

Thousands ByteMan 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 only 4 people : A, B, C, D. As Leo likes structured things, he thought to form groups.

All the ways to form homogeneous teams 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 people are awaited.

Input

The input begins 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.

The answer can be a big number and could not fit in a 64bit container.

Example

Input:

```
3
2 2
1 6
2 3
```

Output:

```
5
27
27
```

Explanations

With lower letters, here are 27 ways for 2×3 people.

```
{{a,c,e},{b,d,f}}, {{a,c,d},{b,e,f}}, {{a,b},{c,e},{d,f}}, {{a,f},{b,e},{c,d}},
{{a,b,f},{c,d,e}}, {{a,c},{b,f},{d,e}}, {{a,e},{b,f},{c,d}}, {{a,b},{c,d},{e,f}},
{{a,e},{b,d},{c,f}}, {{a,e,f},{b,c,d}}, {{a,b,e},{c,d,f}}, {{a,d,e},{b,c,f}},
{{a,d},{b,e},{c,f}}, {{a,d},{b,c},{e,f}}, {{a,d},{b,f},{c,e}},
{{a,c,f},{b,d,e}}, {{a,b,c},{d,e,f}}, {{a},{b},{c},{d},{e},{f}},
{{a,b,c,d,e,f}}, {{a,c},{b,d},{e,f}}, {{a,c},{b,e},{d,f}}, {{a,b},{c,f},{d,e}},
{{a,f},{b,d},{c,e}}, {{a,f},{b,c},{d,e}}, {{a,e},{b,c},{d,f}},
{{a,d,f},{b,c,e}}, {{a,b,d},{c,e,f}}
```

Constraints

$0 < T \leq 100$

$0 < K \leq 100$

$0 < N \leq 100$

Uniform-random input in the range. Basic 1/6kB Python code can get AC under 1.5s, around 0.18s using PIKE (quite my first PIKE code), (timings edited 2017-02-11 after compiler changes).