Billion ByteMan March

Warning:

Source code size limit is 2048B (half is enough) and time limit could not allow all language to get AC ; I got AC in 1.5s with language C. The **main** difficulty of the problem is to manage efficiently the given precomputed values **and** the memory available, in the given time. Have fun ;-)

The March

Leo invited all his friends to a giant meeting for peace in byteland. All people came in bus which were all full.
Last year, they were thousands of people. As Leo like structured things, he thought to form groups.
Two years ago, all the ways to form homogeneous team with the 4 people (A,B,C,D) were :

{{A,B,C,D}} : one team of 4 (one way),
{{A}, {B}, {C}, {D}} : four 'teams' of 1 (one way more),
{{A,B}, {C,D}} or {{A,C}, {B,D}} or {{A,D}, {B,C}} : two teams of 2 (3 ways more).

for a total of 5 ways. But this year many more people are awaited.
As the answer should not fit in a 64-bit container, you should give your answer modulo M7=1000000007.

Input

The input starts with 2^12 useful precomputed values: factorial(i) MOD M7 for i in [0 ; 2^30[with a step of 2^18, each one on one line.
The input continues with the number T of test cases in a single line.
In each of the next T lines there are two integers : N, K.
N is the quantity of bus that came to the meeting.
K is the common capacity of each bus.

Output

For each test case, your task is to calculate the number of ways people can form homogeneous teams.

Example

Input:
```
1            <--- 0! mod M7
639926614    <--- (2^18)! mod M7
[...]        ( 4093 lines more)
0            <--- (2^30 - 2^18)! mod M7
3
2 2
1 6
2 3
```

Output:

5
27
27

Constraints

0 < T ≤ 300
0 < K ≤ 30000
0 < N ≤ 30000

Uniform-random input in the range.