

Problem

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.

Submissions

- 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 ^.

Leaderboard

For example, for integers 3 and 5,

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

Discussions

AND operation	OR operation	XOR operation
00000011	00000011	000000
& 00000101	00000101	^ 000001
-----	-----	-----
00000001 = 1	00000111 = 7	000001

Editorial

You will be given an integer n , and a threshold,

k. For each number i from 1 through n , find the maximum value of the logical and, or and xor when 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

$n = 3$

$k = 3$

Change Theme Language: C



```

3  #include <math.h>
4  #include <stdlib.h>
5  //Complete the following function.
6
7
8  void calculate_the_maximum(int n, int
k) {
9      //Write your code here.
10
11     int maxAND, maxOR, maxXOR, andV,
orV, xorV;
12
13     maxAND = maxOR = maxXOR = 0;
14
15     for(int i = 1; i <= n; i++){
16         for(int j = i+1; j <= n ; j++){
17             andV = i & j;
18             orV = i | j;
19             xorV = i ^ j;
20
21             if((andV > maxAND) && (andV <
k))
22                 maxAND = andV;
23             if((orV > maxOR) && (orV < k))
24                 maxOR = orV;
25             if((xorV > maxXOR) && (xorV <
k))
26                 maxXOR = xorV;
27         }
28     }
29
30     printf("%d\n%d\n%d", maxAND, maxOR,
maxXOR);
31 }
32
33 int main() {
34     int n, k;
35
36     scanf("%d %d", &n, &k);
37     calculate_the_maximum(n, k);
38
39     return 0;
40 }
41

```

--INSERT--

Line: 39 Col: 14