



A: Magic Scale

Time limit: 3 seconds



A treasure is locked into a room of a well-guarded dungeon. At the door of the room, there is a magic scale and a set of N weights. The scale has an associated unknown target total weight W . We have the following properties:

- It is guaranteed that a subset of the weights will sum to exactly W .
- When the total weight on the scale is exactly W , the door opens, unlocking the treasure.
- If the total weight on the scale is less than W , nothing happens.
- As a security measure, if the total weight on the scale exceeds W , the door will be forever locked.

One way to guarantee opening the door and reaching the coveted treasure is therefore to try all subsets of the N weights in order of increasing total weight. However, multiple subsets might have the same total weight, so a better strategy is to try one subset for each given total weight.

Help our adventurer by enumerating the first K possible total weights in increasing order, together with one corresponding subset of weights for each total weight.

Input

- The first line contains a single integer N .
- The second line contains a single integer K .
- The third line contains a space-separated list of N integers representing the N weights.

Output

K lines representing the first K possible total weights in increasing order, together with the corresponding weights. The format of each of the K lines is:

```
total_weight: weight_1 weight_2 ... weight_p
```

where `weight_1 weight_2 ... weight_p` are p space-separated weights that sum to `total_weight` (if there are multiple options, any will be accepted).

It is guaranteed that it will be possible to find at least K different weight sums given the input data.

Limits

- $1 \leq N \leq 1000$
- $1 \leq K \leq 1000$
- Each weight is in $[1, 1000000]$

Sample Input

```
5
10
1 12 4 5 100
```

Sample Output

```
0:
1: 1
4: 4
5: 1 4
6: 1 5
9: 4 5
10: 1 4 5
12: 12
13: 1 12
16: 4 12
```