# Task Bitovi

What came first, the chicken or the egg? Is it better to live a hundred years as a millionaire or seven days in poverty? How to become a chess grandmaster? ~~How to raise blinds?~~ How to pass the final exams? How to train a dragon? These are interesting questions we can ponder only after the competition, but now we offer one *less interesting* computer science problem.

You are given two sets of numbers $A$ and $B$ of size $N$. In one move, you can select an arbitrary element from set $A$ and change one arbitrary digit (bit) in its binary representation. The resulting number must not be an element of set $A$ immediately before the change.

For example, the number 5 in binary is $0101_2$. In one move, it can become $13 = 1101_2$, $1 = 0001_2$, $7 = 0111_2$, or $4 = 0100_2$ if we change its 4th, 3rd, 2nd, or 1st bit, respectively.

Determine a sequence of moves by which set $A$ becomes equal to set $B$. Sets are equal if they have the same size and there is no element in set $A$ that does not belong to set $B$.

**Note:** The number of moves does not have to be minimal, but it must satisfy the task constraints.

## Input

The first line contains the integer $N$ ($1 \leq N \leq 2^{15}$), the size of the sets $A$ and $B$.

The second line contains $N$ different integers $a_i$ ($0 \leq a_i < 2^{15}$), the elements of the set $A$.

The third line contains $N$ different integers $b_i$ ($0 \leq b_i < 2^{15}$), the elements of the set $B$.

## Output

In the first line, print the number of required moves.

In the remaining lines, print the numbers $x$ and $y$ ($0 \leq x, y < 2^{15}$) – we change the number $x$ from set $A$ to the number $y$. The numbers $x$ and $y$ must differ by exactly one bit, and $x \in A$ and $y \notin A$ must hold at the moment we execute the move.

## Scoring

The number of moves you are allowed to make in all subtasks must not exceed $2^{19}$. It can be shown that a solution always exists within the given constraints.

| Subtask | Points | Constraints |
|---|---|---|
| 1 | 10 | $a_i, b_i \leq 2^{14}$ |
| 2 | 15 | $N \leq 7$ |
| 3 | 30 | $N \leq 2^7$ |
| 4 | 15 | No additional constraints. |

## Examples

| input | input | input |
|---|---|---|
| 3<br>0 1 2<br>1 2 3 | 3<br>4 8 31<br>0 4 8 | 5<br>0 1 2 4 5<br>7 6 5 3 2 |
| **output** | **output** | **output** |
| 2<br>1 3<br>0 1 | 5<br>31 30<br>30 28<br>28 24<br>24 16<br>16 0 | 9<br>1 3<br>3 7<br>0 1<br>1 0<br>2 6<br>0 2<br>7 3<br>5 7<br>4 5 |

**Clarification of the first example:**
If we first make the move `0 1`, and then `1 3`, between these two moves, set $A$ would have two identical elements, which the task does not allow. Another possible solution is `2 3`, `0 2`.

**Clarification of the second example:**
$31 = 11111_2$. By removing bits from least to most significant, we obtain the numbers 30, 28, 24, 16, and 0 in sequence. After all moves, set $A$ becomes equal to set $B$.