An output with absolute or relative error of 10^{-6} will be considered correct.

Note

In the second sample **customer 1** will get service from time **1 - 5** (5 unit service time). During this time **customer 2** and **customer 3** arrives and become child of **customer 1**. **Customer 2** will get service from time **6 - 8** (3 unit service time). **Customer 4** arrives during this time and becomes child of **customer 2**. The service time for **customer 3** is **9 - 12**. During this time 3 customers arrive and become child. Thus we get the following trees:



Average size of **generation 0**: $(1 + 1) / 2 = 1$

Average size of **generation 1**: $(2 + 1) / 2 = 1.5$

Average size of **generation 2**: $(4 + 1) / 2 = 2.5$

Average size of **generation 3**: $(2 + 0) / 2 = 1$

The grand average is $(1 + 1.5 + 2.5 + 1) / 4 = 1.5$

Output for Sample Input

Sample Input

```
2
4
2 3
3 4
4 5
30 7
12
1 5
4 3
5 4
8 5
9 5
9 1
11 2
13 1
14 1
35 2
36 2
38 2
```

Case 1: 1.00000000
Case 2: 1.50000000

Problem Setter: Md. Towhidul Islam Talukder

Courtesy to: Md. Mahbulul Hasan

ACM-ICPC 2012 Asia Regional Dhaka Site Online Preliminary Contest D - Spiral Robot

Time Limit: 2000 ms

Memory Limit: 32768 KB

The head scientist of a Bangladeshi robotics company called Jilapee robotics just made a kind of robot called 'Bidhushekhur Alpha' (Yes, he liked Professor Shonku series a lot).

A robot of that kind can move on specially built grids and follow two types of basic instructions. Those are the following:

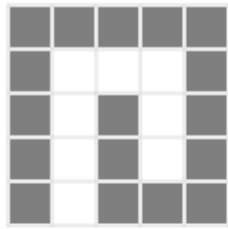
1. **MOVE x** : Moves forward x steps towards current direction.
2. **TURN** : Turns 90 degrees clockwise.

Note that the current direction in instruction **type 1** is the direction of the robot right before making the move. It's only possible to place the robot in the grid with its direction facing any of the four cardinal directions. (Namely east, west, north and south)

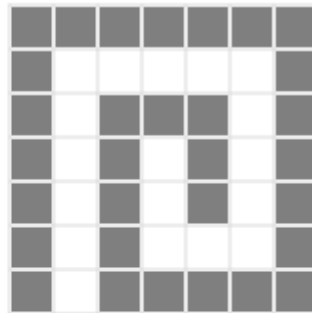
The robot is still primitive and can only follow a pre-loaded set of instructions written in its memory chip. Wanting to make its movements look interesting, the scientist used the following algorithm to create a set of instructions. After following these instructions, the robot's path will look like a spiral.

```
set K = Some odd number
K = K - 1
MOVE K
TURN
while K > 0
    MOVE K
    TURN
    MOVE K
    TURN
    K = K - 2
```

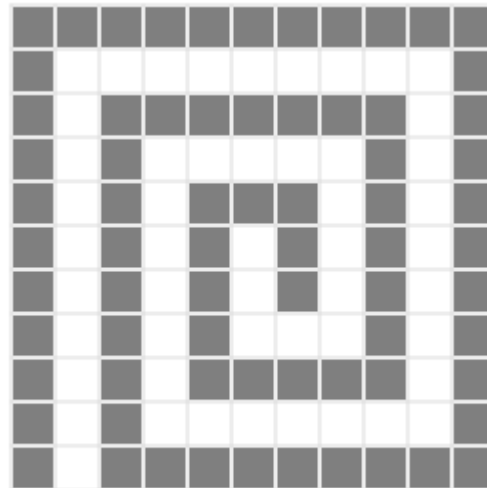
The **K** mentioned in this algorithm is called the order of the spiral.



