



2881 - Dice

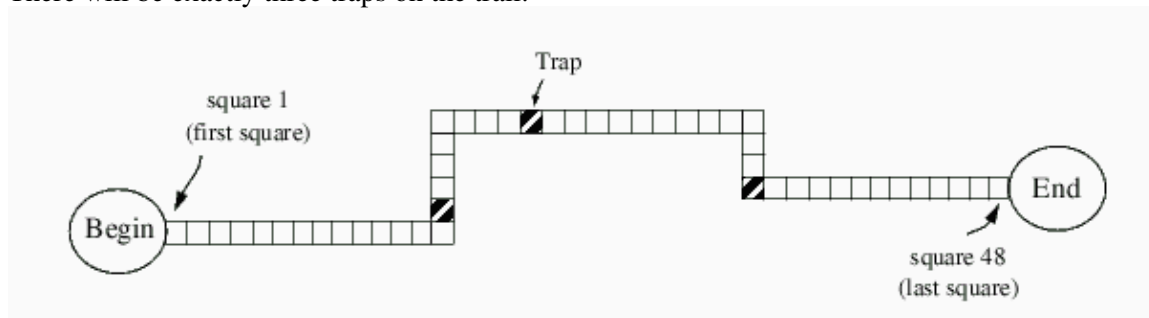
Latin America - South America - 2003/2004

A simple boardgame that generations of children have played consists of a board containing a trail of squares and a set of colored pieces. At the beginning of the game each player is assigned a piece; all pieces are initially positioned right before the first square of the trail.

The game proceeds in rounds. At each round, players roll a pair of dice, and move their pieces forward a number of squares equal to the rolled result. Players roll the dice always in the same order (player A, then player B, etc.) in the rounds.

Most of the squares on the board are plain squares, but some are "traps". If a player's piece falls on a trap square at the end of the player's move, the player misses the next round. That is, she/he does not roll the dice, and her/his piece stays one round without moving.

There will be exactly three traps on the trail.



The winner of the game is the player whose piece reaches the end of the trail first. The end of the trail is after the last square of the board. Consider, for example, the board in the figure above, which has squares numbered from 1 to 48. At the start, the pieces are positioned at the place marked 'Begin' in the figure, that is, before the square number 1. Therefore, if a player rolls a 7 (dice showing 2 and 5 for example), her/his piece is positioned at square number 7 at the end of the first round of the game. Furthermore, if a player's piece is positioned at square 41, the player needs a roll result of at least 8 to reach the end of the trail and win the game. Notice also that there will be no draw in the game.

You will be given the number of players, the number of squares in the trail, the location of the traps and a list of dice rolls results. You must write a program that determines the winner.

Input

Your program should process several test cases. The first line of a test case contains two integers P and S representing respectively the number of players and the number of squares in the trail ($1 \leq P \leq 10$ and $3 \leq S \leq 10000$). The second line describes the traps, represented by three distinct integers T_1 , T_2 and T_3 , denoting their positions in the trail ($1 \leq T_1, T_2, T_3 \leq S$). The third line contains a single integer N indicating the number of dice rolls in the test. Each of the following N lines contain two integers D_1 and D_2 ($1 \leq D_1, D_2 \leq 6$), representing the results of the dice rolls. The end of input is indicated by $P = S = 0$. The set of dice roll results in a test will be always the exact number necessary for a player to win the game.

A player is identified by a number from 1 to P . Players play in a round in sequential order from 1 to P . The input must be read from standard input.

Output

For each test case in the input, your program should output a single integer: the number representing the winner. The output must be written to standard output.

Sample Input

```
2 10
2 4 8
4
1 1
3 4
1 2
6 5
3 7
4 5 7
7
1 2
2 2
2 1
1 1
1 2
1 1
1 1
0 0
```

Sample Output

```
1
3
```

South America 2003-2004



3785 - Extraordinarily Tired Students

Asia - Xian - 2006/2007

When a student is too tired, he can't help sleeping in class, even if his favorite teacher is right here in front of him. Imagine you have a class of extraordinarily tired students, how long do you have to wait, before all the students are listening to you and won't sleep any more? In order to complete this task, you need to understand how students behave.

When a student is awoken, he struggles for a minutes listening to the teacher (after all, it's too bad to sleep all the time). After that, he counts the number of awoken and sleeping students (including himself). If there are strictly more sleeping students than awoken students, he sleeps for b minutes. Otherwise, he struggles for another a minutes, because he knew that when there is only very few sleeping students, there is a big chance for them to be punished! Note that a student counts the number of sleeping students only when he wants to sleep again.

Now that you understand each student could be described by two integers a and b , the length of awoken and sleeping period. If there are always more sleeping students, these two periods continue again and again. We combine an awoken period with a sleeping period after it, and call the combined period an awoken-sleeping period. For example, a student with $a = 1$ and $b = 4$ has an awoken-sleeping period of awoken-sleeping-sleeping-sleeping. In this problem, we need another parameter c ($1 \leq c \leq a + b$) to

describe a student's initial condition: the initial position in his awoken-sleeping period. The 1st and 2nd position of the period discussed above are awoken and sleeping, respectively.

Now we use a triple (a, b, c) to describe a student. Suppose there are three students $(2, 4, 1)$, $(1, 5, 2)$ and $(1, 4, 3)$, all the students will be awoken at time 18. The details are shown in the table below.

