A: The aliens, they are stealing my IP

Description

You have learned from a reliable source on the internet that aliens are using gamma ray machines to steal your precious intellectual property. Luckily, after you type in your credit card number, the helpful web site explains how to build a gamma ray baffle. It turns out what you need to do is build a set of north facing walls, and then locate your chair behind as many walls as possible.

Input Format

The first line of each test case is the count of the number of walls; a count of zero indicates the end of input. The next lines describe the walls, one per line. The first number on the line is the y coordinate, the next two are the left and right x coordinates. You may assume that all coordinates are integers in the range -2^{31} to $2^{31}-1$ and no two segment endpoints have the same x or y coordinate. You may assume each test case has no more than 10000 walls.

Output Format

For each test case, print "depth k achieved in interval $[l, r]$" on a line by itself, where k is the maximum number of walls covering any point, and l and r are the left and right endpoints. If the maximum depth is achieved for multiple segments, report the rightmost one.

Sample Input

```
3
1 1 10
0 0 4
7 2 3
2
0 1 5
1 6 10
0
```

Sample Output

```
depth 3 achieved in interval [2,3]
depth 1 achieved in interval [6,10]
```

B: Skipping Strings

Description

Sue Spicious likes to communicate with her friends on social network sites, but she knows that these sites aren't very secure, and she also doesn't like the way that the sites' owners tend to assert ownership over all content. Her messages aren't likely to be of interest to many others, but she's not interested in having them data-mined by e-commerce companies.

To provide some simple protection against the data-miners, Sue encodes all of her messages through a scrambling procedure, which operates separately on each line of her message. Sue starts with the number 16, the age when she and her friends met, and pulls out every 16th character of the line, placing these characters in order at the beginning of her coded line of text. Then she halves the number, and pulls out every 8th character of the remaining characters of the line. She continues in this way for every 4th character and 2nd character, and then finally adds the remaining characters to the end of her coded line of text.

You are a friend of one of Sue's friends. While you don't know Sue, you find the posting of these encoded messages on your mutual friend's page rather irritating, and you want to read them. To do this, you need to write a program to reverse the process, extracting the original text.

Input Format

The input consists of lines of text, where each line contains from 1 to 80 characters, and is terminated by a newline `'\n'` character. The input ends with an empty line (newline character only). Each line is a scrambled line of Sue's message.

Output Format

The output consists of the unscrambled message, line by line. Do not print a blank line at the end.

Sample Input

```
f738c146ad0259be
tr eyaso tr eh slasf uodTiiny  oot .
s y.ltpew   d titmgh aeeuwu'p emrnsaseeunstmSeontoh oatse roc eh
ocnnresv npdeEeeyt .
```

Sample Output

```
0123456789abcdef
This is only a test for you to read.
Sue wouldn't post the important messages where you can see them.
Even encrypted ones.
```

C: Crossroads

Description

You have received a compelling business proposition from Mr. M. Ephistopheles. In exchange for something you are not really using, he offers you wealth and power. The tricky part is not the ethical dilemma, but where to meet. Mr. Ephistopheles has proposed several meeting points, each as the intersection of two roads. It turns out the guy is some kind of practical joker, because the first one you try, the roads don't actually intersect. You decide to write a computer program to test which of the meeting points (if any) actually exist.

Input Format

The first line contains the number of test cases. Each succeeding line is a test case, containing the coordinates for the endpoints of two segments of road (8 numbers in total). You may assume all coordinates are integers between -10000 and 10000 inclusive. There may be any amount of whitespace (spaces and tabs) before and after any input number.

Output Format

For each pair of segments that intersect, print "hit". For the other pairs, print "miss".

Sample Input

```
3
-1 -1 1 1 -1 1 1 -1
0 -100 0 100 0 0 1 1000
0 -100 0 100 1 0 1 1000
```

Sample Output

```
hit
hit
miss
```

D: How Many Islands?

Description

King Steven has initiated a new census scheme in which some people, data providers, are chosen from each square kilometer of the realm to provide data. We are not going to analyse very much of this data, only the locations of the providers, which is the first piece of data that each must provide. Each report starts with how far north or south of the throne room that the person providing the data lives, such as 10N for ten kilometers north of the throne room, or 7S for seven kilometers south of the throne room. The second piece of data is the number of kilometers east or west of the throne room, such as 3W for three kilometers west or 17E for seventeen kilometers east. The measurements are accurate to the nearest kilometer.

Two providers are on the same island if they are within one kilometer of each other, that is, the distances from the throne are the same in one direction, and differ by only one in the other direction. For example, 2N3E and 2N4E are on the same island, but 2N3E and 1N2E may not be on the same island. There may be more than one data provider at any given location. Of course if data providers x and y are on the same island, and data providers y and z are also on the same island, then x and z are also on the same island.

Note that the throne itself is at 0N0W, which is the same as 0N0E, 0S0E and 0S0W. The king himself is one of the data providers, and he is always the last data provider. No one else dares to provide data for the throne area.

