

## 10858 Unique Factorization

Some of you may have heard the joke about a boy who stopped going to school because he thought that his teacher was crazy. When his parents asked as to why he felt that way, the boy replied, “two days back, my math teacher said,  $12 = 4 \times 3$ , and yesterday she said  $12 = 6 \times 2$ . How can I trust a teacher like this who contradicts her own saying?”. His parents explained the situation to him, so the boy was convinced and continued with his studies. Few years later, the boy wonders, how many ways are there to factorize a number. He sits with pen and paper and discovers that the task is getting quite tedious for large numbers. So he asks you for help. He wants you to write a program to solve his problem.

### Input

The input file will contain at most 20 cases. Each line of input contains a positive integer,  $N \leq 2000000$ . The last case is followed by a value of 0 for  $N$ , which should not be processed.

### Output

For every case, there will be one or more lines of output. The first line of each set of output should contain the number  $F$ , where  $F$  is the number of ways to uniquely factorize  $N$ . If  $F > 0$ , then the next  $F$  lines should contain the Factorizations of  $N$ . The numbers in the factorizations should be sorted in non-decreasing order and individual factorizations should be sorted in dictionary order. When printing numbers in a line, there should be a space between each number. See sample output of  $N = 20$  for clarification. Permutations of factorization are considered same. That is,  $12 = 4 \times 3$  is same as  $12 = 3 \times 4$ .

**Note:** To limit the answers to finite values, the factorizations should not contain any 1, and there should be at least two numbers in the factorizations. So for this problem,  $20 = 1 \times 20$ , is not a factorization of 20.

### Sample Input

```
1
20
0
```

### Sample Output

```
0
3
2 2 5
2 10
4 5
```