Problem E
Magic Square Numbers

Input file: *testdata.in*
Time limit: 1 second

Problem Description

A **magic square number** is an integer which is a perfect square and can be expressed as the sum of two perfect squares (we do not count 0). For example, 25 is a magic square number because $25 = 5 \;\; 5$ and $25 = 9 + 16$, whereas 9 is not a magic square number. Let n be a magic square number and $n = i + j$, where both i and j are perfect squares and $i \leq j$, then (i, j) is called a factorization of n. In our previous example, (9, 16) is a factorization of 25. However, we do not count (16, 9) as a factorization of 25.

Write a program that allows users to determine if an input integer is a magic square number and find all its factorizations.

Input Format

Input will consist of a series of lines. Each line consists of an integer n $(1 \leq n \leq 2,147,483,647)$ to be checked if it is a magic square. The input will be terminated by a line consisting of a zero (0).

Output Format

Output will consist of a series of lines, one for each line of the input. If the input number n is a magic square, then the output should start with a positive integer k, where k is the number of ways to factorize the input integer, and followed by k pairs of positive integers each forms a factorization of the input integer. If there are two or more factorizations of input number n, then output should be ordered by the first number in each factorization. The output should be a value of zero if otherwise.

Sample Input

```
25
81
100
2500
0
```

Sample Output

```
1 (9 16)
0
1 (36 64)
2 (196 2304) (900 1600)
```