Problem A
Sum of Consecutive Prime Numbers

Some positive integers can be represented by a sum of one or more consecutive prime numbers. How many such representations does a given positive integer have? For example, the integer 53 has two representations $5 + 7 + 11 + 13 + 17$ and 53. The integer 41 has three representations $2+3+5+7+11+13$, $11+13+17$, and 41. The integer 3 has only one representation, which is 3. The integer 20 has no such representations. Note that summands must be consecutive prime numbers, so neither $7 + 13$ nor $3 + 5 + 5 + 7$ is a valid representation for the integer 20.

Your mission is to write a program that reports the number of representations for the given positive integer.

Input

The input is a sequence of positive integers each in a separate line. The integers are between 2 and 10 000, inclusive. The end of the input is indicated by a zero.

Output

The output should be composed of lines each corresponding to an input line except the last zero. An output line includes the number of representations for the input integer as the sum of one or more consecutive prime numbers. No other characters should be inserted in the output.

Sample Input

```
2
3
17
41
20
666
12
53
0
```

Output for the Sample Input

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

Problem B
Atomic Car Race

In the year 2020, a race of atomically energized cars will be held. Unlike today's car races, fueling is not a concern of racing teams. Cars can run throughout the course without any refueling. Instead, the critical factor is tire (tyre). Teams should carefully plan where to change tires of their cars.

The race is a road race having n checkpoints in the course. Their distances from the start are a_1, a_2, $\cdots$, and a_n (in kilometers). The n-th checkpoint is the goal. At the i-th checkpoint ($i < n$), tires of a car can be changed. Of course, a team can choose whether to change or not to change tires at each checkpoint. It takes b seconds to change tires (including overhead for braking and accelerating). There is no time loss at a checkpoint if a team chooses not to change tires.

A car cannot run fast for a while after a tire change, because the temperature of tires is lower than the designed optimum. After running long without any tire changes, on the other hand, a car cannot run fast because worn tires cannot grip the road surface well. The time to run an interval of one kilometer from x to $x + 1$ is given by the following expression (in seconds). Here x is a nonnegative integer denoting the distance (in kilometers) from the latest checkpoint where tires are changed (or the start). r, v, e and f are given constants.

$$
\begin{array}{ll}
1/(v - e \times (x - r)) & \text{(if } x \geq r) \\
1/(v - f \times (r - x)) & \text{(if } x < r)
\end{array}
$$

Your mission is to write a program to determine the best strategy of tire changes which minimizes the total time to the goal.

Input

The input consists of multiple datasets each corresponding to a race situation. The format of a dataset is as follows.

n
a_1 a_2 $\ldots$ a_n
b
r v e f

The meaning of each of the input items is given in the problem statement. If an input line contains two or more input items, they are separated by a space.

n is a positive integer not exceeding 100. Each of $a_1, a_2, \cdots$, and a_n is a positive integer satisfying $0 < a_1 < a_2 < \ldots < a_n \leq 10000$. b is a positive decimal fraction not exceeding 100.0. r is a nonnegative integer satisfying $0 \leq r \leq a_n - 1$. Each of v, e and f is a positive decimal fraction. You can assume that $v - e \times (a_n - 1 - r) \geq 0.01$ and $v - f \times r \geq 0.01$.

The end of the input is indicated by a line with a single zero.

Output

For each dataset in the input, one line containing a decimal fraction should be output. The decimal fraction should give the elapsed time at the goal (in seconds) when the best strategy is taken. An output line should not contain extra characters such as spaces.

The answer should not have an error greater than 0.001. You may output any number of digits after the decimal point, provided that the above accuracy condition is satisfied.

Sample Input

```
2
2 3
1.0
1 1.0 0.1 0.3
5
5 10 15 20 25
0.15
1 1.0 0.04 0.5
10
1783 3640 3991 4623 5465 5481 6369 6533 6865 8425
4.172
72 59.4705 0.0052834 0.0611224
0
```

Output for the Sample Input

```
3.5397
31.9249
168.6682
```

Problem C. Gary

Gary is a Beer Robot at the Rocket Engine Bar orbiting Alpha Herculis II. His job is to accept bank notes and coins from the customers, and offer beer in return. Gary has one slot for bank notes and another slot for coins. He has no holes to give back change.

