Filippos Anargyros Katsaros and the GODS

Your friend, Filippos Anargyros Katsaros, came across an amazing discovery. While on his afternoon walk he stumbled upon a tablet. After he took the tablet home and examined it, he discovered that the tablet describes how to communicate with Zeus. After returning home, he quickly documented his discovery and wrote a 329 page report explaining the phenomenon. Upon further review he determined that the Ancient Secret Greek order who had made the tablet did not count in normal time but only in seconds. Your friend would like to report the time in a more understandable way. In order to help your friend out, you must write a program that will convert the time from only seconds to units of days, hours, minutes, and seconds.

Details of the input
The first line of input will be a positive integer, *n*, indicating the number of cases. Each of the next *n* lines of input will consist of a single positive integer, *s*, which represent one time value expressed in seconds.

Details of the output
For each of the *n* test cases, you will display four lines of output, one for each of the converted time measurements (Days, Hours, Minutes, Seconds). For each line, the name of the unit should be displayed followed by a colon, a single space, and the integer value you computed for that unit (e.g., Days: 10). Do not output any additional spacing or any blank lines between the cases.

Sample input
```
3
50
61
3600
```

Sample output
```
Days: 0
Hours: 0
Minutes: 0
Seconds: 50
Days: 0
Hours: 0
Minutes: 1
Seconds: 1
Days: 0
Hours: 1
Minutes: 0
Seconds: 0
```

Quarri

Contrary to common belief, the lost city of Atlantis never sank. Recent studies have debunked this and have proved that everything in the universe was merely a new form of particle, called the Quarri. The Quarri particle has a magical state that it can be either active or in hibernation based on the other Quarri particles around it. Based on the magical properties of Quarri, scientists believe that it can break all laws of physics and react in many different ways. As a result, it was able to transport Atlantis to another universe. In order to further prove Quarri's mysterious properties, a team of scientist have dispatched you to develop a program to determine the state of Quarri after a given number of time cycles.

Quarri follows a few simple rules. Every element of Quarri has two states, active and hibernation, that can change on any time cycle according to the state of the Quarri around it. For any active Quarri, if it has less than two active neighbors, it falls into hibernation. If it has four or more neighbors that are active, it is knocked into hibernation by the other Quarri. If the active Quarri has two or three active neighbors, it will remain active. For any hibernating Quarri, if it has exactly three active neighbors, it will be bumped into its active state; otherwise, it will remain hibernating.

Quarri clump together in groups of 400 (20x20), A Quarri will disregard all other Quarri outside its group. A Quarri will neighbor either 3, 5 or 8 Quarri based on its location. If it is on the edge of its group, it will merely have fewer neighbors than the Quarri in the center of the group. It will not neighbor with a Quarri in another group or across its own group.

Details of the input

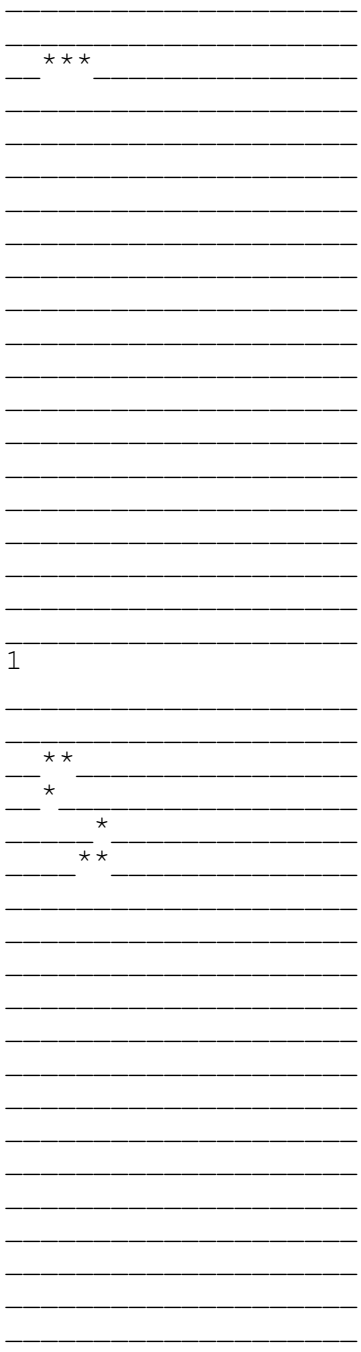
The first line of input will be a positive integer, *n*, which represents the number of test cases. For each test case there are 21 lines of input. The first line contains a positive integer, *t*, which represents how many time cycles should be simulated for the group of Quarri. Each of the next 20 lines contain twenty characters that specify the state of each Quarri with "_" representing hibernating and "*" representing active.

