You are given two numbers a and b where 0 ≤ a ≤ b. Imagine you construct an array of all the integers from a to b inclusive. You need to count the number of 1s in the binary representations of all the numbers in the array.

**Example**

For a = 2 and b = 7, the output should be  
rangeBitCount(a, b) = 11.

Given a = 2 and b = 7 the array is: [2, 3, 4, 5, 6, 7]. Converting the numbers to binary, we get [10, 11, 100, 101, 110, 111], which contains 1 + 2 + 1 + 2 + 2 + 3 = 11 1s.

**Input/Output**

* **[time limit] 3000ms (cs)**
* **[input] integer a**

*Constraints:*  
0 ≤ a ≤ b.

* **[input] integer b**

*Constraints:*  
a ≤ b ≤ 10.

* **[output] integer**

<https://codefights.com/arcade/code-arcade/corner-of-0s-and-1s/eC2Zxu5H5SnuKxvPT>

static string toBin(int n)

{

string bin = "";

while (n > 0)

{

bin = bin.Insert(0, (n % 2).ToString());

n /= 2;

}

return bin;

}

static int rangeBitCount(int a, int b)

{

int sum = 0;

for (int i = a; i <= b; i++)

{

string bin = toBin(i);

int unos = bin.Count(f => f == '1');

sum += unos;

}

return sum;

}