Problem H. Hash Function

Input file:	standard input
Output file:	standard output
Time limit:	3 seconds
Memory limit:	64 megabytes

A hash function h_n is given, which encrypts the number A, consisting of $2n$ bits, as follows:

Let $A = (a_{2n-1}a_{2n-2}\cdots a_1a_0)_2$, that is, a_i is the i-th bit of the number A.

The number $B = (b_{2n-1}b_{2n-2}\cdots b_1b_0)_2$, also consisting of $2n$ bits, is calculated as follows:

$$b_i = a_i \oplus a_{2i+1}, \text{ for } 0 \le i < n,$$

$$b_i = a_i \oplus a_{4n-2i-2}, \text{ for } n \le i < 2n,$$

where $\oplus$ is bitwise exclusive OR (XOR). In other words,

$$B = A \oplus (a_0a_2\cdots a_{2n-4}a_{2n-2}a_{2n-1}a_{2n-3}\cdots a_3a_1)_2.$$

Next, the number $C = B \oplus \mathrm{RSH}(B)$ is calculated, also consisting of $2n$ bits, where $\mathrm{RSH}(B)$ is a cyclic right shift by 1 bit. In other words,

$$C = B \oplus (b_0b_{2n-1}b_{2n-2}\cdots b_2b_1)_2.$$

Finally, the hash value is calculated as $h_n(A) = 239A + 153C \bmod (2^{2n-1} - 1)$.

For example, let $n = 4$ and $A = 00001101_2 = 13$.

Then, $B = 00001101_2 \oplus 11000010_2 = 11001111_2 = 207$.

Further, $C = 11001111_2 \oplus 11100111_2 = 00101000_2 = 40$.

Finally, $h_4(A) = 239 \times 13 + 153 \times 40 \bmod (2^7 - 1) = 9\,227 \bmod 127 = 83$.

Your goal is to invert this hash function, that is, for given n and H, find A such that $h_n(A) = H$.

Input

You are given two integers n and H $(2 \le n \le 16,\ 0 \le H < 2^{2n-1} - 1)$.

It is guaranteed that for the input there exists A $(0 \le A < 2^{2n})$ such that $h_n(A) = H$.

Output

Print one integer A $(0 \le A < 2^{2n})$ such that $h_n(A) = H$.

If there are several such A — output any.

Example

standard input	standard output
4 83	13