A number of students are members of a club that travels annually to exotic locations. Their destinations in the past have included Indianapolis, Phoenix, Nashville, Philadelphia, San Jose, Atlanta, Eindhoven, Orlando, Vancouver, Honolulu, Beverly Hills, Prague, Shanghai, and San Antonio. This spring they are hoping to make a similar trip but aren't quite sure where or when.

An issue with the trip is that their very generous sponsors always give them various knapsacks and other carrying bags that they must pack for their trip home. As the airline allows only so many pieces of luggage, they decide to pool their gifts and to pack one bag within another so as to minimize the total number of pieces they must carry.

The bags are all exactly the same shape and differ only in their linear dimension which is a positive integer not exceeding 1000000. A bag with smaller dimension will fit in one with larger dimension. You are to compute which bags to pack within which others so as to minimize the overall number of pieces of luggage (i.e. the number of outermost bags). While maintaining the minimal number of pieces you are also to minimize the total number of bags in any one piece that must be carried.

Input

Standard input contains several test cases. Each test case consists of an integer $1 \leq n \leq 10000$ giving the number of bags followed by n integers on one or more lines, each giving the dimension of a piece. A line containing `0` follows the last test case.

Output

For each test case your output should consist of k, the minimum number of pieces, followed by k lines, each giving the dimensions of the bags comprising one piece, separated by spaces. Each dimension in the input should appear exactly once in the output, and the bags in each piece must fit nested one within another. If there is more than one solution, any will do. Output an empty line between cases.

Sample Input

```
6
1 1 2 2 2 3
0
```

Sample Output

```
3
1 2
1 2
3 2
```