

Problem A: 4 Square

It's been known for over 200 years that every positive integer can be written in the form $x^2 + y^2 + z^2 + w^2$, for x, y, z, w non-negative integers. In fact, every positive integer can be written in the form $ax^2 + by^2 + cz^2 + dw^2$ for x, y, z, w non-negative integers, for certain non-negative integers a, b, c, d . One example, is, of course if $a = b = c = d = 1$. Another example is when $a = 1, b = 2, c = 5, d = 7$. You will be asked to find solutions for various of these problems.

Input

The first line of input will contain n (> 0); n test cases will follow. Each test case will consist of two lines. The first line will be of the form $a b c d$, where $0 \leq a \leq b \leq c \leq d$. The second line will be of the form $m n_1 n_2 \cdots n_m$. You may assume $0 < m \leq 20$ and all n_i are non-negative integers.

Output

Each n_i of each test case should generate one line of output, either giving the values of x, y, z, w so that $n_i = ax^2 + by^2 + cz^2 + dw^2$, or printing a message indicating this can't be done. If there is more than one possible solution, you should give the one with smallest x value. If there is a tie, then give the one of those with the smallest y value, and so on. The format of these lines should be exactly as in the examples below. A blank line should follow each input set.

Sample Input

```
3
1 1 1 1
3 7 15 20
1 2 5 5
2 8 15
1 3 7 8
4 1 2 4 20
```

Sample Output

```
1 1 1 2
1 1 2 3
0 0 2 4

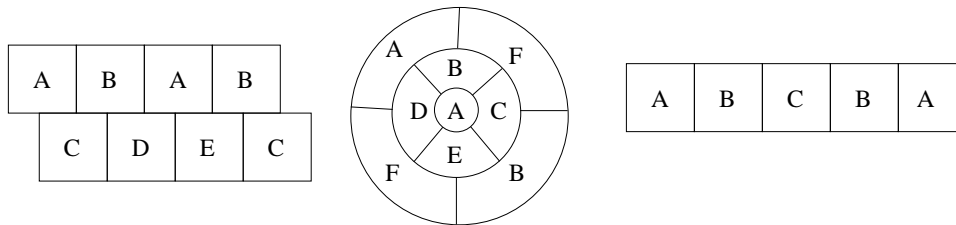
0 2 0 0
Impossible to do.

1 0 0 0
Impossible to do.
1 1 0 0
0 2 0 1
```

Problem B: Empire Mapping

It is well-known that every map drawn in a plane can be colored with no more than four colors, where “colored” means adjacent countries must have different colors. But what if the countries form empires? An empire is a collection of two or more countries. They may be adjacent or not. The question is to find the minimum number of colors needed to color a map where every country in the empire is colored the same and adjacent countries are colored differently (unless they are in the same empire, of course).

In the three maps below, empires are labeled with letters. The left two maps require 5 colors, while the right one requires only 2 colors



Input

There will be multiple test cases. The first line of a test case contains the integer m ($m \leq 20$), indicating the number of countries on the map. A value of $m = 0$ indicates end of input. The countries will be numbered 1 through m . Next follows the description of the empires. Each description will be of the form $n c_1 c_2 \dots c_n$, giving the n countries in an empire. A line containing 0 ends the empire description. The empires will be disjoint and not all countries must belong to an empire. There follows m lines. Line k gives the number countries adjacent to country k followed by a list of those countries adjacent to country k (all on one line); if country j is adjacent to country k , then k is also adjacent to j .

Output

For each test case, output one line of the form

Map m can be colored in b colors.

where b is the minimum number of colors required and m is the number of the map (starting at 1).

