CSCI 406: AlgoBOWL: Problem Description

Reminder: You are NOT allowed to consult the internet to solve this problem.

Your input is a sorted list of positive integers $X = \{x_1, \cdots, x_n\}$ with length $n \geq 2$. The objective is to design an algorithm that determines a *minimal length* sequence of addition operations starting with 1 that computes all values in X. In other words, can you find the shortest sequence of addition operations that each take only previously computed sums or the number 1, which compute the values in X?

For example, for the sequence $X = \{2, 5\}$, a naïve solution repeatedly adds 1 to the previous sum (or 1 if there is no previous sum) until all numbers in X have been computed:

$$1 + 1 = \mathbf{2}$$
$$2 + 1 = 3$$
$$3 + 1 = 4$$
$$4 + 1 = \mathbf{5}$$

The naïve solution requires 4 addition operations, but an optimal solution for this input X only requires 3 operations:

$$1 + 1 = \mathbf{2}$$
$$2 + 2 = 4$$
$$4 + 1 = \mathbf{5}$$

A sequence is invalid if any addition operation takes an operand that has not previously been computed in the sequence (and is not 1):

$$1 + 1 = \mathbf{2}$$
$$2 + 3 = \mathbf{5}$$

$\Longrightarrow$ **INVALID! 3 has not been previously computed in this sequence.**

Input Format: The first line contains one integer, n, the length of the sequence X. The second line contains n ascending positive integers which represent the list X.

```
2
2 5
```

Input Restrictions: $2 \leq n \leq 1000$; $\max(X) \leq 10^9$

Output Format: Line 1 will contain the number of addition operations j in your solution. Then on each of the next j lines, print the operands from the j^{th} addition operation in your solution. The addition operations may appear in any order in the output, as long as any given operation appears after the two operations that compute its operands. The operands for a given operation need not appear in any particular order.

```
3
1 1
2 2
4 1
```