	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	😴	😴	😴	😴			😴	😴	😴	😴				
2	😴	😴	😴		😴	😴	😴	😴	😴		😴	😴	😴	😴
3	😴		😴	😴	😴	😴		😴	😴	😴	😴			

Table 1. An example

Write a program to calculate the first time when all the students are not sleeping.

Input

The input consists of several test cases. The first line of each case contains a single integer n ($1 \leq n \leq 10$), the number of students. This is followed by n lines, each describing a student. Each of these lines contains three integers a, b, c ($1 \leq a, b \leq 5$), described above. The last test case is followed by a single zero, which should not be processed.

Output

For each test case, print the case number and the first time all the students are awoken. If it'll never happen, output -1 .

Sample Input

```
3
2 4 1
1 5 2
1 4 3
3
1 2 1
1 2 2
1 2 3
0
```

Sample Output

```
Case 1: 18
Case 2: -1
```

Xian 2006-2007

Problemsetter: Rujia Liu



3930 - Drop the Triples

Latin America - South America - 2007/2008

The inhabitants of a small Caribbean island in the region known as Bermuda's Triangle love to spend their warm summer nights playing cards. As a tribute to the region where they live, all of their card games have some connection to triangles. One of the most popular games in the island is known as Triples, and has very simple rules.

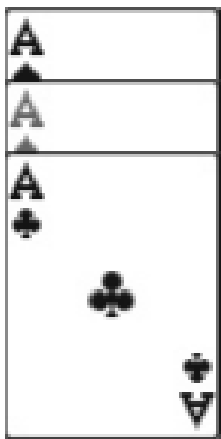
The game is played between two players, with a set of standard playing cards. Cards are distinguished only by their values, from 1 (Ace) to 13 (King). The cards are shuffled and placed as a pile in the center of the table, face down. This pile is called the stock. The two players take turns at playing. At each turn, a player

- draws the top card from the stock, adding it to her/his hand; and
- decides whether she/he wants to "drop some triples".

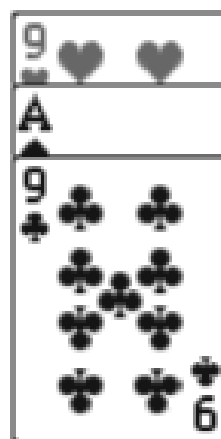
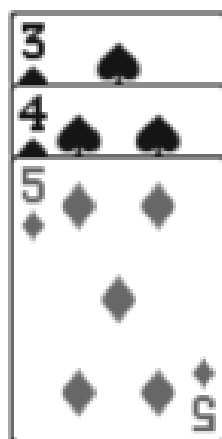
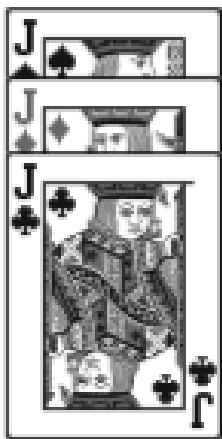
Dropping a triple consists of choosing three cards (a triple) from the hand and placing them on the table, face up. The dropped triples stay on the table until the end of the game. Only some sets of three cards form a valid triple. There are two types of valid triples:

- Perfect triples are made of three cards whose values represent the length of sides of an equilateral triangle;
- Common triples are made by three cards whose values represent the length of sides of any other (not equilateral) triangle.

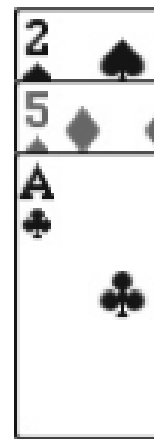
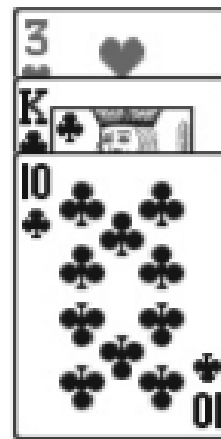
The figure below shows examples of perfect triples (a), common triples (b), and invalid triples (c).



(a)



(b)



(c)

Only valid triples can be dropped, but a player may drop any number of triples at a given turn. In particular, since players know the number of cards in the stock at every turn, a player may decide to drop all triples in her/his last turn. Some players, however, normally drop some triples during the game, to maintain as few cards in their hands as possible.

The game finishes when the stock is empty. The winner is the player that dropped the largest number of perfect triples. If both players dropped the same number of perfect triples, the winner is the player that dropped the largest number of common triples. If both players dropped the same number of perfect triples and the same number of common triples, the result is a tie.

Given the description of the cards in the stock, write a program that determines the winner of a game of Triples, considering both players play as best as possible.

Input

The input contains several test cases. The first line of a test case contains one integer N representing the number of cards in the stock ($6 \leq N \leq 10^4$). The next line contains N integers X_i , separated by single spaces, representing the cards in the stock ($1 \leq X_i \leq 13$, for $1 \leq i \leq N$). The cards are given in the order they are drawn by the players: the first card in the input (X_1) is the first card drawn, the second card in the input (X_2) is the second card drawn, and so on. Several cards with the same value may be present in the stock, and not necessarily all card values are present in the stock. The end of input is indicated by $N = 0$.

Output

For each test case your program must output a single line, containing '1' if the first player to play wins the game, '2' if the second player wins, or '0' if there is a tie.