Gary is an old and experienced bar-robot. Pirates from nearby planetary systems like to listen to his stories about the good old times when a simple railgun was the heaviest spacecraft armament in existence, and boarding was a fast and effective way to capture a tradeship.

However, being an old worn-out robot has certain drawbacks. Perhaps the most nasty one is that he can accept only one type of coins and one type of bank notes. And Gary is pretty much aware that, for example, one cannot pay exactly 5 credits using only 4-credit bank notes and 3-credit coins.

Fortunately, Gary's self-repair equipment allows him to perform some fine tuning and re-pick the denominations of the coins and bank notes his slots would accept. Now he wants to cleverly choose the numbers a and b which would be the denominations of these bank notes and coins. More specifically, for certain a and b he wants to estimate how good or bad this choice would be. And for that, he needs to know how many positive integer amounts of credits q could not be paid using only a-credit bank notes and b-credit coins, and the greatest integer amount of credits l that could not be paid.

Input

On the first line of the input file there are two integer numbers a and b separated by a single space ($1 \leqslant a,\, b \leqslant 10^9$).

Output

You should print two numbers, q and l, on the first line of the output file. Numbers should be separated by a single space. If q or l is infinite, print "`infinity`" instead of the respective number.

Examples

standard input	standard output
3 4	3 5
2 2	infinity infinity

Problem D
Network Mess

Gilbert is the network admin of Ginkgo company. His boss is mad about the messy network cables on the floor. He finally walked up to Gilbert and asked the lazy network admin to illustrate how computers and switches are connected. Since he is a programmer, he is very reluctant to move throughout the office and examine cables and switches with his eyes. He instead opted to get this job done by measurement and a little bit of mathematical thinking, sitting down in front of his computer all the time. Your job is to help him by writing a program to reconstruct the network topology from measurements.

There are a known number of computers and an unknown number of switches. Each computer is connected to one of the switches via a cable and to nothing else. Specifically, a computer is never connected to another computer directly, or never connected to two or more switches. Switches are connected via cables to form a tree (a connected undirected graph with no cycles). No switches are 'useless.' In other words, each switch is on the path between at least one pair of computers.

All in all, computers and switches together form a tree whose leaves are computers and whose internal nodes switches (See Figure 9).

Gilbert measures the distances between *all pairs of computers*. The distance between two computers is simply the number of switches on the path between the two, plus one. Or equivalently, it is the number of cables used to connect them. You may wonder how Gilbert can actually obtain these distances solely based on measurement. Well, he can do so by a very sophisticated statistical processing technique he invented. Please do not ask the details.

You are therefore given a matrix describing distances between leaves of a tree. Your job is to construct the tree from it.

Input

The input is a series of distance matrices, followed by a line consisting of a single '0'. Each distance matrix is formatted as follows.

$$
\begin{array}{llll}
N & & & \\
a_{11} & a_{12} & \cdots & a_{1N} \\
a_{21} & a_{22} & \cdots & a_{2N} \\
\vdots & \vdots & \ddots & \vdots \\
a_{N1} & a_{N2} & \cdots & a_{NN}
\end{array}
$$

