Task 1: Morton Numbers

Background

The Morton number of two integers x and y is the integer formed by interleaving the bits of x and y in binary such that the bits of x are in the even position and the bits of y are in the odd position.

If we consider x and y to represent the 2D coordinates of a point, then the Morton numbers have the property that if two points are close to each other, then their respective Morton numbers will also be close to one another. Furthermore the Morton numbers are proportional to x and y.

Problem Description

In the problem, you will be given two integers x and y in base 10 ($0 \leq x, y \leq 2^{16} - 1$) and you should return the Morton number of x and y in base 10.

For example if x = 4 and y = 5,
then x = **0000000000000100** and y = 0000000000000101 in binary. Hence the Morton number of x and y is **0**0**0**0**0**0**0**0**0**0**0**0**0**0**0**0**0**0**0**0**0**0**0**0**0**0**1**1**0**0**0**1 in binary which is equivalent to 49. Note that the bits in bold represents the bits from x.

Input (Morton.in)

The input consists of the two integers, x and y, on the same line separated by whitespace. A line is terminated by DOS eof: \r\n

Output (Morton.out)

Output the Morton number of x and y.

Sample Input

```
4 5
```

Sample Output

```
49
```

Additional Notes

1. Converting from binary to decimal

The place values of the digits in the binary number are 2^0, 2^1, 2^2 and so on from right to left. For example:

10011011_2
=> $1*2^0 + 1*2^1 + 0*2^2 + 1*2^3 + 1*2^4 + 0*2^5 + 0*2^6 + 1*2^7$
=> $1 + 2 + 0 + 8 + 16 + 0 + 0 + 128$
=> 155_{10}

2. Converting from decimal to binary

To convert a whole number to binary, use successive division by 2 until the quotient is 0. The remainders form the answer, with the first remainder as the least significant bit (LSB) and the last as the most significant bit (MSB). $43_{10} = 101011_2$. The working is shown below:

2 | 43
2 | 21 rem 1 ← LSB
2 | 10 rem 1
2 | 5 rem 0
2 | 2 rem 1
2 | 1 rem 0
0 rem 1 ← MSB