Details of the Output

For each test case, there are 21 lines of output. The first line is of the form “Case *i*:” where *i* is the current case (1..*n*). Each of the next 20 lines contains 20 characters that indicate the state of the Quarri after *t* iterations.

Sample input

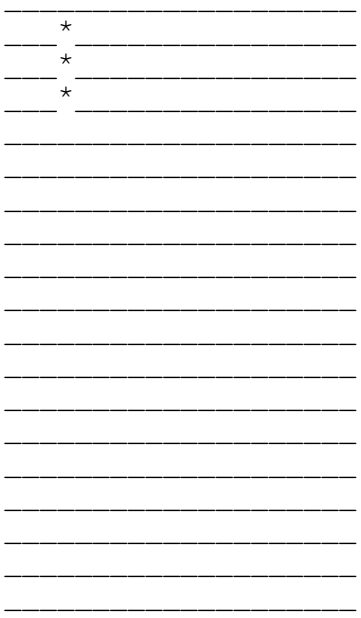
```
2
3
```



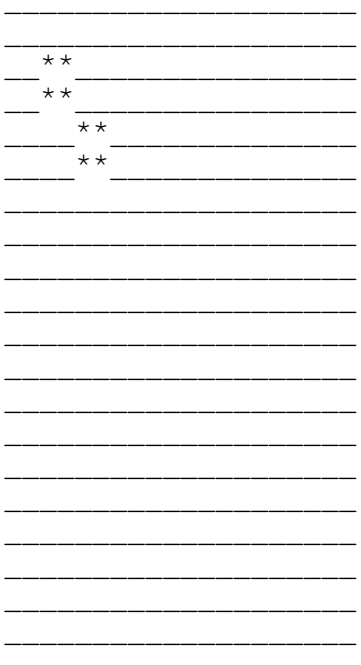
```
1
```


Sample output

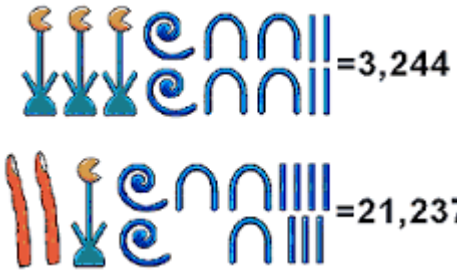
```
Case 1:
```



```
Case 2:
```



Egyptian Math



In Ancient Egypt, Horus was the god of mathematics. Horus founded the basis for all basic math for the duration of the Ancient Egyptian Empire. Before he taught the Egyptians the concept of mathematics, their world was a mess. The buildings were crooked and falling down. There was no way to count their animals so when one would wander away, no one noticed it simply disappeared forever. Finally, Horus had seen enough of this chaos and flew down from his perch up high to teach the people of Egypt the basic principles of mathematics. The people were now able to count, add, subtract, multiply, and divide. As he prepared to leave them, however, Horus noticed that the children were spending their time playing and not practicing their basic mathematics. When he approached one of the children and asked why they were not practicing mathematics, they said that it just wasn't fun. Outraged by this, Horus knew he could not leave without correcting the situation. Horus showed the children that practicing mathematics could be fun by demonstrating a little magic trick with numbers.