$K = 5$



$K = 7$



$K = 11$

Examples of spirals with different orders, all facing northward (considering top of this page as north), starting at the cell at lower left corner (south-west)

Your friend (a crazy man, to say the least) is thinking of doing an interesting experiment. He got a $R \times C$ grid placed on top of a table. There are B bombs in B cells of the grid. He bought $4 \times R \times C$ Bidhushekhori Alphas of order K .

For each cell in the grid, he'll do the following:

- Step 1: Put one of the robots on that cell, setting its direction towards north and switch it on.
- Step 2: Put one of the robots on that cell, setting its direction towards south and switch it on.
- Step 3: Put one of the robots on that cell, setting its direction towards east and switch it on.
- Step 4: Put one of the robots on that cell, setting its direction towards west and switch it on.

Note that the initial cell of a particular robot can contain a bomb and it won't explode before the beginning of the robot's movement. After performing one of the above steps, he will wait till the robot finishes processing its in-memory set of instructions. After that he will put the robot in a shelf and will never use it again. If a robot steps on one of the cells with a bomb, the bomb will explode immediately. If a robot goes out of the grid, it will fall from the table and explode. In case of an explosion, your friend will throw away the junk. After each explosion by a bomb, a new bomb will be put in place of the exploded one.

You want to quickly calculate the number of robots that will be in the shelf after the experiment.

Input

The first line of input will contain a single integer T , the number of test cases. In the next line there will be four numbers R , C , K and B . These are the quantities mentioned in the problem statement. The each of the next B lines will contain two integers each. i -th line will have the 1-based row and column number of the i -th bomb. No two bombs will be in the same cell. Cells with bombs are not given in any particular order. Ranges of all the numbers are given in the constraints section.

Output

For each test case, print case number and an integer containing the number of robots in the shelf after the experiment. See the sample output for exact format.

Constraints

$T \leq 5$
 $1 \leq R, C \leq 200$
 $3 \leq K \leq 199$, odd
 $0 \leq B \leq R * C$
 $1 \leq \text{Row number of a bomb} \leq R$
 $1 \leq \text{Column number of a bomb} \leq C$

Sample Input Output for Sample Input

```

4
3 3 3 2
2 2           Case 1: 1
2 3           Case 2: 4
3 3 3 0       Case 3: 0
5 6 7 1       Case 4: 156816
2 1
200 200 3 0

```

Problem Setter: Mir Wasi Ahmed

Courtesy to: Rujia Liu

ACM-ICPC 2012 Asia Regional Dhaka Site Online Preliminary Contest E - Two Points

Time Limit: 2000 ms

Memory Limit: 32768 KB

In the land of mangoes, there were two mangoes called Good and Fresh. They don't like each other since their birth. They get really angry the moment they see each other. Now that they are matured and ripe (well, only physically), it's time to get off the tree and enjoy the real mangworld. But how can one enjoy when there is so much hatred in the mind for the other? To keep the mangworld peaceful, we must force them to live separately. They shouldn't be able to see each other.

They live in a city called Mangsister. This city can be considered as a flat rectangle with vertices $(-10^9, -10^9)$, $(-10^9, 10^9)$, $(10^9, 10^9)$ and $(10^9, -10^9)$. An infinite wall, with really small thickness separates the whole city into two areas. It was built after the Mango World War II. Mangoes from one part cannot see people from the other part. In the land of mangoes, there is a house on every point with integer co-ordinates; a mango house is so small that it can be considered as a point. Additionally, no house is built on the wall. For the cause of world peace and harmony, you are asked to pick **any** two integer co-ordinate locations within the city such that they are on the opposite sides of the wall. Good and Fresh will be ordered to stay on houses built on those points. Now, who will get the first point and who will get the second? Well, not our headache.

Input

First line of input will contain an integer T , the number of test cases. Then there will be four integers X_1, Y_1, X_2, Y_2 . The wall goes through points (X_1, Y_1) and (X_2, Y_2) . You can be sure that the points (X_1, Y_1) and (X_2, Y_2) are not the same and **both will not be on one of the edges of the city**. So there will always be a home for Good and a home for Fresh.