Sample Input

```
7
5 6 5 6 5 6 8
12
13 13 13 13 13 13 1 3 2 9 3 9
12
1 2 1 2 1 2 3 1 4 2 5 3
0
```

Sample Output

```
0
2
1
```

South America 2007-2008



4205 - Clicking Checkboxes

Asia - Dhaka - 2008/2009

Creating a good and easy Graphical user interface is the heart of operating a computer in modern days. One important part of this interface is enabling users to give inputs using mouse and keyboard. Many different types of forms are there to take inputs from users. Such as (a) Text Box (b) Radio Button (c) Check Box (d) Combo Box etc. In this problem we will concentrate on check box.

Check boxes are used when a user needs to select more than one choice. Such a checkbox is shown in the figure on the left below. The problem with checkbox is that when someone has to choose n options or names he has to make n clicks, which can be very annoying when n is high. So the new software giant **toggle** is using a new type of checkbox, which shows good performance when number of boxes to be checked is high. In normal check box, a box is checked or unchecked if and only if it is clicked but in **toggle** checkbox when a box is clicked then all the boxes below it is also gets its status. That is if a box is originally not checked then someone clicks this box then this box and all the boxes below it will be checked. Similarly if a box is checked and someone clicks on it then it will become unchecked and so will all the boxes below it. So the selection made on the left can be made by only five clicks (The boxes that has to be clicked are marked with red or dark square on the right figure) in **toggle** checkbox (We are calling naming new type of checkbox as **toggle** checkbox).

<p>Choose your favourite problemsetters:</p> <ul style="list-style-type: none"><input checked="" type="checkbox"/> Rezaul Alam Chowdhury<input checked="" type="checkbox"/> Derek Kisman<input checked="" type="checkbox"/> Igor Naverniouk<input checked="" type="checkbox"/> Md. Kamruzzaman<input checked="" type="checkbox"/> Mohammad Mahmudur Rahman<input type="checkbox"/> Md. Sajjad Hossain<input type="checkbox"/> Gordon Cormack<input type="checkbox"/> Rujia Liu<input type="checkbox"/> Shahriar Manzoor<input checked="" type="checkbox"/> Syed Monowar Hossain<input checked="" type="checkbox"/> Petr<input checked="" type="checkbox"/> Per Austrin<input checked="" type="checkbox"/> Robert Roos<input checked="" type="checkbox"/> Don Chamberlin<input type="checkbox"/> Mathius Ruhl<input type="checkbox"/> Alex Chertov<input type="checkbox"/> Joachim Wulff<input type="checkbox"/> Jimmy Mardell<input checked="" type="checkbox"/> Abdullah al Mahmud Satej<input checked="" type="checkbox"/> Miguel Revilla	<p>Choose your favourite problemsetters:</p> <ul style="list-style-type: none"><input checked="" type="checkbox"/> Rezaul Alam Chowdhury<input checked="" type="checkbox"/> Derek Kisman<input checked="" type="checkbox"/> Igor Naverniouk<input checked="" type="checkbox"/> Md. Kamruzzaman<input checked="" type="checkbox"/> Mohammad Mahmudur Rahman<input checked="" type="checkbox"/> Md. Sajjad Hossain<input type="checkbox"/> Gordon Cormack<input type="checkbox"/> Rujia Liu<input type="checkbox"/> Shahriar Manzoor<input checked="" type="checkbox"/> Syed Monowar Hossain<input checked="" type="checkbox"/> Petr<input checked="" type="checkbox"/> Per Austrin<input checked="" type="checkbox"/> Robert Roos<input checked="" type="checkbox"/> Don Chamberlin<input checked="" type="checkbox"/> Mathius Ruhl<input type="checkbox"/> Alex Chertov<input type="checkbox"/> Joachim Wulff<input type="checkbox"/> Jimmy Mardell<input checked="" type="checkbox"/> Abdullah al Mahmud Satej<input checked="" type="checkbox"/> Miguel Revilla
---	---

Figure 1: 12 clicks are required in ordinary checkbox

Figure 2: 5 clicks are required in toggle checkbox.

The researchers of toggle have shown that **toggle** checkboxes show better performance when at least 50% boxes are to be checked. For example for the selection shown in the above picture we needed 12 clicks to check 12 boxes in normal checkbox but only five clicks were required for **toggle** checkboxes. So in this situation **toggle boxes** show better performance. In many other selection, **toggle** checkbox would also show strictly better performance. Your job is to count in how many selection patterns (where at least m boxes out of the n boxes in the checkbox are to be selected) toggle check box would show better performance and in how many selection patterns ordinary checkbox would show better performance.

Input

The input file contains 1000 lines of inputs. The description of each line is given below:

Each line contains two integers n ($0 < n < 64$) and m ($0 \leq m \leq n$). These two values indicate that you have to consider a checkbox with n boxes to check and consider only the input patterns where at least m of them are checked. Assume that initially all the boxes are unchecked.

Input is terminated by a line containing two zeroes. This line should be processed.

Output

For each line of input produce one line of output.

This line contains the serial of output followed by two integers TC and NC. Here TC is the total number of selection patterns where toggle checkbox requires less clicks and NC is the total number of selection patterns where ordinary or normal checkboxes require less clicks.