Sample Input

```
8
2 1 3
2 2 4
2 5 8
0
2 2 5
4 1 3 5 6
4 2 4 6 7
3 3 7 8
3 1 2 6
4 2 3 5 7
4 3 4 6 8
2 4 7
9
2 1 6
2 2 8
2 7 9
0
4 2 3 4 5
5 1 3 5 6 7
5 1 2 4 7 8
5 1 3 5 8 9
5 1 2 4 6 9
4 2 5 7 9
4 2 3 6 8
4 3 4 7 9
4 4 5 6 8
5
2 1 5
2 2 4
0
1 2
2 1 3
2 2 4
2 3 5
1 4
0
```

Sample Output

```
Map 1 can be colored in 5 colors.
Map 2 can be colored in 5 colors.
Map 3 can be colored in 2 colors.
```

Problem C: Schmoozin'

Bruce Wade Hughes is an up-and-coming rock-and-roll star. All he needs is a break. So, Bruce hops a greyhound to Nashville to schmooze with the recording industry A-list. He goes to parties and scans the room for the A-list people. A small camera hidden in his Elvis-style sunglasses relays the setup back to his friend (you) in a van outside. You identify the groups of A-list people and relay this information back to Bruce so he can go to those groups and not waste any time. (He's in a hurry to be a star!)

The party rooms are all rectangular grids and each type of person is marked with an upper case letter. If no one is occupying a place, it is marked with a period. A group of A-list people is a connected group if each A-list member is a horizontal or vertical neighbor of another A-list person. You are to determine how many A-list groups there are that have at least 3 members. (Bruce doesn't want to worry about any groups with fewer than 3 people.)

Input

There will be multiple test cases. The first line of the test case is the number of rows r in the grid followed by the number of columns c . (The end of input is if $r = 0$.) Each of these integers will be positive and no greater than 50. There then follows r lines each containing c characters, each an upper case letter or a period.

Output

For each test case, generate one line of the form

Party n has m A-list groups.

where n is the number of the test case (starting at 1) and m is the number A-list groups with at least 3 members.

Sample Input

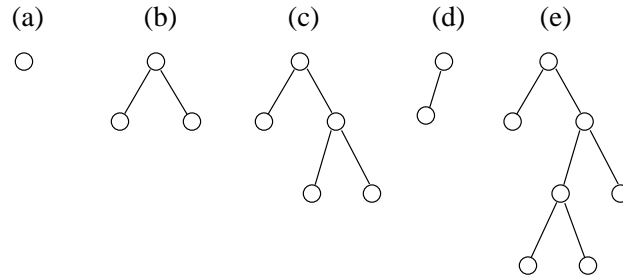
```
4 5
AAAABA
A..BA
A...A
ACA.A
5 5
AA.AA
ABA.A
BAA.A
CCC.A
BAAA.
0 4
```

Sample Output

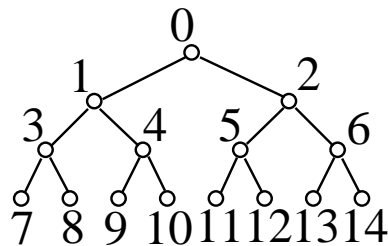
```
Party 1 has 2 A-list groups.
Party 2 has 4 A-list groups.
```

Problem D: Right Trees

A right-tree is a binary tree that has the following property: for every node in the tree, the right subtree has at least as many nodes as the left subtree. Thus of the following (a), (b) and (c) are right-trees while (d) and (e) are not.



The binary trees are encoded using a string of characters. The node positions in a binary tree can be numbered from top to bottom, left to right starting with Position 0 for the root node. The figure below shows the position numbers for a tree of height 4. The scheme continues on the next row (with nodes 15, 16, ...) for larger trees.



Input

Let s be the binary string that encodes a tree where s_i is the character at position i . The characters in s are numbered left to right starting at 0 so that s_0 is the left-most character in the string. If $s_i = 1$ then there is a node at position i in the tree, else $s_i = 0$ indicates no node. The string terminates with the last "1". All trees given as input here will have height no more than 8. Also assume the strings are syntactically correct in that if a "0" indicates no node at some point, there can never be any "1"s in positions that are nodes on that "0"s subtree.

The input consists of an integer n followed by n test cases, one per line. Each test case consists of a string encoded binary tree.

Output

For each problem instance, report one of the following messages corresponding to whether the tree is a right-tree or not:

Tree n is a right-tree.

Tree n is not a right-tree.

accordingly, where n is the test case number, starting at 1.

