Problem Statement:

You need to process a list of integers and group them into **consecutive sequences**. A consecutive sequence is defined as a group of numbers where each number is exactly **one greater** than the previous one. Your task is to identify all such consecutive sequences in the list.

Input Format:

- The first input is an integer N, the length of the list.
- The next N lines contain the list elements as integers, sorted in **ascending** order.

Output Format:

- A list of lists, where each inner list represents a consecutive sequence found in the input.

Constraints:

- $1 \leq N \leq 10^5$
- $-10^6 \leq$ list elements $\leq 10^6$
- The list is sorted in **non-decreasing order**.

Example 1

Input:

10

1

2

3

5

6

8

9

10

15

16

Output:

[[1, 2, 3], [5, 6], [8, 9, 10], [15, 16]]

Example 2

Input:

5

4

5

6

7

8

Output:

[[4, 5, 6, 7, 8]]

Solution:

 # Added longest arithmetic sequence finder in Python. Work in progress, optimizing further!

```
N = int(input())

ls = []

for i in range(N):
    x = input()
    ls.append(int(x))

max_seq_dict = {}

i = 0

while i < N:
    c = 1
    for j in range(i+1, N):
        if (int(ls[j])-int(ls[i])) == (j-i):
            c += 1
        else:
            break
    max_seq_dict[i] = c
    i += c


final_ls = []
```

```
print(max_seq_dict)


for key,val in max_seq_dict.items():
    if val > 1:
        start_index = key
        stop_index = start_index + val
        temp = ls[start_index:stop_index]
        final_ls.append(temp)

print(final_ls)
```