Sample Input

Output for Sample

5 3	Case 1: 9 3
10 3	Case 2: 370 419
0 0	

Problem setter: Shahriar Manzoor, Special Thanks: Derek Kisman, Md. Arifuzzaman Arif

Look at the illustration of first sample input on next page

<p>Favourite problemsetters?</p> <p><input checked="" type="checkbox"/> Rezaul Alam Chowdhury <input checked="" type="checkbox"/> Derek Kisman <input checked="" type="checkbox"/> Igor Naverniouk <input type="checkbox"/> Md. Kamruzzaman <input type="checkbox"/> Mohammad Mahmudur Rahman</p> <p>Normal Click = 3 Toggle Checkbox Toggle Click = 2 is better</p>	<p>Favourite problemsetters?</p> <p><input checked="" type="checkbox"/> Rezaul Alam Chowdhury <input checked="" type="checkbox"/> Derek Kisman <input type="checkbox"/> Igor Naverniouk <input checked="" type="checkbox"/> Md. Kamruzzaman <input type="checkbox"/> Mohammad Mahmudur Rahman</p> <p>Normal Click = 3 Normal Checkbox Toggle Click = 4 is better</p>	<p>Favourite problemsetters?</p> <p><input checked="" type="checkbox"/> Rezaul Alam Chowdhury <input checked="" type="checkbox"/> Derek Kisman <input type="checkbox"/> Igor Naverniouk <input type="checkbox"/> Md. Kamruzzaman <input checked="" type="checkbox"/> Mohammad Mahmudur Rahman</p> <p>Normal Click = 3 Toggle Click = 3</p>	<p>Favourite problemsetters?</p> <p><input checked="" type="checkbox"/> Rezaul Alam Chowdhury <input type="checkbox"/> Derek Kisman <input checked="" type="checkbox"/> Igor Naverniouk <input checked="" type="checkbox"/> Md. Kamruzzaman <input type="checkbox"/> Mohammad Mahmudur Rahman</p> <p>Normal Click = 3 Normal Checkbox Toggle Click = 4 is better</p>
<p>Favourite problemsetters?</p> <p><input checked="" type="checkbox"/> Rezaul Alam Chowdhury <input type="checkbox"/> Derek Kisman <input checked="" type="checkbox"/> Igor Naverniouk <input type="checkbox"/> Md. Kamruzzaman <input checked="" type="checkbox"/> Mohammad Mahmudur Rahman</p> <p>Normal Click = 3 Normal Checkbox Toggle Click = 5 is better</p>	<p>Favourite problemsetters?</p> <p><input checked="" type="checkbox"/> Rezaul Alam Chowdhury <input type="checkbox"/> Derek Kisman <input type="checkbox"/> Igor Naverniouk <input checked="" type="checkbox"/> Md. Kamruzzaman <input checked="" type="checkbox"/> Mohammad Mahmudur Rahman</p> <p>Normal Click = 3 Toggle Click = 3</p>	<p>Favourite problemsetters?</p> <p><input type="checkbox"/> Rezaul Alam Chowdhury <input checked="" type="checkbox"/> Derek Kisman <input checked="" type="checkbox"/> Igor Naverniouk <input checked="" type="checkbox"/> Md. Kamruzzaman <input type="checkbox"/> Mohammad Mahmudur Rahman</p> <p>Normal Click = 3 Toggle Checkbox Toggle Click = 2 is better</p>	<p>Favourite problemsetters?</p> <p><input type="checkbox"/> Rezaul Alam Chowdhury <input checked="" type="checkbox"/> Derek Kisman <input checked="" type="checkbox"/> Igor Naverniouk <input type="checkbox"/> Md. Kamruzzaman <input checked="" type="checkbox"/> Mohammad Mahmudur Rahman</p> <p>Normal Click = 3 Toggle Click = 3</p>
<p>Favourite problemsetters?</p> <p><input type="checkbox"/> Rezaul Alam Chowdhury <input checked="" type="checkbox"/> Derek Kisman <input type="checkbox"/> Igor Naverniouk <input checked="" type="checkbox"/> Md. Kamruzzaman <input checked="" type="checkbox"/> Mohammad Mahmudur Rahman</p> <p>Normal Click = 3 Toggle Click = 3</p>	<p>Favourite problemsetters?</p> <p><input type="checkbox"/> Rezaul Alam Chowdhury <input type="checkbox"/> Derek Kisman <input checked="" type="checkbox"/> Igor Naverniouk <input checked="" type="checkbox"/> Md. Kamruzzaman <input checked="" type="checkbox"/> Mohammad Mahmudur Rahman</p> <p>Normal Click = 3 Toggle Checkbox Toggle Click = 1 is better</p>	<p>Favourite problemsetters?</p> <p><input checked="" type="checkbox"/> Rezaul Alam Chowdhury <input checked="" type="checkbox"/> Derek Kisman <input checked="" type="checkbox"/> Igor Naverniouk <input checked="" type="checkbox"/> Md. Kamruzzaman <input type="checkbox"/> Mohammad Mahmudur Rahman</p> <p>Normal Click = 4 Toggle Checkbox Toggle Click = 2 is better</p>	<p>Favourite problemsetters?</p> <p><input checked="" type="checkbox"/> Rezaul Alam Chowdhury <input checked="" type="checkbox"/> Derek Kisman <input checked="" type="checkbox"/> Igor Naverniouk <input type="checkbox"/> Md. Kamruzzaman <input checked="" type="checkbox"/> Mohammad Mahmudur Rahman</p> <p>Normal Click = 4 Toggle Checkbox Toggle Click = 3 is better</p>
<p>Favourite problemsetters?</p> <p><input checked="" type="checkbox"/> Rezaul Alam Chowdhury <input checked="" type="checkbox"/> Derek Kisman <input type="checkbox"/> Igor Naverniouk <input checked="" type="checkbox"/> Md. Kamruzzaman <input checked="" type="checkbox"/> Mohammad Mahmudur Rahman</p> <p>Normal Click = 4 Toggle Checkbox Toggle Click = 3 is better</p>	<p>Favourite problemsetters?</p> <p><input checked="" type="checkbox"/> Rezaul Alam Chowdhury <input type="checkbox"/> Derek Kisman <input checked="" type="checkbox"/> Igor Naverniouk <input checked="" type="checkbox"/> Md. Kamruzzaman <input checked="" type="checkbox"/> Mohammad Mahmudur Rahman</p> <p>Normal Click = 4 Toggle Checkbox Toggle Click = 3 is better</p>	<p>Favourite problemsetters?</p> <p><input type="checkbox"/> Rezaul Alam Chowdhury <input checked="" type="checkbox"/> Derek Kisman <input checked="" type="checkbox"/> Igor Naverniouk <input checked="" type="checkbox"/> Md. Kamruzzaman <input checked="" type="checkbox"/> Mohammad Mahmudur Rahman</p> <p>Normal Click = 4 Toggle Checkbox Toggle Click = 1 is better</p>	<p>Favourite problemsetters?</p> <p><input checked="" type="checkbox"/> Rezaul Alam Chowdhury <input checked="" type="checkbox"/> Derek Kisman <input checked="" type="checkbox"/> Igor Naverniouk <input checked="" type="checkbox"/> Md. Kamruzzaman <input checked="" type="checkbox"/> Mohammad Mahmudur Rahman</p> <p>Normal Click = 5 Toggle Checkbox Toggle Click = 1 is better</p>