Given the locations of all the providers, how many islands form the kingdom?

Input Format

The first line contains the number of test cases. Each test case consists of a list of locations in the form nXmY where n and m are nonnegative integers no greater than 99, X is either N or S, and Y is either E or W. The locations are separated by white space, blanks, tabs and/or new line characters. The last location for each test case is for the king at the throne room.

Output Format

For each test case, on one line give the number of the test case, and the number of islands in the kingdom for that test case.

Sample Input

```
2
1N1E 0N0E 1N1E 1N0W
1N0E 0N0W
```

Sample Output

```
1 2
2 1
```

E: Balloon Collecting

Description

The UNB Engineering Society has recently held a contest to develop the world's best balloon flotation compound. The team that won the competition has managed to develop a new gas compound that is able to float a conventional balloon well beyond that of helium gas. The team called their invention the "Super Hyper-Mega Global Anti-Gravitational Liquid Thermamine Compound", or the term SHMGALTC for short. Due to the excitement of winning the competition, the winning team released a large number of balloons into the air; however, these balloons are beginning to interfere with the airspace over Fredericton used for approaches to the airport.

UNB has been ordered to collect as many of the balloons as possible, and they need your help. UNB has only one large balloon capable of carrying human passengers, and they only have enough gas left to lift the balloon once, and then descend to the ground. All balloons will thus need to be caught as they descend. Thus UNB's collecting balloon must lift to a certain height and then begin its descent, capturing the balloons one at a time as they descend. The balloons must be captured in order; they can skip a balloon in the sequence, but once skipped it cannot be captured later. They need your help to figure out the maximum number of balloons that can be caught.

Input Format

The input contains on or more test cases, where each test case will consist of one or more non-negative integers representing the list of incoming balloons. These balloons must be processed in the order that they are given in. Each test case is terminated by a -1 (which does not represent one of the balloons), and the test cases are terminated by the end of file.

Note that there is no constraint on the number of balloons in the input.

Output Format

The ouput should consist of one line per test case, stating "The maximum number of balloons possible to catch is: X", where X is replaced by the maximum number of balloons that you should be able to collect. Each test case should be on its own line.

Sample Input

```
389 400 100 56 78 34 1 100 300 -1
1 2 3 3 2 1 -1
```

Sample Output

```
The maximum number of balloons possible to catch is: 5
The maximum number of balloons possible to catch is: 3
```

F: Count Luring's Programs

Description

Atan Luring invented an abstract computer in the 1930s. It had somewhat limited powers, however. The memory of the machine consisted of a stack of integers. There were only two operations: duplicate and add. Duplicate was denoted by D, which took the top element of the stack without popping it off the stack and pushed a second copy onto the stack, so the stack became one element larger. Add was denoted by A, which popped the top two elements off the stack and pushed the sum of these numbers back onto the stack, decreasing the number of numbers on the stack by one. The machine crashed if a program tried to pop a value from the stack when the stack was empty.

A program is called **legal** if it could start with a single element on the stack and also stop leaving a single element on the stack without causing the stack to ever become empty. How many legal programs are there of a given length? Note there is a single legal program of length 0, the empty program, and no legal programs of length 1. There are two legal programs of length 4, DDAA and DADA.

Input Format

There is no input.

Output Format

There are thirty lines of output, each containing a single integer value. On the first line of output print the number of legal programs of length 1, on the second the number of legal programs of length 2, etc. The last line contains the number of legal programs of length 30.

G: Fabrication Station Permutation

Description

You work for *Hot Bredz*, a manufacturer of custom toasters. Your manufacturing plant is highly automated, but unfortunately since the introduction of the jPod in 2011, no-one remembers how to program anymore. You are left with a fix set of *moves*, controlled by a wall of buttons. An example move is $(1, 2, 3)(5, 7)$ which means take the item at station 1 to station 2, the item at station 2 to station 3, and the item at station 3 back to station 1. In parallel, interchange the items at stations 5 and 7. Your mission is to write a program to determine given a set of moves, what stations are reachable from each other by applying 0 or more moves.

Input Format

The first line of the input is the number of stations. Zero stations denotes the end of input. The next line is the number of moves. The moves follow, one per line. Each move consists of a series of *cycles*. A cycle is just a comma separated list of numbers, in parentheses. Note that you should ignore whitespace, except newlines, in this problem. You may assume there are at most 16384 stations, and the that stations are numbered from 1.

Output Format

Output for each test case, the (sorted) sets of *orbits*, i.e. mutually reachable stations, using square brackets to mark sets. Your orbits should be ordered by their smallest element.

Sample Input

```
8
2
(3,5)(4,6)
(1,2,4,3)(5,6,8)
8
2
(3,5)(4,6)
(1,2,4,3)(5,6,7,8)
0
```

Sample Output

```
[ [ 1, 2, 3, 4, 5, 6, 8 ], [ 7 ] ]
[ [ 1, 2, 3, 4, 5, 6, 7, 8 ] ]
```