Horus told the children that he could transform any four digit number into either 6174 or 0000 based on a simple trick. The children were intrigued. Horus took a number, made two copies of it and then sorted the digits in each number, on in ascending order and one in descending order. He then subtracted the number whose digits were in ascending order from the one whose digits were in descending order. He then repeated this process on the result and continued to do so until the result was either 6174 or 0000.

For example, if the starting four-digit number is 9281, the trick goes like this:

9821 – 1289 = 8532 (not 6174 or 0000, so repeat)
8532 – 2358 = 6174 (magic!!)

He did it many times, letting the children pick any four digit number they wanted to as the starting point and it always worked. After the children learned this trick, they quickly taught this technique to everyone they could and began to practice mathematics often.

Your goal in this problem is to simulate Horus's trick. Write a program that converts any four digit number into either 6174 or 0000, counting how many times you had to repeat the sorting and subtracting steps to achieve the result.

Details of the input
The first line of input will be a positive integer, *n*, specifying the number of cases to follow. Each of the next *n* lines will contain a single, four-digit positive integer, *d*, that is to be converted.

Details of the output
For each of the *n* cases, display a single integer that indicates the number of repetitions of the sorting and subtracting steps that were repeated to get to either 6174 or 0000. Each time a number is sorted two ways and then subtracted counts as a single repetition.

Sample input
```
3
9281
2891
4118
```

Sample output
```
2
2
6
```

Birthday Surprise

Hera has decided that this year for Zeus's birthday, she will write him a special story but make a game of her gift by leaving out two nouns, two adjectives, and two verbs somewhere within the story. Then she will hide six scrolls that each contains one of those words all around Mount Olympus. Although Zeus likes scavenger hunts, he does not take kindly to word puzzles. So Hera has appointed you to write a program that will help Zeus piece together the story with the words that Zeus finds.

Details of the input

The first line contains a positive integer, *n*, indicating the number of cases. For each case, the first line of input is a positive integer, *w*, that indicates the number of total words in the story. The next six lines each contains a single string that specifies one of the missing words. The first two words are the missing nouns , *n1* and *n2*. The next two words are the missing verbs, *v1* and *v2.* The final two are the missing adjectives, *a1* and *a2*. The remainder of the input for each case is the paragraph containing w words, with each of the six missing words replaced by a placeholder for the word. The placeholder is delimited by square brackets ([ ]) that contain an indication of which word should be substituted for the placeholder. For example, the placeholder [n1] should be replaced the first noun, *n1*. There will be no other square brackets in the input paragraph. There may be punctuation adjacent to the placeholders and, if so, it should be preserved in the output.

Details of the output

For each case, the paragraph with the missing words substituted for the placeholders should be output. The entire paragraph is output on a single line with a single space between each output string. (Note: when viewing the output in the display window, it will wrap around and appear as though it is on multiple lines.)

Sample input

```
1
21
Zeus
Hera
running
collecting
loving
caring
Dear [n1], I hope you are enjoying [v1] around [v2] your birthday
present. You are a [a2] and [a1] husband. [n2]
```

Sample output

```
Dear Zeus, I hope you are enjoying running around collecting your
birthday present. You are a caring and loving husband. Hera
```

Where in the World Am I?

Eurybia, goddess of the mastery of the seas, always knew that Archias would become a renowned navigator. Even though he is still young, she takes every opportunity to *challenge him in ways that will help develop his mental and navigational skills. One day she came up with a creative way to give him directions and see how well he could decipher and follow them.*

Eurybia tells Archias to remember the number 0 and begins explaining the rules. If she says a number less than the number he is remembering, he is to turn 90^o to the left, forget the old number, and remember the new number. Similarly, a number greater than the number he is remembering means he should turn 90^o to the right, forget the old number, and remember the new number. Her silence, or repeating the number he's already remembering indicates he should continue going straight.