Illustration of first sample Input/Output:

When the total number of boxes that can be checked is five and at least three boxes has to be selected the total

possible patterns for checking is $\binom{5}{3} + \binom{5}{4} + \binom{5}{5} = 10 + 5 + 1 = 16$. Of these 16 patterns nine patterns are such that Toggle Checkbox shows better performance, three patterns are such that Normal Checkbox shows better performance while other four others are tied.

Dhaka 2008-2009



4225 - Prime Bases

North America - Rocky Mountain - 2008/2009

Given any integer base $b \geq 2$, it is well known that every positive integer n can be uniquely represented in base b . That is, we can write

$$n = a_0 + a_1 * b + a_2 * b * b + a_3 * b * b * b + \dots$$

where the coefficients $a_0, a_1, a_2, a_3, \dots$ are between 0 and $b-1$ (inclusive).

What is less well known is that if p_0, p_1, p_2, \dots are the first primes (starting from 2, 3, 5, ...), every positive integer n can be represented uniquely in the "mixed" bases as:

$$n = a_0 + a_1 * p_0 + a_2 * p_0 * p_1 + a_3 * p_0 * p_1 * p_2 + \dots$$

where each coefficient a_i is between 0 and p_i-1 (inclusive). Notice that, for example, a_3 is between 0 and p_3-1 , even though p_3 may not be needed explicitly to represent the integer n .

Given a positive integer n , you are asked to write n in the representation above. Do not use more primes than it is needed to represent n , and omit all terms in which the coefficient is 0.

Input

Each line of input consists of a single positive 32-bit signed integer. The end of input is indicated by a line containing the integer 0.

Output

For each integer, print the integer, followed by a space, an equal sign, and a space, followed by the mixed base representation of the integer in the format shown below. The terms should be separated by a space, a plus sign, and a space. The output for each integer should appear on its own line.

Sample Input

```
123
456
123456
0
```

Sample Output

```
123 = 1 + 1*2 + 4*2*3*5
456 = 1*2*3 + 1*2*3*5 + 2*2*3*5*7
123456 = 1*2*3 + 6*2*3*5 + 4*2*3*5*7 + 1*2*3*5*7*11 + 4*2*3*5*7*11*13
```

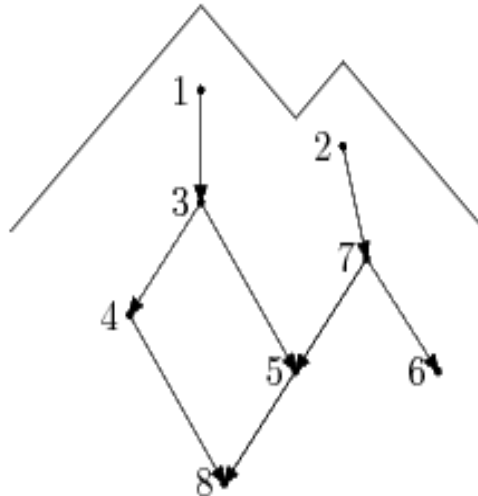
Rocky Mountain 2008-2009



4597 - Inspection

Europe - Northeastern Europe - 2009/2010

You are in charge of a team that inspects a new ski resort. A ski resort is situated on several mountains and consists of a number of slopes. Slopes are connected with each other, forking and joining. A map of the ski resort is represented as an acyclic directed graph. Nodes of the graph represent different points in ski resort and edges of the graph represent slopes between the points, with the direction of edges going downwards.



Your team has to inspect each slope of the ski resort. Ski lifts on this resort are not open yet, but you have a helicopter. In one flight the helicopter can drop one person into any point of the resort. From the drop off point the person can ski down the slopes, inspecting each slope as they ski. It is fine to inspect the same slope multiple times, but you have to minimize the usage of the helicopter. So, you have to figure out how to inspect all the slopes with the fewest number of helicopter flights.

