

B. Toy Sum

time limit per test: 1 second

memory limit per test: 256 megabytes

input: standard input

output: standard output

Little Chris is very keen on his toy blocks. His teacher, however, wants Chris to solve more problems, so he decided to play a trick on Chris.

There are exactly s blocks in Chris's set, each block has a unique number from 1 to s . Chris's teacher picks a subset of blocks X and keeps it to himself. He will give them back only if Chris can pick such a non-empty subset Y from the remaining blocks, that the equality holds:

"Are you kidding me?", asks Chris.

For example, consider a case where $s = 8$ and Chris's teacher took the blocks with numbers 1, 4 and 5. One way for Chris to choose a set is to pick the blocks with numbers 3 and 6, see figure. Then the required sums would be equal: $(1 - 1) + (4 - 1) + (5 - 1) = (8 - 3) + (8 - 6) = 7$.

However, now Chris has exactly $s = 10^6$ blocks. Given the set X of blocks his teacher chooses, help Chris to find the required set Y !

Input

The first line of input contains a single integer n ($1 \leq n \leq 5 \cdot 10^5$), the number of blocks in the set X . The next line contains n distinct space-separated integers x_1, x_2, \dots, x_n ($1 \leq x_i \leq 10^6$), the numbers of the blocks in X .

Note: since the size of the input and output could be very large, don't use slow output techniques in your language. For example, do not use input and output streams (cin, cout) in C++.

Output

In the first line of output print a single integer m ($1 \leq m \leq 10^6 - n$), the number of blocks in the set Y . In the next line output m distinct space-separated integers y_1, y_2, \dots, y_m ($1 \leq y_i \leq 10^6$), such that the required equality holds. The sets X and Y should not intersect, i.e. $x_i \neq y_j$ for all i, j ($1 \leq i \leq n$; $1 \leq j \leq m$). It is guaranteed that at least one solution always exists. If there are multiple solutions, output any of them.

Examples

input
3 1 4 5
output
2 999993 1000000

input
1 1
output
1 1000000