Sample Input

```
5
1
111
1110011
11
1110011000011
```

Sample Output

```
Tree 1 is a right-tree.
Tree 2 is a right-tree.
Tree 3 is a right-tree.
Tree 4 is not a right-tree.
Tree 5 is not a right-tree.
```

Problem E: Foreign Coins

Dell Gaightweigh is a traveling salesman who often visits obscure foreign countries. Many of the countries have coins of peculiar denominations. One day while sipping a cup of java in the country called Independent Bailiwick of Maka, Dell wondered how many ways there were to pay for the coffee using the coins of I.B.M. Help Dell solve his task.

Input

You will be given several data sets. Each data set contains a list of the coin denominations for a particular country. This is followed by a list of currency amounts that you must pay using only coins from this country. For each amount, compute the number of different ways to pay this amount using these coins — assuming you have an unlimited supply of these coins.

The first line of input is an integer p . This is followed by p data sets. Each data set starts with an integer n ($0 < n \leq 6$) indicating the number of different coin denominations. This is followed by a list of n coin denominations in increasing order. Afterwards is an integer m ($0 < m \leq 10$). This is followed by a list of m integer amounts that you must pay using the coins. No coin value will be greater than 100 and no amount will be greater than 200.

Output

For each of the m amounts, print a single integer indicating how many ways you can pay this amount. These values should be on one line separated by a single space.

Sample Input

```
2
2
1 2
4
0 2 3 5
4
2 3 6 11
4
0 5 8 12
```

Sample Output

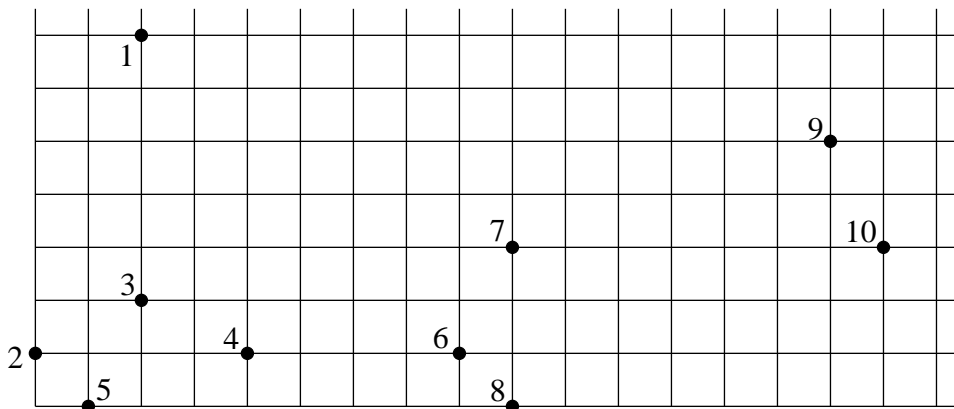
```
1 2 2 3
1 1 3 6
```

Problem F: Star Gazing

One night, while camping out in the wide open spaces, Big Ed was looking at the stars. Now Ed never bothered to learn his constellations, but decided that grouping the stars was a reasonable thing to do after all. But Big Ed was going to do it in a sensible manner. He decided on the following simple rules:

1. Every star is in the same constellation as its closest neighbor.
2. If the closest neighbor is not unique, then the star and all its closest neighbors are in the same constellation.
3. If A is in the same constellation as B , then B is in the same constellation as A .
4. If A is in the same constellation as B , and B is in the same constellation as C , then A is in the same constellation as C .

For example, if the picture of the sky looked like the following:



Then there are 3 constellations: $\{1, 2, 3, 4, 5\}$, $\{6, 7, 8\}$, $\{9, 10\}$.

Input

Input will consist of a sequence of sky descriptions. Each begins with a single integer n ($0 \leq n \leq 50$) on a line indicating the number of stars in the universe. The coordinates for the n stars follow as a pair x and y ($0 \leq x, y \leq 500$). A value of $n = 0$ indicates end of input.

Output

For each sky description, print a single line of the form

Sky s contains c constellations.

where s is the number of the sky description (starting at 1) and c is the number of constellations.

Sample Input

```
10
0 1
16 3
1 0
2 7
9 0
4 1
2 2
8 1
9 3
15 5
5
10 10
10 11
20 10
20 11
15 5
0
```

Sample Output

```
Sky 1 contains 3 constellations.
Sky 2 contains 1 constellations.
```