3345 Numbering Cities

Scientists and engineers have found a way for human to settle in a far-away planet. The new immigrants have started building small towns and communication links connecting nearby towns. In building the communication network, they first chose a town as a center. Then they connected each nearby town to the center by a communication link.

As more and more people migrate to the new planet, a number of such networks have been built. These networks need to be connected to make it possible for people in every town in the planet to communicate. They connected these networks by connecting the centers using a simple path.

The first group of immigrants are mostly scientists and engineers. They prefer numbering their towns rather than naming them. Consider n towns connected by a network with only one center. If we number the center 0 and the nearby towns by integers in $\{1, 2, \ldots, n-1\}$. There is an interesting property of this numbering method. Suppose that we label of each link by the absolute value of the difference of the numbers of the two towns it connects. All links will get a distinct label. Furthermore, the labels of the links are exactly $\{1, 2, \ldots, n-1\}$. Thus, links are numbered automatically. This is called perfect numbering of the towns.

The immigrants love the numbering method very much, and wants to know if the numbering can be extended to the networks with many centers. Therefore, you are invited to write a program to solve the problem.

Let us study more about perfect numbering of the towns. Assume that all towns are properly numbered. It is easy to see that if the numbers of any two towns connected to the same center are switched, then the new numbering is also perfect. Therefore, perfect numbering of the towns is not unique.

It is convenient to assign a name to each town. Assume that there are at most 26 centers, and they are labeled by `a0, b0, ..., z0`. The towns connected to center `a0` will be called `a1, a2, ...`, the towns connected to center `b0` will be called `b1, b2, ...`, and so on.

To make the problem simple, we want a perfect numbering such that the numbers assign to the towns connected to the same center are ordered by their names. For example, `ai`'s label will be smaller than `aj`'s label whenever $i < j$. Finally, it is required to label `a0` with 0 whenever possible.

Input

The input file contains several test cases. Each test case is described in 2 lines. The first line is an integer n, which is the number of centers. The second line contains n integers. Each integer indicates the number of nearby towns connected to the corresponding center. For example,

```
4
3 2 2 4
```

means that there are 4 centers `a0`, `b0`, `c0`, and `d0`. The center `a0` connects 3 nearby towns, the center `b0` connects 2 nearby towns, etc.

Assume that each center can connect at most 60 towns.

The last test case is followed by a '`0`', which signals the end of input file.

Output

For each test case print out the labels of each town. The output should be ordered by labels, not by the names of the towns. Print 8 '`(`*town*`,`*label*`)`' pairs in a line, except the last line. Print 1 blank line in front of each test case. If there are no solutions, print '`no solutions`'.

Sample Input

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

Sample Output

```
(a0, 0) (b1, 1) (b2, 2) (c0, 3) (d1, 4) (d2, 5) (d3, 6) (d4, 7)
(d0, 8) (c1, 9) (c2, 10) (b0, 11) (a1, 12) (a2, 13) (a3, 14)

(a0, 0) (c0, 1) (e0, 2) (d0, 3) (b0, 4)
```