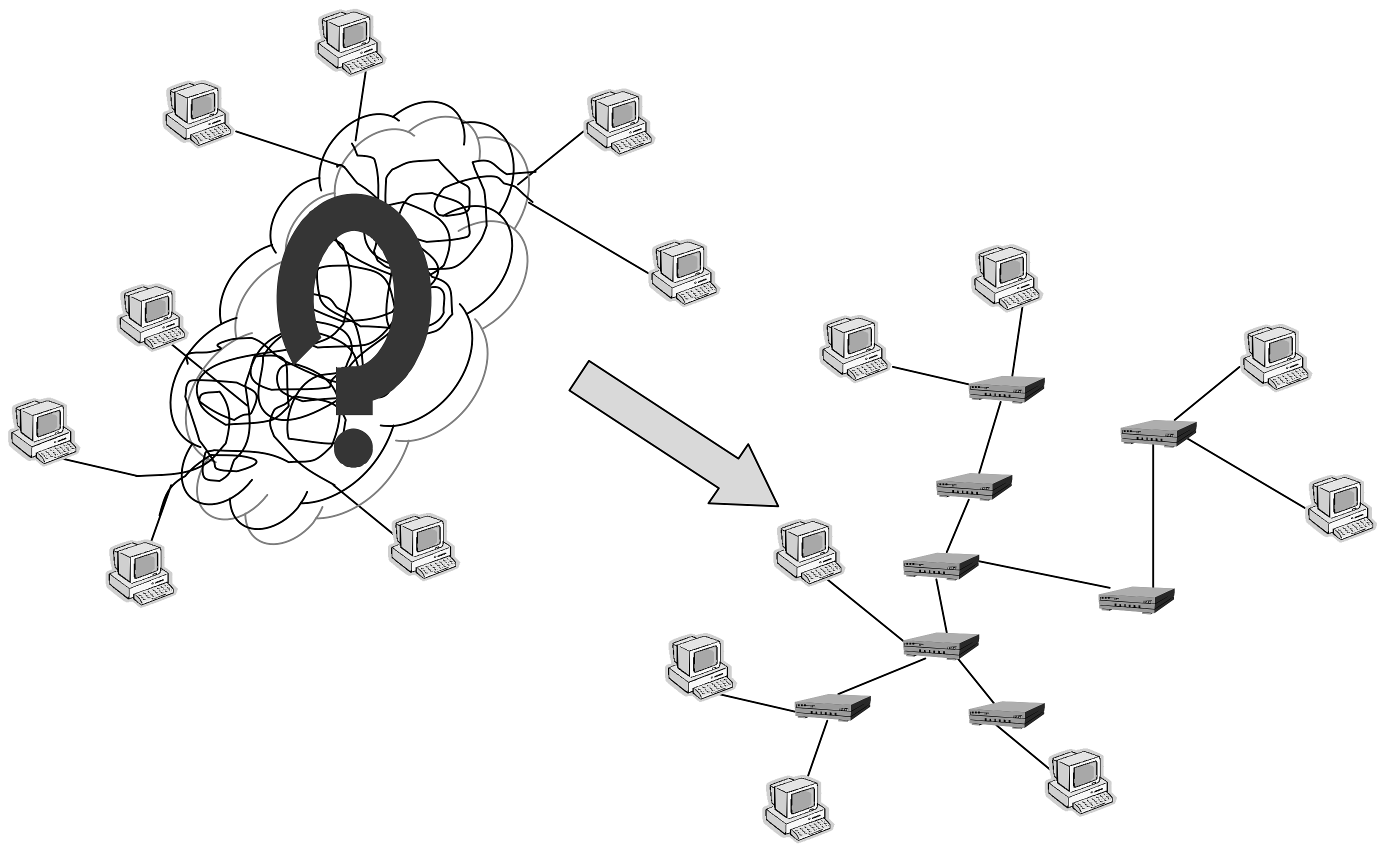


Figure 9: Computers and Switches

N is the size, i.e. the number of rows and the number of columns, of the matrix. a_{ij} gives the distance between the i-th leaf node (computer) and the j-th. You may assume $2 \leq N \leq 50$ and the matrix is symmetric whose diagonal elements are all zeros. That is, $a_{ii} = 0$ and $a_{ij} = a_{ji}$ for each i and j. Each non-diagonal element a_{ij} ($i \neq j$) satisfies $2 \leq a_{ij} \leq 30$. You may assume there is always a solution. That is, there is a tree having the given distances between leaf nodes.

Output

For each distance matrix, find a tree having the given distances between leaf nodes. Then output the degree of each internal node (i.e. the number of cables adjoining each switch), all in a single line and in ascending order. Numbers in a line should be separated by a single space. A line should not contain any other characters, including trailing spaces.

