Bitwise Operators

Inside the CPU, mathematical operations like addition, subtraction, multiplication and division are done in bit-level. To perform bit-level operations in C programming, bitwise operators are used which are explained below.

Bitwise AND operator & The output of bitwise AND is 1 if the corresponding bits of two operands is 1. If either bit of an operand is 0, the result of corresponding bit is evaluated to 0. It is denoted by &.

Bitwise OR operator | The output of bitwise OR is 1 if at least one corresponding bit of two operands is 1. It is denoted by |.

Bitwise XOR (exclusive OR) operator ^ The result of bitwise XOR operator is 1 if the corresponding bits of two operands are opposite. It is denoted by ⊕.

For example, for integers 3 and 5,

3 = 00000011 (In Binary)
5 = 00000101 (In Binary)

AND operation	OR operation	XOR operation
00000011	00000011	00000011
& 00000101	| 00000101	^ 00000101
00000001 = 1	00000111 = 7	00000110 = 6

Given set $S = \{1, 2, 3, \ldots, n\}$, find:

- the maximum value of a&b which is less than a given integer k, where a and b (where a<b) are two integers from set S.
- the maximum value of a|b which is less than a given integer k, where a and b (where a<b) are two integers from set S.
- the maximum value of a⊕b which is less than a given integer k, where a and n (where a<b) are two integers from set S.

Input

The only line contains 2 space-separated integers, n and k, respectively.

Output

The first line of output contains the maximum possible value of a&b.
The second line of output contains the maximum possible value of a|b.
The second line of output contains the maximum possible value of a⊕b.

Sample input	Sample output
5 4	2 3 3
12 4	3 3 3
11 2	1 0 1