Input

Input consists of several datasets. The first line of each dataset contains a single integer number n ($2 \leq n \leq 100$) -- the number of points in the ski resort. The following n lines of the input file describe each point of the ski resort numbered from 1 to n . Each line starts with a single integer number m_i ($0 \leq m_i < n$ for i from 1 to n) and is followed by m_i integer numbers a_{ij} separated by spaces. All a_{ij} are distinct for each i and each a_{ij} ($1 \leq a_{ij} \leq n$, $a_{ij} \neq i$) represents a slope going downwards from point i to point a_{ij} . Each point in the resort has at least one slope connected to it.

Output

On the first line of each dataset write a single integer number k -- the minimal number of helicopter flights that are needed to inspect all slopes. Then write k lines that describe inspection routes for each helicopter flight. Each route shall start with single integer number from 1 to n -- the number of the drop off point for the helicopter flight, followed by the numbers of points that will be visited during inspection in the corresponding order as the slopes are inspected going downwards. Numbers on a line shall be separated by spaces. You can

write routes in any order.

Sample Input

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

Sample Output

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

Northeastern Europe 2009-2010

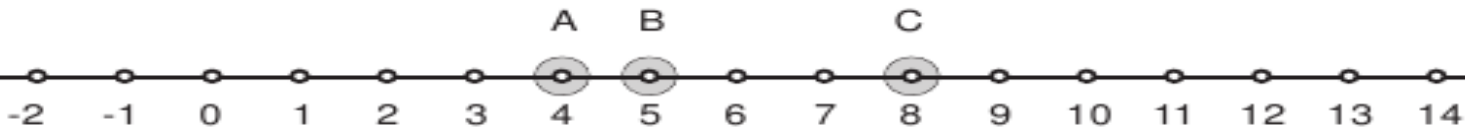


4813 - Ingenious Metro

Latin America - South America - 2010/2011

The King of Logonia will inaugurate soon a new and revolutionary metro, based on an invention of the Royal Engineers, which allows teletransportation.

The new metro consists of a very long tunnel with a station at each kilometer. There are also T teletransporters, which are located at some of the stations. In each station there is a keyboard with T keys, where each key corresponds to one teletransporter. The figure below illustrates a metro system with three teletransporters, located in stations marked A, B and C.



The metro works as follows. The user goes in a station (the start station) and presses the key corresponding to the teletransporter he wants to use. The user is then teletransported to the station which is at the same distance from the teletransporter as the start station, but on the opposite side relative to the teletransporter. More precisely, if the location of the start station is i and the user presses the key corresponding to the teletransporter located in position j , he will be taken to the station located at position $2 \times j - i$. For example, if the user is in station 6 and wants to go to station -2, he can use the teletransporter C (goes from 6 to 10) and then the teletransporter A (goes from 10 to -2).

The King, however, knows that it is possible that there is no sequence of teletransporters that will take the user from a given station X to a given station Y . To avoid that the users keep trying to go where they cannot go, he wants to make a program available in the Internet to help users. The King wants you to write a program which, given the position of each teletransporter, answers a series of queries. For each query the start and the destination stations are given, and your program must determine if it is possible for the user to go from start to destination.

Input

Each test case is given using several lines. The first line contains two integers T and Q indicating respectively the number of teletransporters ($1 \leq T \leq 10^5$) and the number of queries ($1 \leq Q \leq 10$). The second line contains T different integers t_i indicating the position of the teletransporters ($-10^7 \leq t_i \leq 10^7$). Each of the Q following lines describes a query and contains two distinct integers S and D indicating the position of the start and destination stations ($-10^7 \leq S, D \leq 10^7$).

The last test case is followed by a line containing two zeros.

Output

For each test case output a single line containing the answers to the Q queries, in the same order that the queries were given in the input. For each query you must output an uppercase 'Y' if it is possible to reach the destination station from the start station using the metro, or an uppercase 'N' otherwise.

Sample Input

```
1 1
-2
-6 2
5 2
10 20 30 40 50
10 15
20 40
5 3
0 5 -3 -8 4
-1 499
4 237
-1 -591
0 0
```

Sample Output

```
Y
N Y
Y N Y
```

South America 2010-2011



4950 - Selling Land

Europe - Northwestern - 2010/2011

G Selling Land

As you may know, the country of Absurdistan is full of abnormalities. For example, the whole country can be divided into unit squares that are either grass or swamp. Also, the country is famous for its incapable bureaucrats. If you want to buy a piece of land (called a parcel), you can only buy a rectangular area, because they cannot handle other shapes. The price of the parcel is determined by them and is proportional to the perimeter of the parcel, since the bureaucrats are unable to multiply integers and thus cannot calculate the area of the parcel.

Per owns a parcel in Absurdistan surrounded by swamp and he wants to sell it, possibly in parts, to some buyers. When he sells a rectangular part of his land, he is obliged to announce this to the local bureaucrats. They will first tell him the price he is supposed to sell it for. Then they will write down the name of the new owner and the coordinates of the south-east corner of the parcel being sold. If somebody else already owns a parcel with a south-east corner at the same spot, the bureaucrats will deny the change of ownership.

Per realizes that he can easily trick the system. He can sell overlapping areas, because bureaucrats only check whether the south-east corners are identical. However, nobody wants to buy a parcel containing swamp.

