Threesome Poker

Problem Statement

Help IBM puzzlemaster to test answers to May's 2015 challenge:

> Three people are playing the following betting game.
>
> Every five minutes, a turn takes place in which a random player rests and the other two bet against one another with all of their money. The player with the smaller amount of money always wins, doubling his money by taking it from the loser.
>
> For example, if the initial amounts of money are 1, 4, and 6, then the result of the first turn can be either 2,3,6 (1 wins against 4); 1,8,2 (4 wins against 6); or 2,4,5 (1 wins against 6). If two players with the same amount of money play against one another, one of the two players would win, and the game will immediately end for all three players after that turn.
>
> The task for the challenge is to find initial amounts of money for the three players, where none of the three has more than 255, and in such a way that the game cannot end in less than one hour.
>
> In the example above (1,4,6), there is no way to end the game in less than 15 minutes. One possible sequence of moves is:
>
> (1,4,6)
> (1,8,2), after player 2 plays player 3 in turn 1
> (2,6,2), after player 1 plays player 2 in turn 2
> (4,6,0), after player 1 plays player 2 in turn 3, and we arbitrarily choose player 2 as the winner
>
> Note: The first turn is played after five minutes, so the game above, with its 3 turns, takes 15 minutes. Furthermore, the maximum number of turns that can be completed in one hour is 11.

Input Format

Three space-separated, positive integers representing a potential solution to the challenge.

Output Format

The program should output either the word "Ok" if the answer is correct (i.e. the game needs an hour or longer to end), or in case there exists a series of turns in which a player wins the game in less than an hour, then the program should output at most 12 lines showing the amounts of money that each player has. In particular, in the latter scenario, the first line of the output should be identical to the given input, and then each subsequent line should represent a possible state of the game after a turn is played, with the previous line considered as a starting state. Since by definition in this scenario there will be a winner, one of the amounts in the last line of the output should be zero.

If there are several ways for the game to end in less than an hour, find the shortest one, i.e. the one with the fewest turns. If there are several shortest ones, choose the one in which the winning player and losing player in each round is first in lexicographical order. Please refer to the explanation of the sample input and output for more information about this tie breaking procedure.

Sample Input

```
3 27 8
```

Sample Output

```
3 27 8
6 24 8
6 16 16
6 32 0
```

Explanation

The first line of the output is identical to the given input. Then, in the first turn of the game (second line of the output), player 1 beats player 2. In the second turn (third line of the output), player 3 beats player 2. In the final round (final line of the output) player 2 beats player 3.

This game-play can be summarized in a single line comprised of space separated pairs that describe the winner and the loser of each turn in the format: ([winner],[loser]). For the example above, this string summary would be: "(1,2) (3,2) (2,3)" since at first player 1 beat player 2, then player 3 beat player 2 and finally the game ended with player 2 winning over player 3.

Of course, other game-plays could also result in the same result (i.e. have a player winning the game in less than an hour), some with more turns than the one described above, but some as equally as short (in term of number of turns). For example, such another equally short series of turns is the following:

```
3 27 8
3 19 16
6 16 16
6 0 32
```

However, this game-play would be summarized by the following string: "(3,2) (1,2) (3,2)" since at first player 3 beats player 2, then player 1 beats player 2 and finally the game ended with player 3 winning over player 2.

Note though that the last solution has a string representation that comes later in lexicographical order compared to our first solution (i.e. solution "(1,2) (3,2) (2,3)" comes before "(3,2) (1,2) (3,2)" in lexicographical order).

Similarly, another possible series of turns would be:

```
3 27 8
3 19 16
6 16 16
6 0 32
```

This series would be represented by the string: "(3,2) (1,2) (2,3)", which also comes after "(1,2) (3,2) (2,3)" in lexicographical order.

Indeed if you listed all possible games that end in three turns, "(1,2) (3,2) (2,3)" would be lexicographically smaller than all other string representations of games. Thus the expected output should be:

```
3 27 8
6 24 8
6 16 16
6 32 0
```

Note: There are two additional sample test cases. You can view these cases after clicking on the "Run" button.