

Binary [III]

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

points: 100 · problem code: binary3 · author: glipR

The blurb for this problem is the same as Binary Sequences [I], save for the statement of the problem.

This set of problems is centered around a sequence generated in the following manner:

1. Take all natural numbers in increasing order
2. Replace each natural number with its binary representation
3. Concatenate these into a single sequence

For example, the first 4 natural numbers generate the first 8 digits in the sequence: `11011100`.

The first digit is `1`, for `1`. The next two are `10`, for `2`. The next two are `11`, for `3`, and the final three are `100`, for `4`.

In this third problem, we are actually looking at a modification of this sequence. For a jump size j , we are only considering natural numbers divisible by j in our initial sequence. For example, for $j = 3$, our number sequence is `3,6,9,12`, which results in bit sequence `1111010011100`. And are similarly looking for the total number of 1s present before an index i , although rather than using an index, we are specifying the natural number we generate until. However, in this particular problem i **will always be a power of two**

So taking our $j = 3$, and generating for the first $i = 4$ possible natural numbers, the prefix sum string instead looks like:

```
1234455567888
```

as the first 13 digits.

Input

Input will consist of two space separated natural numbers i and j .

Output

Output a single integer, representing the number of 1s in the sequence at or before (1-indexed) natural number i , when using a jump of j .

Constraints

- $1 \leq i \leq 10^{18}$
- i is always a power of 2.
- $2 \leq j \leq 10^5$

Examples

Input

```
2 3
```

Output

```
4
```

Input

```
2 9
```

Output

```
4
```