Street Magician

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	256 megabytes

While visiting the farmers' market, Busy Beaver stops to watch a street magician. The magician presents a row of N boxes, holding M-bit nonnegative integers $a_1, a_2, \ldots, a_N$, where $0 \leq a_i < 2^M$ for all $1 \leq i \leq N$.

The magician magically sorts the boxes into non-decreasing order by a series of magic swaps. In a single magic swap, the magician chooses an index i ($1 \leq i < N$) such that the binary representations of a_i and a_{i+1} differ by exactly one bit, and swaps a_i and a_{i+1}.

Watching the performance, Busy Beaver wonders about the limits of this trick. Out of all 2^{MN} possible initial values of $a_1, a_2, \ldots, a_N$ in the boxes, how many of them can be sorted into non-decreasing order using magic swaps? Since this number may be large, Busy Beaver is content with finding it modulo 10^9+7.

Input

The first and only line of input contains two integers N and M ($1 \leq N, M \leq 50$).

Output

Output a single integer: the number of sequences that can be sorted using magic swaps, modulo $10^9 + 7$.

Scoring

There are five subtasks for this problem.

- (10 points): $1 \leq N, M \leq 5$.
- (20 points): $1 \leq M \leq 4$.
- (10 points): $1 \leq M \leq 10$.
- (10 points): $1 \leq M \leq 15$.
- (50 points): No additional constraints.

Examples

standard input	standard output
3 2	44
50 1	898961331
10 10	649370314

Note

In the first sample, one sequence that can be sorted using magic swaps is $[a_1, a_2, a_3] = [3, 1, 2]$, as follows:

1. Choose $i = 1$. Note that $a_1 = 3$ and $a_2 = 1$ differ by exactly one bit, so this is a magic swap. The sequence becomes $[1, 3, 2]$.
2. Choose $i = 2$. Note that $a_2 = 3$ and $a_3 = 2$ differ by exactly one bit, so this is a magic swap. The sequence becomes $[1, 2, 3]$, which is in non-decreasing order.

Out of the $2^{3 \cdot 2} = 64$ initial sequences, 44 of them can be sorted using magic swaps.