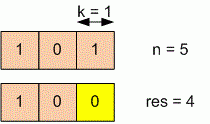
[C - Reset bits](https://vjudge.net/problem/EOlymp-1647" \t "_blank)

 Given two integers **n** and **k**. Reset (set to zero) the last **k** bits in the number **n**, and print the result. It is recommended to find solution without loops.



#### Input

Two numbers **n** (**0** ≤ **n** ≤ 231 - **1**) and **k** (**0** ≤ **k** ≤ **30**).

#### Output

Print the result of resetting **k** last bits.

**Example 1**

Input example #1

5 1

Output example #1

4

### **Creation of Desired MASK**

To clear right MOST 3 bits of a given 8-bit binary number e.g. n = 0010 0111

We should create a MASK with Left Most 5 bits as 1s and right most 3 bits as 0s.

In order to make above MASK first we will create a mask with 5 RIGHT MOST BITS SET and then LEFT SHIFT it 3 times.

How to create a number with 5 RIGHT MOST BITS SET?    1 << 5 gives us a number 00100000, And subtracting 1 from it will give 00011111

So maskPart1 = (1ULL << (32-k) ) - 1.  // Assuming given input is 32-bit number.

final MASK = maskPart1 << k;

ANS = n & MASK;

### **One's Complement Way To Create Mask**

To clear K right Most 3-bits BITS of 8-bit number

n = 0010 0111

mask = 1111 1000 which is 1s complement of 0000 0111 (this is easy to create).

mask = ~(1 << k)

Hex Constant Way To Create Mask

MASK = (0xFFFFFFFF) << 1;

F is HEX NUMBER 15, Binary form of this is 1111.

0xFFFFFFFF is a shorthand way of writing 32-bits all set to 1s.

return n & MASK.

### **Right Shift By K, Then Left Shift By K**

1. n = n >> k;   // This will get rid of k right most bits
2. n = n << k;   // This will fill k right most bits with 0s
3. return n;

​

#include <iostream>

using namespace *std*;

int main(void) {

*ios\_base*::*sync\_with\_stdio*(false);

*cin*.*tie*(nullptr);

*cout*.*tie*(nullptr);

int n, k; *cin* >> n >> k;

int mask = 0XFFFFFFFF;

mask = mask << k;

*cout* << (n&mask) << "\n";

return 0;

}