Codeforces Round #456 Div.2 B. New Year's Eve

#Problem

codeforces.com/contest/912/problem/B

#Analysis

You can choose k or less than k integers from 1~n, calculate maximum xor result

(1) k = 1, You can only choose one integer, so you choose n and get the maximum xor result n

(2) k > 1, whatever k is, such as k = 2, or k = 3, ......, or k = n,

if n has x bits as binary code, the maximum xor result must be 2 ^ x - 1

本题是求从1~n个数中取小于或等于k个数（比如n = 4, k = 3表示从1~4中取1个数或2个数或3个数），对这些数进行异或求和，求最大的异或求和值。

（1）k = 1时，即从1~n个数中1个数，那么最大的数自然是n。结果为n

（2）k > 1时，假如n以二进制表示是x位数，则结果必为2 ^ x - 1

###Example 1

n = 4, k = 2

choose 4 and 3, maximum = 4 xor 3 = 7

可以取4和3这两个数，最大异或结果 = 4 xor 3 = 7 = 2 ^ 3 - 1

### Example 2

n = 4, k = 3

Choose two numbers, 4 and 3. Or choose three numbers, 4, 1 and 2.

maximum = 4 xor 3 = 4 xor 1 xor 2 = 7 = 2 ^ 3 - 1

可以取4和3这两个数，也可以取4、1、2这三个数。

最大异或结果 = 4 xor 3 = 4 xor 1 xor 2 = 7 = 2 ^ 3 - 1

### Example 3

n = 4, k = 4

Choose two numbers, 4 and 3. Or choose three numbers, 4, 1 and 2.

maximum = 7 = 2 ^ 3 - 1

You can not choose all for numbers, because 4 xor 1 xor 2 xor 3 xor 4 = 3, which is not the maximum result.

可以取4和3，也可以取4、1、2。最大异或结果为7。

但不能取4、1、2、3。因为4 xor 1 xor 2 xor 3 xor 4 = 3，不是最大的结果。

### Example 4

n = 6, k = 6

Just choose 1 and 6, maximum = 6 xor 1 = 7 = 2 ^ 3 - 1

只要在1~6中取6和1两个数就行了，6 xor 1 = 7 = 2 ^ 3 - 1

#Code

###Python2

```

import sys

n, k = map(int, raw\_input().split())

if 1 == k:

print n

sys.exit(0)

res = 1

while res < n:

res = res \* 2 + 1

print res

```

###C Language

```

#include <stdio.h>

#include <stdint.h>

int main(void)

{

int64\_t n, k;

scanf("%I64d %I64d", &n, &k);

if (1 == k)

{

printf("%I64d\n", n);

return 0;

}

int bit = 0;

for (; n >> bit; bit++);

printf("%d", n);

printf("%d", bit);

printf("%I64d\n", (1LL << bit) - 1);

return 0;

}

```