Problem F. Fractional XOR Maximization

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	512 mebibytes

Define the *bitwise XOR operation* $\oplus$ for any two real numbers x and y as follows:

$$x \oplus y = \lim_{n \to \infty} \frac{\lfloor 2^n x \rfloor \oplus \lfloor 2^n y \rfloor}{2^n},$$

where $\oplus$ in the right-hand side of the equation is the integer bitwise XOR operation.

For example, $\frac{5}{8} \oplus \frac{3}{8} = \frac{3}{4}$, $\frac{1}{3} \oplus \frac{1}{7} = \frac{4}{9}$, and $\frac{1}{5} \oplus \frac{1}{7} = \frac{6}{65}$.

Recall that the *supremum* of a set $X \subseteq \mathbb{R}$ (denoted by $\sup X$) is the least real number that is greater than or equal to all elements of X.

Given four non-negative rational numbers, a, b, c, and d, find $\sup \{x \oplus y : x \in [a, b], y \in [c, d]\}$. In other words, find the least value that is greater than or equal to all XOR values of an element from $[a, b]$ and an element from $[c, d]$.

Input

The first line contains a single integer T ($1 \le T \le 100$), the number of test cases.

Then T test cases follow. The first line of each test case contains four integers a_{num}, a_{denom}, b_{num}, b_{denom} ($0 \le a_{num}, b_{num} \le 10^{17}$, $1 \le a_{denom}, b_{denom} \le 30$), describing fractions $a = \frac{a_{num}}{a_{denom}}$ and $b = \frac{b_{num}}{b_{denom}}$. The second line of each test case describes fractions c and d in the same format.

It is guaranteed that $a \le b$ and $c \le d$, and all fractions from input are irreducible (in other words, the greatest common divisor of the numerator and denominator is equal to 1).

Output

Print T lines. The i-th line should contain two integers x_{num} and x_{denom} separated by a single space: the numerator and denominator of the answer $x = \frac{x_{num}}{x_{denom}}$ for the i-th test case, expressed as an irreducible fraction. It can be shown that the answer is always a rational number.

Example

standard input	standard output
2 0 1 1 1 0 1 1 1 5 7 5 7 3 16 3 16	2 1 59 112

Note

In the first sample test case, the answer is 2 because we can make XOR value arbitrarily close to 2 choosing 1 from the first interval and $1 - 2^p$ from the second interval, where p is a large enough integer, but we can obtain neither exactly 2 nor any greater number.