Perhaps you all have heard the mythical story about Tower of Hanoi (The details of this story is not required to solve this problem): 'There is a tower of Hanoi with 64 disks and three pegs and the preists make one move everyday and the earth will be destroyed when all the pegs have been moved from one peg to the other following the rules of Tower of Hanoi.' In this problem we deal with a similar story — The story of an ancient temple. The ancient temple has three incredibly large bells. At the beginning of time the three bells rang together. Then the three bells never rang together and when they will ring together again the earth will be destroyed. The three bells have cycle length of t_1, t_2 and t_3 (Here $t_1 < t_2 < t_3$ and all are expressed in miliseconds). By this I mean that the first bell rings at every t_1 seconds from the beginning, the second bell rings at every t_2 second from the beginning and the third bell rings at every t_3 second from the beginning. Also note that the difference of the values of t_1, t_2 and t_3 is not that much different so that ordinary people think many time that they are ringing together.

Given the time difference between destruction of earth and beginning of time you will have to find the values of t_1, t_2 and t_3.

Input

The input file contains at most 600 lines of inputs. Each line contains an integer which denotes (in millisecond) the time difference between the beginning of time and the time of the bells ringing together. Input is terminated by a line containing a single zero. All the input numbers will fit in a 64 bit signed integer.

Output

For each line of input produce two lines or more of output. The first line contains the serial of output. Each of the next lines contains three integers which denote the values of t_1, t_2 and t_3 respectively. The value of t_1, t_2 and t_3 is such that $t_1 < t_2 < t_3$ and $0 < t_1, t_2, t_3 \leq 1000000$ and $|t_1 - t_3| \leq 25$. If you cannot find values of t_1, t_2, t_3 with such constraints then print the line "Such bells don't exist" instead. In case there is more than one solution sort the output in ascending order of the value of t_1, then (in case of a tie) in the ascending order of the value of t_2 and then (still a tie) in ascending order of the value t_3. Print a blank line after the output for each test case. Look at the output for sample input for details.

Sample Input

```
10
103
0
```

Sample Output

```
Scenario 1:
1 2 5
1 2 10
1 5 10
2 5 10

Scenario 2:
Such bells don't exist
```