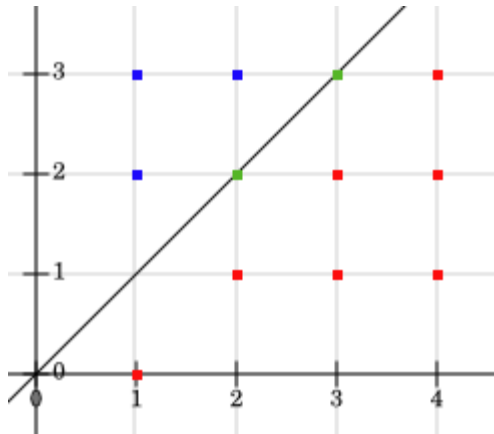
Output

Print case number and four integers: x and y co-ordinates of houses of Good and Fresh respectively. **They should be on opposite sides of the given line and within the boundary of the city (can be on the edges, see constraints)**. People living in these points (if any) will be rehabilitated - you shouldn't worry about them. Look at sample output for exact format.

Constraints

$T \leq 10000$
 $-10^9 \leq X_1, Y_1, X_2, Y_2 \leq 10^9$
 $-10^9 \leq \text{Co-ordinates of output points} \leq 10^9$

Note



In the first sample, the line goes through green points (2, 2) and (3, 3). Blue and red points are on the opposite sides of the line. A pair of points consisting of any of the blue points and any of the red points can be a possible solution. One of the solutions is (2, 3), (3, 2).

Sample Input Output for Sample Input

```

2
2 2 3 3      Case 1: 2 3 3 2
2 2 2 3      Case 2: 0 0 3 5

```

Problem Setter: Mir Wasi Ahmed

Courtesy to: Shahriar Manzoor

ACM-ICPC 2012 Asia Regional Dhaka Site Online Preliminary Contest F - Who Wins

Time Limit: 2000 ms

Memory Limit: 32768 KB

In the land of the Great Sultan Mahbub, his Chief minister of education believes that arranging frequent programming contests is a good idea to give people the impression that their land is progressing rapidly in terms of digital and creative education. As he does not have any idea whatsoever what these contests are all about and what the implications of a good or a bad contest, he just does not care about the quality of the problemset and other technical aspects of the contest. His job is only to make sure there are enough contests of whatever quality every year.

Now, after the end of yet another programming contest, he realizes that the experts have left his team and gone elsewhere. Now given the number of problems solved by each team in a contest, he can't even figure out who is the winner. You must help him.

Input

First line of the input file is a positive integer T ($T \leq 100$) which denotes the number of test cases. Each test case contains two numbers, the first one ($0 \leq f \leq 100$) is the number of problems solved by the first team while the second one ($0 \leq s \leq 100$) is the number of problems solved by the second team. There are only two teams in this contest as everyone else are either irritated with so many programming contests or have gone to the more exciting Gaming Competition.

Output

For each test case produce a line of the form "**Case X: Y team wins.**". X is the serial number of the test case while Y is equal to '**First**' if the first team solves more problem than the second team while it is '**Second**' if the second team solves more than the first one. You can safely assume that the two teams will never solve the same number of problems which can be explained by the magpiehole principle. (Hint: You are not encouraged to search magpiehole principle in google or your discrete math book. You should be able to solve this problem even if you don't know it. :)).

Template Code

```
#include<stdio.h>

int main()
{
```

```
int cases, caseno, first, second;

scanf(" %d",&cases);
for(caseno=1;caseno<=cases;caseno++)
{
    scanf(" %d %d",&first,&second);
    printf("Case %d: ",caseno);
    if(....) //Replace the ... with your code
        printf("First team wins.\n");
    else
        printf("Second team wins.\n");
}

return 0;
}
```

Sample Input Output for Sample Input

```
3
10 5          Case 1: First team wins.
5 10          Case 2: Second team wins.
2 1           Case 3: First team wins.
```

Problem Setter: Mohammad Mahmudur Rahman

Courtesy to: Shahriar Manzoor