1. Simple Chatbot

Every website will have an FAQ section. When Bibi helps his uncle to create a website, he prepares more. As he knows that the in this current trend, people prefers to chat rather than making a phone call, he especially creates the simple chatbot where the chatbot will answer the user’s question according to the data available in “data.txt”. This file contains list of questions keyword and answers separated by “#” as the following example:

```
food list#We have Burger, Pasta, Steak, and also Sushi
cough medicine#For cough medicine, we have Bisolvon, Konidin, and also Woods
flu medicine#For flu, we have Procold, Neosep, and Rhinos
```

Input Format

- The first line will be T, the number of test-case.
- For each test-case, user will type in a *string* as keyword to be looked for

Output Format

- Search the keyword contained in the questions and display it with the format “Case *T: <<result>>“. If data is not found : <<result>> will be “Not Found”*
- If there are more than 1 result, only show the first one found.

Constraint

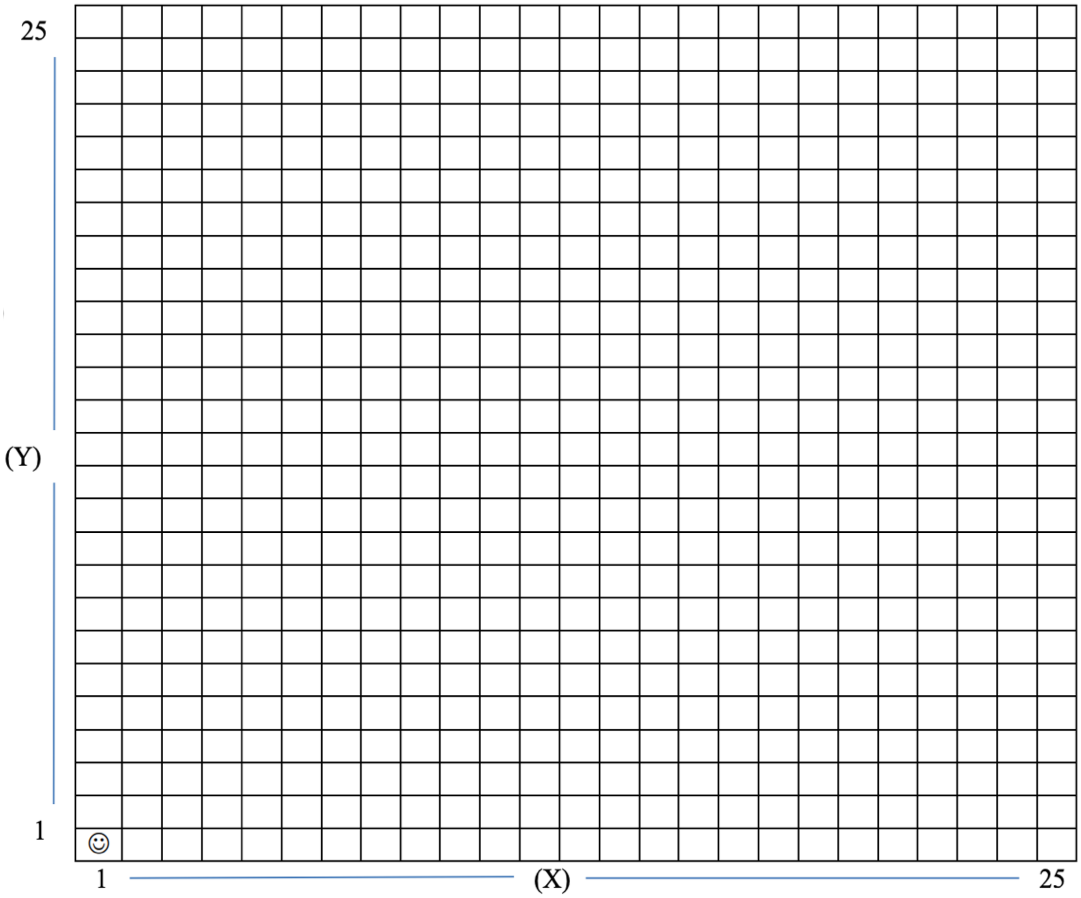
- There will only be 1000 lines of data inside data.txt
- The length of each question and answer pair is at most 499 characters
- All characters are lower letter case

Sample Input	**Sample Output**
2 flu cough	Case 1: For flu, we have Procold, Neosep, and Rhinos cough Case 2: For cough medicine, we have Bisolvon, Konidin, and also Woods

2. **Maze Prize Game**

   Jojo is having a New Year holiday, and he decided to visit their grandparents in his hometown. Both grandpa and grandma are pleased to see him and have heard that Jojo is a brilliant student at BEE University. So they decided to give Jojo a surprise. If Jojo solves the puzzle, grandpa and grandma will give him a prize accordingly.

