

Subdividing a Land

Indigo Real-estate Company is now planning to develop a new housing complex. The entire complex is a square, all of whose edges are equally a meters. The complex contains n subdivided blocks, each of which is a b -meter square. Here both a and b are positive integers.

However the project is facing a big problem. In this country, a percentage limit applies to the subdivision of a land, under the pretext of environmental protection. When developing a complex, the total area of the subdivided blocks must not exceed 50% of the area of the complex; in other words, more than or equal to 50% of the newly developed housing complex must be kept for green space. As a business, a green space exceeding 50% of the total area is a *dead space*. The primary concern of the project is to minimize it.

Of course purchasing and developing a land costs in proportion to its area, so the company also wants to minimize the land area to develop as the secondary concern. You, a member of the project, were assigned this task, but can no longer stand struggling against the problem with your pencil and paper. So you decided to write a program to find the pair of minimum a and b among those which produce the minimum dead space for given n .

Input

The input consists of multiple test cases. Each test case comes in a line, which contains an integer n . You may assume $1 \leq n \leq 10000$.

The end of input is indicated by a line containing a single zero. This line is not a part of the input and should not be processed.

Output

For each test case, output the case number starting from 1 and the pair of minimum a and b as in the sample output.

You may assume both a and b fit into 64-bit signed integers.

Example

Input:

```
1
2
0
```

Output:

```
Case 1: 3 2
Case 2: 2 1
```