Sample Input

```
4
  0  2  2  2
  2  0  2  2
  2  2  0  2
  2  2  2  0
4
  0  2  4  4
  2  0  4  4
  4  4  0  2
  4  4  2  0
2
  0 12
 12  0
0
```

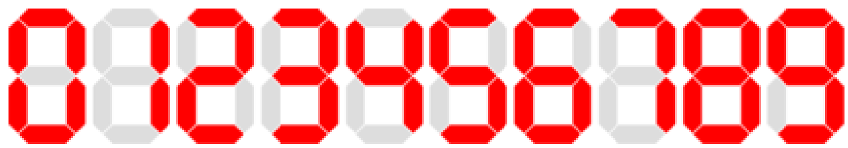
Output for the Sample Input

```
4
2 3 3
2 2 2 2 2 2 2 2 2 2 2
```

Problem E Seven-segment Display

Tom is a boy whose dream is to become a scientist, he invented a lot in his spare time. He came up with a great idea several days ago: to make a stopwatch by himself! So he bought a seven-segment display immediately.

The seven elements of the display are all light-emitting diodes (LEDs) and can be lit in different combinations to represent the arabic numerals like:

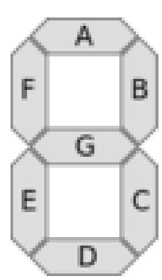


However, just when he finished the programs and tried to test the stopwatch, some of the LEDs turned out to be broken! Some of the segments can never be lit while others worked fine. So the display kept on producing some ambiguous states all the time...

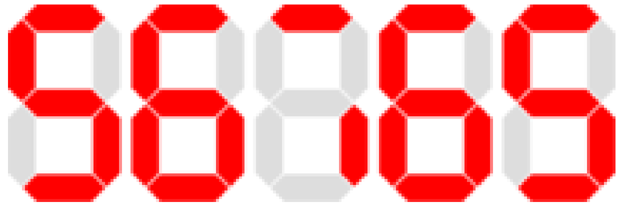
Tom has recorded a continuous sequence of states which were produced by the display and is curious about whether it is possible to understand what this display was doing. He thinks the first step is to determine the state which the display will show **next**, could you help him?

Please note that the display works well despite those broken segments, which means that the display will keep on counting down **cyclically** starting from a certain number (can be any one of 0-9 since we don't know where this record starts from). 'Cyclically' here means that each time when the display reaches 0, it will keep on counting down starting from 9 again.

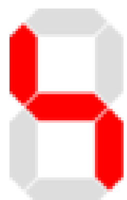
For convenience, we refer the seven segments of the display by the letters A to G as the picture below:



For example, if the record of states is like:



It's not that hard to figure out that ONLY segment B is broken and the sequence of states the display is trying to produce is simply "`9 -> 8 -> 7 -> 6 -> 5`". Then the next number should be 4, but considering of the brokenness of segment B, the next state should be:



Input

The first line of the input gives the number of test cases, t $(1 \le t \le 2000)$. Each test case is a line containing an integer n $(1 \le n \le 100)$ which is the number of states Tom recorded and a list of the n states separated by spaces. Each state is encoded into a 7-character string represent the display of segment A-G, from the left to the right. Characters in the string can either be '`1`' or '`0`', denoting the corresponding segment is on or off, respectively.

Output

For each test case, if the input unambiguously determines the next state of the display print the next state (in the same format as the input). Otherwise, you should print "**`ERROR!`**" (without quotes).

Examples

standard input	standard output
4 1 1111111 2 0000000 0001010 3 0100000 0000111 0000011 5 1011011 1011111 1010000 1011111 1011011	1110000 ERROR! 0100011 0010011

Problem F K-th sequence

Consider the set S of all non-negative integer sequences of length n. Sort all sequences in S by sum of their components (sequences with the same sums are ordered lexicographically), thus obtaining some sorted list of sequences T. Your task is to find k-th sequence in T (numbering is 0-based).

Input

Input consists of one or more test cases. Each case is given on a separate line and consists of two numbers, n and k ($1 \leqslant n \leqslant 1000$, $0 \leqslant k \leqslant 10^{18}$). Input is terminated by a line containing two zeroes; this line should not be processed.

Output

For each test case, output the corresponding sequence. Adhere to the output format shown below as close as possible.

Example

standard input	standard output
2 2 3 6 0 0	Case 1: 1 0 Case 2: 0 2 0