

Bitwise Operators

In this challenge, you will use logical bitwise operators. All data is stored in its binary representation. The logical operators, and C language, use **1** to represent true and **0** to represent false. The logical operators compare bits in two numbers and return true or false, **0** or **1**, for each bit compared.

- 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
```

You will be given an integer *n*, and a threshold, *k*. *Foreach number i from 1 through n, find the maximum value of the logical and, or and xor when compared against all integers through n that are greater than i. Consider a value only if the comparison returns a result less than k*. Print the results of the and, or and exclusive or comparisons on separate lines, in that order.

Example

n = 3
k = 3

The results of the comparisons are below:

```
a b    and or xor
1 2    0  3  3
1 3    1  3  2
2 3    2  3  1
```

For the **and** comparison, the maximum is **2**. For the **or** comparison, none of the values is less than *k*, so the maximum is **0**. For the **xor** comparison, the maximum value less than *k* is **2**. The function should print:

```
2
0
```

Function Description

Complete the `calculate_the_maximum` function in the editor below.

`calculate_the_maximum` has the following parameters:

- `int n`: the highest number to consider
- `int k`: the result of a comparison must be lower than this number to be considered

Prints

Print the maximum values for the `and`, `or` and `xor` comparisons, each on a separate line.

Input Format

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

Constraints

- $2 \leq n \leq 10^3$
- $2 \leq k \leq n$

Sample Input 0

```
5 4
```

Sample Output 0

```
2
3
3
```

Explanation 0

$n = 5, k = 4$

$S = \{1, 2, 3, 4, 5\}$

All possible values of *a* and *b* are:

1. $a = 1, b = 2$; $a \& b = 0$; $a | b = 3$; $a \oplus b = 3$;
2. $a = 1, b = 3$; $a \& b = 1$; $a | b = 3$; $a \oplus b = 2$;
3. $a = 1, b = 4$; $a \& b = 0$; $a | b = 5$; $a \oplus b = 5$;
4. $a = 1, b = 5$; $a \& b = 1$; $a | b = 5$; $a \oplus b = 4$;
5. $a = 2, b = 3$; $a \& b = 2$; $a | b = 3$; $a \oplus b = 1$;
6. $a = 2, b = 4$; $a \& b = 0$; $a | b = 6$; $a \oplus b = 6$;

7. $a = 2, b = 5; a \& b = 0; a | b = 7; a \oplus b = 7;$
8. $a = 3, b = 4; a \& b = 0; a | b = 7; a \oplus b = 7;$
9. $a = 3, b = 5; a \& b = 1; a | b = 7; a \oplus b = 6;$
10. $a = 4, b = 5; a \& b = 4; a | b = 5; a \oplus b = 1;$

- The maximum possible value of $a \& b$ that is also $< (k = 4)$ is **2**, so we print **2** on first line.
- The maximum possible value of $a | b$ that is also $< (k = 4)$ is **3**, so we print **3** on second line.
- The maximum possible value of $a \oplus b$ that is also $< (k = 4)$ is **3**, so we print **3** on third line.