

Reverse Engineering

Sorting is a fascinating topic in computer science and mathematics. Once you get in depth with the sorting algorithms, you will be able to witness some NATURAL beauty. Lots of research has been done in this area. A beautiful but old problem related to sorting is to find the minimum number of adjacent swaps needed to sort some given numbers. But, those golden days of brainstorming are gone. Now almost everyone out there knows the solution to the problem; whether they properly understand sorting or not. Thus, researchers are sad and have decided to come up with a new problem. The problem is given below:

You have N distinct integers. How many ways can you arrange the integers so that the minimum number of adjacent swaps needed to sort them in ascending order is K ?

Researchers cannot yet rate the toughness of this new problem but they believe that solving this problem will require good understanding of sorting algorithms. I do not agree with these researchers and want to join you all to prove them wrong. So I pass you the problem, can you solve it?

Input

Input starts with an integer T (<200000), denoting the number of test cases.

Each case contains two integer N ($1 \leq N \leq 100$) and K ($0 \leq K \leq 10000$).

Output

For each case, print the desired result modulo 1000000007.

Example

Input:

```
3
3 1
10 45
100 56
```

Output:

```
Case 1: 2
Case 2: 1
Case 3: 904490303
```