Archias is heading North when Eurybia starts giving him directions. Making left turns would cycle him through the directions in this order: North, West, South, East. Of course, making right turns would reverse the order: North, East, South, West.

Your program is to determine the final direction in which Archias is headed.

Details of the input
The first line of input will be and integer, *n*, representing the number of test cases to follow. Each test case appears on its own line. It begins with a positive integer, *k,* indicating the number of turns made during that test case, followed by the *k* integers representing the "directions" told to Archias by Eurybia.

Details of the output
For each case, the final direction of travel should be output.

Sample Input
```
2
6 1 2 3 4 -6 -1
4 15 12 12 14
```

Sample Output
```
North
East
```

The Riddle of the Sphinx

You are a sojourner on your way to the city of Thebes. As you walk along the path to the city you encounter the Sphinx, the guardian of the city. Before you may enter the city, you must answer his questions correctly. Being a guardian can be boring after a while, so instead of the usual riddle the Sphinx has decided to switch things up and presents you with a riddle comprised of 9's, 0's and A's. In order to pass his test and enter the city you must reply with a combination of numbers, letters, and symbols that correctly match the Sphinx's challenge.

Each character in the riddle has a specific rule that applies to it. The rules are as follows:

If the riddle character is a 9, the corresponding answer character must be a number.
If the riddle character is a 0, the corresponding answer character must be either a letter or number.
If the riddle character is an A, the corresponding answer character must be a letter.
Any symbol in the riddle that is either "-" or "+" remains the same character in the response.

Your reply to the sphinx must be the same length as his riddle.

Details of the input
The first line of the input contains a positive integer, *n*, that is the number of test cases. For each test case, there will be two lines of input. The first will be the sphinx's riddle and the second your intended response.

Details of the output
For each case, you will output one line. If your intended response does not agree with the sphynx's riddle then print

You have failed my simple question. You may not enter.

If the response solves the riddle then print

You have more wisdom than others before you. You may pass.

Sample Input

```
3
90A09
12AB3
90AA9
12A53
999-99-999A
266-32-532C
```

Sample Output

```
You have more wisdom than others before you. You may pass.
You have failed my simple question. You may not enter.
You have more wisdom than others before you. You may pass.
```

Salute the Sun

Throughout Japan, everyone adores the Great Sun Goddess Amaterasu. She bathes the land in light and provides protection to the people. Currently she is purging demons from the land - imps and chimeras and other creatures serving under the demon Orochi. However, Orochi has also polluted the land, making everything dark and restless. Gathering praise from her people, Amaterasu has been purifying the land and restoring nature to its former glory.

Vanquishing Orochi and his demons is no easy task. Amaterasu can cleanse the area but it is an exhausting task. Amaterasu wants to cleanse the worst area today to help her subjects. Japan's landscape can be viewed as a collection of triangles and each triangle has a danger rating. Amaterasu has enough energy to cleanse one triangular area. This area could be one individual triangle or a collection of smaller triangles that combine to create a larger triangle. In order to help out Amaterasu, you are to find the triangular area with the highest danger rating. The following examples illustrate this computation:

The highest danger rating is the combination of all four triangles (11)

The highest danger rating is the small triangular region in the center(5)

The highest danger rating is the boldfaced triangular region (21)

Details of the input

The first line of input will be a positive integer, *n*, representing the number of test cases. For each test case, there will be a positive integer, *t*, representing the number of small triangles ($0 < t <= 100$, *t* is a perfect square). Following there will be *t* integers that fill the triangles in a row-by-row fashion going from left to right within each row. For example, the input for the largest example above would be

2 -2 5 0 0 12 -1 7 -3 2 4 -8 3 0 1 -2

Details of the output

There is a single line of output for each case which contains the highest danger rating found for that case.

Sample Input

```
2
4
2 2 5 2
4
-2 -2 5 -2
```

Sample output

```
11
5
```