Useful Algorithm

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	1024 megabytes

> "Stop learning useless algorithms, go and solve some problems, learn how to use binary search."
>
> – Um_nik

Binary search is a very useful algorithm that has applications in various problems. The following pseudocode demonstrates one implementation of binary search. The function `BINARYSEARCH` takes two parameters A and k, where $A = a_1, a_2, \cdots, a_n$ is a strictly increasing integer sequence of length n, and k is an integer that appears in sequence A. The function returns the index of integer k in sequence A. Note that in the following pseudocode, $\lfloor x \rfloor$ is the largest integer smaller than or equal to x.

Algorithm 1 The Binary Search Algorithm

```
function BinarySearch(A, k)
    l ← 1
    r ← n
    while l < r do
        m ← ⌊(l+r)/2⌋
        if a_m ≥ k then
            r ← m
        else
            l ← m + 1
        end if
    end while
    return l
end function
```

The use of binary search has a prerequisite, which is, the sequence being searched must be sorted. However, in this problem, we will temporarily ignore this constraint and study the performance of binary search on arbitrary sequences.

Given two integers n and k ($1 \le k \le n$), a permutation $P = p_1, p_2, \cdots, p_n$ of n is good if we can correctly find the index of integer k in permutation P with binary search. In other words, let $i =$ `BINARYSEARCH(`P`,` k`)`, then P is good if $p_i = k$.

We now randomly picks a permutation of n with equal probability. Calculate the probability to pick a good permutation.

Input

There are multiple test cases. The first line of the input contains an integer T ($1 \le T \le 10^4$) indicating the number of test cases. For each test case:

The first and only line contains two integers n and k ($1 \le k \le n \le 10^9$).

Output

For each test case, output one line containing one integer, indicating the probability to pick a good permutation modulo $(10^9 + 7)$.

It can be proven that the answer is a rational number $\frac{P}{Q}$. You need to output the value of $PQ^{-1} \bmod (10^9 + 7)$, where Q^{-1} is the integer that satisfies $QQ^{-1} \bmod (10^9 + 7) = 1$.

Example

standard input	standard output
4 3 2 3 1 3 3 1 1	500000004 333333336 666666672 1

Note

For the first sample test case, permutations $\{1, 2, 3\}$, $\{2, 3, 1\}$, and $\{3, 1, 2\}$ are all good. So the answer is $\frac{3}{3!} = \frac{1}{2}$. Since $2 \times 500000004 \bmod (10^9 + 7) = 1$, you should output $1 \times 500000004 \bmod (10^9 + 7) = 500000004$.