Question - 1
The Social Network

A social network has *n* active users, numbered from *0* to *n — 1*, who selectively friend other users to create groups of friends within the network. We define the following:

- Two users, *x* and *y*, are *direct* friends if they friend each other on the network.
- Two users, *x* and *z*, are *indirect* friends if there exists some direct friend, *y*, common to both users *x* and *z*.
- Two users, *x* and *y*, belong to the same *group* if they are friends (either directly or indirectly) with each other. In other words, if user *x* is part of a group, then all of user *x*'s friends and friends of friends belong to the same group.
- We describe the number of people in each group as an array of *n* integers, *counts*, where each $counts_i$ ($0 \le i < n$) denotes the total number of users in the group that user *i* belongs to. For example, if *counts = [3, 3, 3, 3, 3, 1, 3]*, then there are three groups; users *0*, *1*, *2*, *3*, *4*, and *6* are in one of two *3*-person groups, and user *5* is in a *1*-person group.

ID	0	1	2	3	4	5	6
Group Size	3	3	3	3	3	1	3

- A group is *valid* if all the users in the group have minimal ID numbers. In other words, a group of size *k* must contain the *k* smallest ID numbers belonging to a group of that size with respect to the smallest user ID in the group. For example, if *counts = [3, 3, 3, 3, 3, 1, 3]*, then the grouping *[0, 1, 2]*, *[3, 4, 6]*, and *[5]* is valid; however, the grouping *[0, 1, 4]*, *[2, 3, 6]*, and *[5]* is *not valid* because the group *[0, 1, 4]* does not contain the three smallest user IDs for the set of user IDs belonging to *3*-person groups (i.e., *{0, 1, 2, 3, 4, 6}*).

  ▶ **Example**

- We print the information for each valid group on a new line in the format $user_{smallest\ ID}$ … $user_{largest\ ID}$, where the users within each group are ordered by ascending ID and the groups themselves are ordered by ascending $user_{smallest\ ID}$. For example, we print the valid grouping *[0, 1, 2]*, *[3, 4, 6]*, and *[5]* for *counts = [3, 3, 3, 3, 3, 1, 3]* as:

```
0 1 2
3 4 6
5
```

Complete the *socialGraphs* function. It has one parameter: an array of *n* integers, *counts*, where each $counts_i$ denotes the total number of users in the group that user *i* belongs to. The function must print the information for each *valid group* in the format specified above.

Help

/

Input Format

The first line contains an integer, *n*, describing the number of elements in *counts* (i.e., the total number of users active on the social network).
Each line *i* of *n* subsequent lines (where $0 \le i < n$) contains an integer describing $counts_i$.

Constraints

- $1 \le n \le 2 \times 10^5$
- $1 \le counts_i \le n$, where $0 \le i < n$.
- It is guaranteed that a valid grouping always exists for the given *counts* array.

Output Format

The function must *print* the information for each *valid group* on a new line. The users within a group must be ordered by ascending user ID, and each group must be ordered by ascending smallest user ID.

▼ Sample Case 0

Sample Input

```
4
2
2
2
2
```

Sample Output

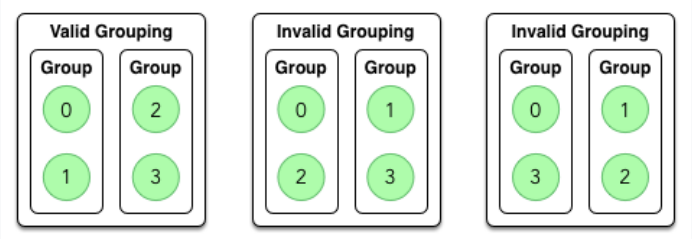
```
0 1
2 3
```

Explanation

We express *counts = [2, 2, 2, 2]* as the following table of group sizes:

ID	0	1	2	3
Group Size	2	2	2	2

The *valid* grouping here is the groups *[0, 1]* and *[2, 3]*:



Possible groupings for *counts = [2, 2, 2, 2]*.

We then print each group on a new line, where the groups and their user IDs are listed in ascending order.

/