C. Ehab and a 2-operation task

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

You're given an array a of length n. You can perform the following operations on it:

- choose an index i $(1 \le i \le n)$, an integer x $(0 \le x \le 10^6)$, and replace a_j with $a_j + x$ for all $(1 \le j \le i)$, which means add x to all the elements in the prefix ending at i.
- choose an index i $(1 \le i \le n)$, an integer x $(1 \le x \le 10^6)$, and replace a_j with $a_j \% x$ for all $(1 \le j \le i)$, which means replace every element in the prefix ending at i with the remainder after dividing it by x.

Can you make the array **strictly increasing** in no more than $n + 1$ operations?

Input

The first line contains an integer n $(1 \le n \le 2000)$, the number of elements in the array a.

The second line contains n space-separated integers $a_1, a_2, \ldots, a_n$ $(0 \le a_i \le 10^5)$, the elements of the array a.

Output

On the first line, print the number of operations you wish to perform. On the next lines, you should print the operations.

To print an adding operation, use the format "$1\ i\ x$"; to print a modding operation, use the format "$2\ i\ x$". If i or x don't satisfy the limitations above, or you use more than $n + 1$ operations, you'll get *wrong answer* verdict.

Examples

input	Copy
3 1 2 3	

output	Copy
0	

input	Copy
3 7 6 3	

output	Copy
2 1 1 1 2 2 4	

Note

In the first sample, the array is already increasing so we don't need any operations.

In the second sample:

In the first step: the array becomes $[8, 6, 3]$.

In the second step: the array becomes $[0, 2, 3]$.