# Problem B. Bloom

| | |
|---|---|
| Input file: | *standard input* |
| Output file: | *standard output* |
| Time limit: | 3 seconds |
| Memory limit: | 256 mebibytes |

Bloom filter is a space-efficient probabilistic data structure that is used to test whether an element is a member of a set. Elements can be added to the set but not removed.

A Bloom filter for a universe $S$ is an array of $m$ bits initially set to zero and $l$ different hash functions $S \to [0, m)$: $h_1, \ldots, h_l$.

To add an element $s \in S$ to the set, feed it to each hash function and set bits $h_1(s), \ldots, h_l(s)$ to one. To test whether an element is in the set, look at bits $h_1(s), \ldots, h_l(s)$. If any of them is zero, this element is "definitely not in the set", otherwise it is "possibly in the set". So, Bloom filter allows false positive matches but not false negatives.

Your teammate implemented his own Bloom filter for integers from 0 to $10^9$ inclusive. He selected a prime number $m$ and defined one hash function (otherwise it would be too complicated for him): $h(s) = (As + B) \bmod m$ where $0 \leqslant A, B < m$. Then he selected $n$ numbers $a_1, \ldots, a_n$ and added them to the filter. After this operation, $k$ bits were set: $b_1, \ldots, b_k$.

You want to reproduce his experiment. Unfortunately, you don't know the values of $A$ and $B$. So you decided to find all pairs ($A$, $B$) such that adding all numbers $a_1, \ldots, a_n$ to the Bloom filter using your parameters will set exactly those bits.

## Input

The first line of input contains three integers $n$, $k$ and $m$ ($1 \leqslant n, k \leqslant 10^6$, $2 \leqslant m \leqslant 10^6$, $m$ is prime). The second line of input contains $n$ integers $a_1, \ldots, a_n$ ($0 \leqslant a_i \leqslant 10^9$) which are the numbers added to the filter. The third line contains $k$ integers $b_1, \ldots, b_k$ ($0 \leqslant b_j < m$) which are the numbers of bits that must be set to one after adding all elements to the filter. All other bits must be zeroes.

## Output

On the first line of output, print one integer $p$, the number of pairs. On each of the next $p$ lines, print two integers $A$ and $B$ which represent a pair of parameters such that adding all numbers $a_1, \ldots, a_n$ to the Bloom filter using these parameters will set exactly the bits $b_1, \ldots, b_k$.

It is guaranteed that the number of pairs $p$ will be at most $10^6$.

## Examples

| standard input | standard output |
|---|---|
| 4 4 11<br>1 6 8 9<br>0 2 8 9 | 1<br>3 6 |
| 4 4 11<br>1 6 8 9<br>1 2 3 4 | 0 |
| 6 2 11<br>11 12 22 23 33 34<br>0 1 | 2<br>1 0<br>10 1 |