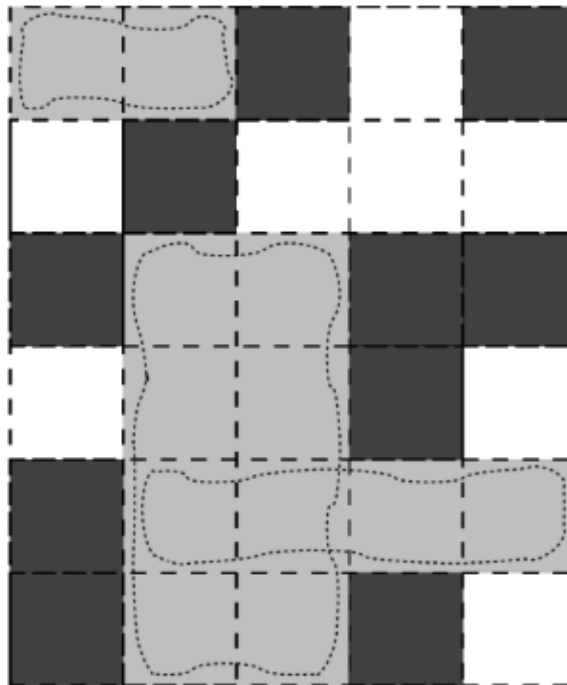


Figure 1: In this example, dark squares represent swamp. Per may, for example, sell three overlapping grey areas, with dimensions 2×1 , 2×4 and 4×1 respectively. The total perimeter is $6 + 12 + 10 = 28$. Note that he can get more money by selling even more land. This figure corresponds to the case in the sample input.

Now Per would like to know how many parcels of each perimeter he needs to sell in order to maximize his profit. Can you help him? You may assume that he can always find a buyer for each piece of land, as long as it doesn't contain any swamps. Also, Per is sure that no square within his parcel is owned by somebody else.

Input

On the first line a positive integer: the number of test cases, at most 100. After that per test case:

- One line with two integers n and m ($1 \leq n, m \leq 1\,000$): the dimensions of Per's parcel.
- n lines, each with m characters. Each character is either '#' or '.'. The j -th character on the i -th line is a '#' if position (i, j) is a swamp, and '.' if it is grass. The north-west corner of Per's parcel has coordinates $(1, 1)$, and the south-east corner has coordinates (n, m) .

Output

Per test case:

- Zero or more lines containing a complete list of how many parcels of each perimeter Per needs to sell in order to maximize his profit. More specifically, if Per should sell p_i parcels of perimeter i in the optimal solution, output a single line " $p_i \times i$ ". The lines should be sorted in increasing order of i . No two lines should have the same value of i , and you should not output lines with $p_i=0$.

Sample in- and output

Input	Output
1	6 x 4
6 5	5 x 6
..##	5 x 8
.#...	3 x 10
#...##	1 x 12
...#.	
#....	
#...#.	

Northwestern 2010-2011



5029 - Encoded Barcodes

Asia - Tianjin - 2010/2011

All the big malls need a powerful system for the products retrieval. Now you are employed design a sub-system: reading the barcodes and return the matching products.

A barcode is an optical machine-readable representation of data, which shows certain data on certain products. A barcode consists of a series of bars with different widths. In our system, the barcodes have been scanned and the widths have been recorded. Every consecutive eight bars are considered as representing the ASCII code of a character, each bar for each bit. Ideally, there should be only two kinds of widths in the eight bars, and the width of the wider bar is twice of the narrower. The wider bar indicates 1, while the narrower indicates 0. However, due to the inaccuracy of printing and scanning, there will be an error of at most 5%. That is, if the pretended exact width is x , you may get a value in the range $[0.95x, 1.05x]$.

For example, the width sequence "10.0 20.0 10.0 10.0 10.0 10.0 10.0 20.0" is a valid barcode of our system, and it means $(01000001)_2$, which is $(65)_{10}$ and the corresponding character is "A". Note that "10.5 20.1 10.1 10.2 9.9 9.7 10.0 19.9" is also a valid barcode representing the same letter.

You are given the names of all the products and many queries. Every name contains lower-case letters only, and the length is no more than 30. The queries are represented as barcodes. For each query, you should decode it to a string S , and report the amount of products whose prefix is S . For the output may be very large, you only need to output the sum of all the queries for each case.

Input

There are several test cases in the input. The first line of each case contains two integers N and M ($1 \leq N \leq 10000$, $1 \leq M \leq 2000$), indicating the number of products and queries. Then N lines follow, indicating the names of the products. Note that the names may be duplicated. Then M query blocks follow. The first line of each query block is an integer K ($0 < K \leq 30$) indicating the length of the query, then K lines follow, each line contains 8 positive float numbers, indicating the barcode for each character.

You can assume that the barcodes are always valid, and always represent lower-case letters.

Output

Output one line for each test case, indicating the sum of all the query results as described above.

Explanation for the sample:

There is only one test case. The first query is "a", and the answer is 3. The second query is "ap", and the answer is 2. The third query is "c", and the answer is 0. So the total sum is $3+2+0 = 5$.