   The puzzle is just like a maze, where there is a stick figure, and Jojo can decide the starting coordinate [X, Y]. Then, Jojo is allowed to pick a random instruction card. This instruction card will contain paired instructions and movement steps with options: RIGHT, LEFT, UP, and DOWN. The puzzle board has the size of 25 x 25, and the coordinates start with 1, with the following illustration:



   Jojo needs to solve the puzzle in time to obtain the prize rewards as in the table priority rules below:

   1. PrizeA will be obtained if the stick figures reach coordinate where X is equal to Y
   2. PrizeB will be given if the stick figures reach coordinate where X is an odd number
   3. PrizeC will be given if the stick figures reach coordinate where X is an even number

   To ensure that Jojo can solve the puzzle in time, he decided to create the program where the instructions list will be in testdata.in as in the following format: INSTRUCTION#STEP "testdata.in" example:

```
5 3
4
RIGHT#4
LEFT#2
UP#3
DOWN#1
```

If the position after movement is greater than 25, then the coordinate will be leftmost for X or bottom for Y (which is 1). If the position after movement is less than 1, then the coordinate will be rightmost for X or topmost (which is 25).

FORMAT INPUT

The first line of Input will be X and Y, the starting coordinate.
The second input line will be N, the number of instructions that must be read. Then, N lines contain the direction and number of steps that should be taken, separated by the # symbol.

FORMAT OUTPUT

The output will be the coordinate the stick figures are finally at and the prize Jojo wins.
The format will be: X Y<<space>>PrizeName.

CONSTRAINT

- X and Y must be between 1 and 25
- N must be between 1 and 1000
- Instructions are between: RIGHT, LEFT, UP, and DOWN
- Steps are integers. 0 ≤ steps ≤ 25

Sample Input	Sample Output
5 3 4 RIGHT#3 LEFT#2 DOWN#1 UP#5	6 7 PrizeC
20 22 4 RIGHT#3 LEFT#2 DOWN#1 UP#5	21 1 PrizeB

3. Searching Efficiency

Today, Lili has just learnt about Binary search algorithm in the algorithm and programming class. The lecturer told her that binary search is more efficient than linear search to be used. Lili was curious about it, and she wanted to test the performance of the two algorithms. Coincidentally, Mrs Aninda, her lecturer, asked her to develop an application to search each student's final exam score by their name. Help Lili to check how many steps efficiency Lili can obtain if she implements a binary search algorithm compared to the linear search algorithm.

FORMAT INPUT

The Input will consist of several lines.

The first line consists of an integer N defining the number of student data the program will store. After that, N lines follow, consisting of each student's name and score data.

After all data has been read, the following line will consist of an integer T, which is the test case number. After that, T lines will follow, consisting of string F, the student's name that will be searched. Please be noted that the student's name to be searched is **NOT case-sensitive.** All data given is sorted already in ascending order.

FORMAT OUTPUT

The program will have 1 line output with the following format "Y Z".

Where Y represents the student's score being searched, and Z represents the efficiency test result as defined below:

Z will consist of the number of step efficiency that Lili obtained by using binary search compared to linear search; however, if the number of steps in binary search is EQUAL with the number of steps in linear search, print "SAME". If the number of steps in binary search is bigger than linear search, print "-1" and if the name is not found in the data, print both "-1 -1".

CONSTRAINT

$1 \leq T \leq 20$

$0 \leq N \leq 500$

$0 \leq \text{Score} \leq 100$

$3 \leq |\text{Student's name, F}| \leq 50$

SAMPLE INPUT	SAMPLE OUTPUT
10 Adora#97 Andi#80 Ani#76 Bibi#100 Biru#92 Farah#76 Feliza#68 Xavier#99 Yeruma#80 Zamora#65 4 Andi Biru Adora yaruma	80 SAME 92 4 97 -1 -1 -1

Note: It is guaranteed that the data is already sorted ascendingly.

4. Fractal B

You have a function in C programming language that can print a fractal pattern.

```
void fractal_B(int x) {
      int i;
      if (x <= 0) return;
      else {
            fractal_B(x - 1);
            for (i = 1; i <= x; i++) printf("*");
            printf("\n");
            fractal_B(x - 2);
      }
}
```

Given an integer X, if the fractal_B function is called, determine how many asterisk characters will be printed.

CONSTRAINT

- $1 \leq X \leq 80.$

FORMAT INPUT

The Input consists of one line, which is an integer X

FORMAT OUTPUT

The output will be the number of asterisk characters that are printed

SAMPLE INPUT	SAMPLE OUTPUT
1	1
2	3
3	7
80	160500643816367005

Hint: The answer is still enough to be stored in the long data type.

Hint: The fractal function above will have exponential time complexity. To achieve full point in this problem, you need to analyze the given fractal function, convert the fractal function into iteration to determine how many asterisks will be printed.