Greetings From Globussoft

- Given below are 5 Programming questions, you have to solve any 3 out of 5 questions.

- These 5 questions you can attempt in any technology like C/C++, java, .Net, PHP

- To solve these 3 questions you've max. 3 hours.

- While Solving these questions you are not allowed to use any Search Engine like Google, Yahoo, Bing ...

All the best for your test

Globussoft

QUESTION - 1

The manhattan distance between two points $A(x_1,y_1)$ and $B(x_2,y_2)$ is defined as following:

$$D(A,B) = |x_1 - x_2| + |y_1 - y_2|$$

Given N points $A_1, A_2, ..., A_N$, for each point A_i you need to calculate the minimum $D(A_i , A_j)$ ($j \neq i$).

Input

- The first line contains a positive integer N ($1 \leq N \leq 200000$).
- The i-th line of the next N lines contains two integers x and y which are co-ordinates of the i-th point($0 \leq x, y \leq 10^7$)

Output

- Print N lines, in which the i-th line contains the minimum distance for the i-th point.

Example

```
Input:
4
0 0
0 1
1 0
1 1

Output:
1
1
1
1
```

QUESTION – 2

N ($1 \leq N \leq 50,000$) cows conveniently numbered 1, ..., N are driving in separate cars along a highway in Cowtopia. Cow i can drive in any of M different high lanes ($1 \leq M \leq N$) and can travel at a maximum speed of S_i ($1 \leq S_i \leq 1,000,000$) km/hour.

After their other bad driving experience, the cows hate collisions and take extraordinary measures to avoid them. On this highway, cow i reduces its speed by D ($0 \leq D \leq 5,000$) km/hour for each cow in front of it on the highway (though never below 0 km/hour). Thus, if there are K cows in front of cow i, the cow will travel at a speed of $\max(S_i - D*K, 0)$. While a cow might actually travel faster than a cow directly in front of it, the cows are spaced far enough apart so crashes will not occur once cows slow down as described.

Cowtopia has a minimum speed law which requires everyone on the highway to travel at a a minimum speed of L ($1 \leq L \leq 1{,}000{,}000$) km/hour, so sometimes some of the cows will be unable to take the highway if they follow the rules above. Write a program that will find the maximum number of cows that can drive on the highway while obeying the minimum speed limit law.

Input

The first line contains the four integers N, M, D, and L. For the next N lines, line i+1 contains the integer S_i.

Output

Print a single integer denoting the maximum number of cows that can take the highway.

Example

```
Input:
3 1 1 5
5
7
5

Output:
2
```

We can obtain two cows by putting either cow with speed 5 first and the cow with speed 7 second.

QUESTION – 3

You are trying to build a house, but unfortunately you currently have only four available walls with side lengths a, b, c, and d. You want your house to be as big as possible, so you would like to know the largest possible area of any quadrilateral you can construct with these four side lengths.

Input

The first line contains the integer T ($1 \leq T \leq 2{,}000$), the number of tests. Each test contains a single line with four real numbers: a, b, c, and d ($0 < a, b, c, d < 1{,}000$). Note that it will always be possible to form a valid quadrilateral with these lengths; that is, the sum of any three side lengths will be strictly larger than the other one.

Output

For each test case, print a single line containing the largest possible area. Your output will be accepted if it is within 0.01 of the official answer.

Example

```
Input:
2
1 2 1 2
0.5 0.5 0.5 0.5

Output:
2.00
0.25
```

For the first test case, it is optimal to construct a rectangle, and for the second, a square is optimal.

QUESTION – 4

The next MechaGodzilla invasion is on its way to Earth. And once again, Earth will be the battleground for an epic war.

MechaGodzilla's army consists of many nasty alien monsters, such as Space Godzilla, King Gidorah, and MechaGodzilla herself.

To stop them and defend Earth, Godzilla and her friends are preparing for the battle.

Problem specification

Each army consists of many different monsters. Each monster has a strength that can be described by a positive integer. (The larger the value, the stronger the monster.)

The war will consist of a series of battles. In each battle, the weakest of all the monsters that are still alive is killed.

If there are several weakest monsters, but all of them in the same army, one of them is killed at random. If both armies have at least one of the weakest monsters, a random weakest monster of MechaGodzilla's army is killed.

The war is over if in one of the armies all monsters are dead. The dead army lost, the other one won.

You are given the strengths of all the monsters. Find out who wins the war.

Input specification

The first line of the input file contains an integer T specifying the number of test cases. Each test case is preceded by a blank line.

Each test case starts with line containing two positive integers NG and NM – the number of monsters in Godzilla's and in MechaGodzilla's army. Two lines follow. The first one contains NG positive integers – the strengths of the monsters in Godzilla's army. Similarly, the second one contains NM positive integers – the strengths of the monsters in MechaGodzilla's army.

Output specification

For each test case, output a single line with a string that describes the outcome of the battle.

If it is sure that Godzilla's army wins, output the string "Godzilla".

If it is sure that MechaGodzilla's army wins, output the string "MechaGodzilla".

Otherwise, output the string "uncertain".

Example

input:
```
2

1 1
1
1

3 2
1 3 2
5 5
```

output:
```
Godzilla
MechaGodzilla
```

QUESTION – 5

Everyone tried it, but only few chosen ones succeeded. It is a hard task with an unclear path, but a famous end – should you reach it. Many compare it to finding the Holy Grail, or even to finding Waldo. The task is to find a perfect rhyme.

Problem specification

Given is a wordlist **L**, and a word **w**. Your task is to find a word in **L** that forms a perfect rhyme with **w**. This word **u** is uniquely determined by these properties:

- It is in L.
- It is different from w.
- Their common suffix is as long as possible.
- Out of all words that satisfy the previous points, u is the lexicographically smallest one.

Notes

A prefix of a word is any string that can be obtained by repeatedly deleting the last letter of the word. Similarly, a suffix of a word is any string that can be obtained by repeatedly deleting the first letter of the word.

For example, consider the word *different*.

This word is both its own prefix and suffix. Its longest other prefix is *differen*, and its longest other suffix is *ifferent*. The string *rent* is its yet another, even shorter suffix. The strings *eent* and *iffe* are neither prefixes nor suffixes of the word *different*.

Let **u** and **v** be two different words. We say that **u** is lexicographically smaller than **v** if either **u** is a prefix of **v**, or if *i* is the first position where they differ, and the *i*-th letter of **u** is earlier in the alphabet than the *i*-th letter of **v**.

For example, *dog* is smaller than *dogs*, which is smaller than *dragon* (because *o* is less than *r*).

Input specification

The input file consists of two parts. The first part contains the wordlist **L**, one word per line. Each word consists of lowercase English letters only, and no two words are equal.

The first part is terminated by an empty line.

The second part follows, with one query word **w** per line.

You may assume that in either part of the input, the length of a word will be no more than 30. And the number of words in each part of the input will be no more than 250000. The input file will be less than 5MB.

Output specification

For each query in the input file output a single line with its perfect rhyme. The output must be in lowercase.

Example

input:
```
perfect
rhyme
crime
time

crime
rhyme
```

output:
```
time
crime
```