Sample Input

```
4 3
apple
apple
```

avatar
book
1
1 2 2 1 1 1 1 2
2
1 2 2 1 1 1 1 2
10.1 20.1 19.9 20.0 10.2 9.8 9.9 10.0
1
1 2 2 1 1 1 2 2

Sample Output

5

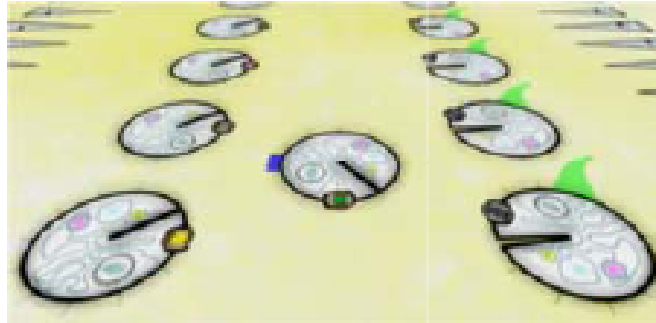
Tianjin 2010-2011



5055 - Indisputable Right

Asia - Tehran - 2010/2011

Flatland is a 2-dimensional world in which some creatures have an odd miserable life. Despite the many handicaps, the poor creatures have recently achieved the wireless technology in Flatland. So, the priest circles (governors of Flatland who kill the apostles of 3D as soon as they see one) created the national motto ``Wireless technology is our indisputable right!" and organized a wireless spreading committee for connecting all important points of the Flatland by wireless antennas.

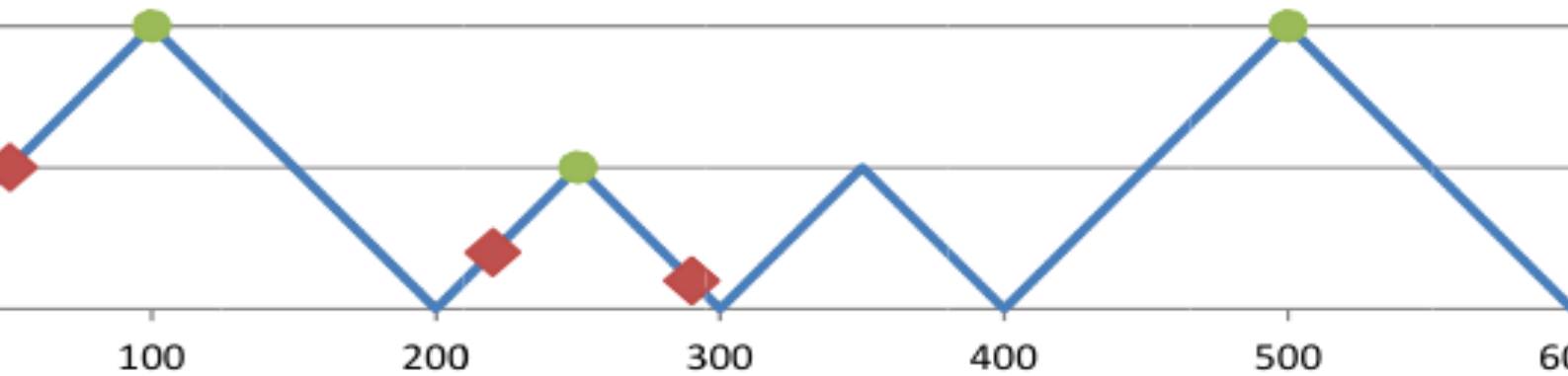


Everything went well until the committee reached the mountainous region of Flatland. This region can be modeled with a horizontal base line covered by a contiguous serie of mountain. Each mountain is a right-angled isosceles triangle with its base on the base line and its right angle as the peak. Currently, there are only two types of mountains in this region: those with a height of 50, and those with a height of 100. You can see a sample of the mountainous region in the figure below.

The wireless spreading comitte first installed their antennas on the points they had already planned for the mountainous region. But in the end, they found out that a pair of antennas could directly communicate only if the line segment connecting the two antennas had no crossing intersection with any obstacles -- specifically the mountains here. Note that two antennas can communicate if only the border of obstacles is on their connecting line segment.

Since removing a wireless antenna in Flatland is nowadays as a deadly sin as ``chromatization", the only way to connect the set of already installed antennas is to install some extra antennas on the appropriate points of the mountains! Wireless antennas can generally relay the messages and connect to ones which cannot communicate directly. In order to reduce the expenses of installing extra antennas, the wireless spreading committee has hired you to find the minimum number of antennas needed to install.

For example, consider the 4 diamond-shaped points in the figure below as the place of the wireless antennas already installed on the mountains. Now the whole set becomes connected if you instal 3 new antennas in the places specified by the circle-shaped points.



Input

There are multiple test cases in the input. Each test case starts with a line containing the integer n ($1 \leq n \leq 500$), the number of mountains, followed by the integer m ($1 \leq m \leq 10000$), the number of already installed antennas. The second line contains n integers h_1, h_2, \dots, h_n ($h_i \in \{50, 100\}$), respectively representing the height of the mountains on the base line from left to right. The third (last) line of the test case contains m integers specifying the position of the antennas already installed on the mountains. The position of each antenna is specified by its X coordinate, assuming the left-most point of the left-most mountain being the origin of the coordinate system. You yourself can calculate the Y coordinate of antenna as it is put on the boundaries of a mountain.

The input terminates with a line of the form "0 0" which should not be processed as a test case.

Note: The Sample Input test case is related to the configuration illustrated in the above figure.

Output

For each test case, write a single line containing the minimum number of extra wireless antennas needed for making the given set of antennas connected.

Sample Input

```
5 4
100 50 50 100 50
50 220 290 625
0 0
```

Sample Output

```
3
```

Tehran 2010-2011