

Jellyfish and Hack

Input file: **standard input**
Output file: **standard output**
Time limit: 8 seconds
Memory limit: 512 megabytes

It is well known that quick sort works by randomly selecting a ‘pivot’ element from the array and partitioning the other elements into two sub-arrays, according to whether they are less than or greater than the pivot. But Jellyfish thinks that choosing a random element is just a waste of time, so she always chooses the first element to be the pivot. The time her code needs to run can be calculated by the following pseudocode:

```
function fun(A)
  if A.length > 0
    let L[1 ... L.length] and R[1 ... R.length] be new arrays
    L.length = R.length = 0
    for i = 2 to A.length
      if A[i] < A[1]
        L.length = L.length + 1
        L[L.length] = A[i]
      else
        R.length = R.length + 1
        R[R.length] = A[i]
    return A.length + fun(L) + fun(R)
  else
    return 0
```

Now you want to show her that her code is slow. When the function $\text{fun}(A)$ is greater than or equal to lim , her code will get Time Limit Exceeded. You want to know how many distinct permutations P of $[1, 2, \dots, n]$ satisfies $\text{fun}(P) \geq \text{lim}$. Because the answer may be large, you will only need to find the answer modulo $10^9 + 7$.

Input

The only line of the input contains two integers n and lim ($1 \leq n \leq 200$, $1 \leq \text{lim} \leq 10^9$).

Output

Output the number of different permutations that satisfy the condition modulo $10^9 + 7$.

Examples

standard input	standard output
4 10	8
8 32	1280

Note

In the first example, $P = [1, 4, 2, 3]$ satisfies the condition, because:
 $\text{fun}([1, 4, 2, 3]) = 4 + \text{fun}([4, 2, 3]) = 7 + \text{fun}([2, 3]) = 9 + \text{fun}([3]) = 10$

Do remember to output the answer modulo $10^9 + 7$.