Vera and Sorting

2017 Winter Waterloo Local ACM Contest, Problem D

Vera is very smart and invented a new sorting algorithm. She coded the following Python function to count how many steps her algorithm takes.

```
def steps(array):
  if len(array) == 0:
    return 0
  pivot = array[0]
  count = 0
  lesser = []
  greater = []
  for element in array:        ## looks at each element in the array
    count += 1
    if element < pivot:
      lesser.append(element)   ## e.g. [1,3].append(5) => [1,3,5]
    elif element > pivot:
      greater.append(element)
  return count + steps(lesser) + steps(greater)
```

A *permutation* P is an ordered set of integers $P_1, P_2, \cdots, P_N$, consisting of N distinct positive integers, each of which are at most N. We will call the number N the $size$ of the permutation.

You are given integers N and K.

Help Vera count the number of permutations P of size N such that steps(P) returns the value K. This number could be large, so output it modulo $10^9 + 7$.

Constraints

- $1 \leq N \leq 30$
- $1 \leq K \leq 900$
- N, K are integers

Input Specification

The input will be in the format:

$N\ K$

Output Specification

Output one integer, the number of possible permutations, modulo $10^9 + 7$.

Sample Input 1

```
3 5
```

Sample Output 1

```
2
```

Explanation of Sample Output 1

The 2 valid permutations are $2, 1, 3$ and $2, 3, 1$.

Sample Input 2

```
5 29
```

Sample Output 2

```
0
```

Sample Input 3

```
20 100
```

Sample Output 3

```
262859528
```