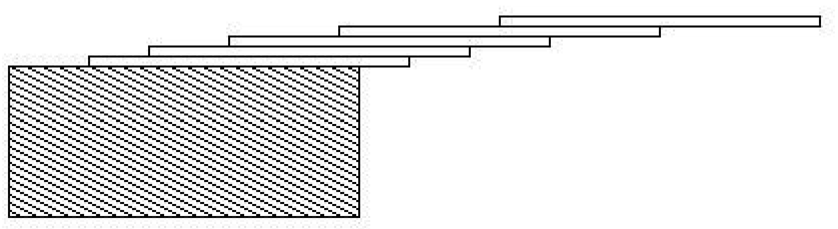
题目 1

Description

How far can you make a stack of cards overhang a table? If you have one card, you can create a maximum overhang of half a card length. (We're assuming that the cards must be perpendicular to the table.) With two cards you can make the top card overhang the bottom one by half a card length, and the bottom one overhang the table by a third of a card length, for a total maximum overhang of 1/2 + 1/3 = 5/6 card lengths. In general you can make *n* cards overhang by 1/2 + 1/3 + 1/4 + ... + 1/(*n* + 1) card lengths, where the top card overhangs the second by 1/2, the second overhangs tha third by 1/3, the third overhangs the fourth by 1/4, etc., and the bottom card overhangs the table by 1/(*n* + 1). This is illustrated in the figure below.



Input

The input consists of one or more test cases, followed by a line containing the number 0.00 that signals the end of the input. Each test case is a single line containing a positive floating-point number c whose value is at least 0.01 and at most 5.20; c will contain exactly three digits.

Output

For each test case, output the minimum number of cards necessary to achieve an overhang of at least c card lengths. Use the exact output format shown in the examples.

Sample Input

```
1.00
3.71
0.04
5.19
0.00
```

Sample Output

```
3 card(s)
61 card(s)
1 card(s)
273 card(s)
```

题目 2

Description

The quality of pseudo random-number generators used in some computations, especially simulation, is a significant issue. Proposed generation algorithms are subjected to many tests to establish their quality, or, more usually, their lack of it. One of the common tests is the run test.

In this test, sequences are tested for ``runs up'' and ``runs down.''

We will examine series of data values for the ``Up'' and ``Down'' sequences each series contains.

Within a series, an ``Up'' sequence continues as long as each data-value received is not less than the previous data-value. An ``Up'' sequence terminates when a data-value received is less than the previous data-value received.

A ``Down'' sequence continues as long as each data-value received is not greater than the previous data-value. A ``Down'' sequence terminates when a data-value received is greater than the previous data-value received.

An ``Up'' sequence can be initiated by the termination of a ``Down'' sequence and vice versa. (Sequences initiated in this manner have length one at this initiation point.)

All the initial data-values are part of an ``Up'' sequence, and contribute to its length, if the first deviation of the data-values is upwards.

All the initial data-values are part of a ``Down'' sequence, and contribute to its length, if the first deviation of the data-values is downwards.

If the data-values received don't allow classification as either an ``Up'' or a ``Down'' sequence, the data should be considered to have neither sequence.

Find the average length of both the ``Up'' and the ``Down'' sequences encountered for each input line in the data file. Report these average lengths as each input line is processed.

Input

Each of the separate series to be examined is contained on a single line of input.

Each series to be analyzed consists of at least one and no more than 30 unsigned, non-zero integers. Each integer in a series has at least one digit and no more than four digits. The integers are separated from each other by a single blank character. Each of the series will be terminated by a single zero (0) digit. This terminator should not be considered as being part of the series being analyzed.

The set of series to be analyzed is terminated by a single zero (0) digit as the input on a line. This terminator should not be considered to be a series, and no output should be produced in response to its encounter.

Output

A line with two real values is to be emitted for each input data set encountered. It must begin with the message ``Nr values = N: ", where N is the number of input data in the line; and then to continue with the average values for runs.

First, the average ``Up" run length, then the average ``Down" run length. Separate these values with a space.

Answers must be rounded to six digits after the decimal point.

Sample Input

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

Sample Output

```
Nr values = 3:  2.000000 0.000000
Nr values = 3:  0.000000 2.000000
Nr values = 5:  2.000000 2.000000
Nr values = 5:  4.000000 0.000000
Nr values = 5:  0.000000 4.000000
Nr values = 7:  0.000000 6.000000
Nr values = 7:  1.000000 5.000000
Nr values = 4:  0.000000 0.000000
Nr values = 7:  2.500000 1.000000
```

题目 3

Description

有两堆石子，数量任意，可以不同。游戏开始由两个人轮流取石子。游戏规定，每次有两种不同的取法，一是可以在任意的一堆中取走任意多的石子；二是可以在两堆中同时取走相同数量的石子。最后把石子全部取完者为胜者。现在给出初始的两堆石子的数目，如果轮到你先取，假设双方都采取最好的策略，问最后你是胜者还是败者。

Input

输入包含若干行，表示若干种石子的初始情况，其中每一行包含两个非负整数 a 和 b，表示两堆石子的数目，a 和 b 都不大于 1,000,000,000。

Output

输出对应也有若干行，每行包含一个数字 1 或 0，如果最后你是胜者，则为 1，反之，则为 0。

Sample Input

```
2 1
8 4
4 7
```

Sample Output

```
0
1
0
```