Problem A. Quicksort

Input file:	*standard input*
Output file:	*standard output*
Time limit:	2 seconds
Memory limit:	512 mebibytes

Bob has written a fake QuickSort implementation as below. Can you figure out, if he randomly chooses a permutation $p = [p_1, p_2, \ldots, p_n]$ obtained from $1, 2, \ldots, n$ with equal probability, the expected number of inversions of p after calling `QuickSort(p, 1, n, k)`?

Functions — a fake QuickSort implementation

```
1: function QuickSort(A, l, r, h)                    ▷ Elements in A would be modified
2:     if h > 1 and l < r then
3:         m ← Partition(A, l, r)
4:         QuickSort(A, l, m − 1, h − 1)
5:         QuickSort(A, m + 1, r, h − 1)
6: function Partition(A, l, r)                       ▷ Elements in A would be modified
7:     i ← l
8:     j ← r
9:     m ← ⌊(l+r)/2⌋
10:    pivot ← A_m
11:    A_m ← A_i
12:    while i < j do
13:        while i < j and A_j ≥ pivot do
14:            j ← j − 1
15:        if i < j then
16:            A_i ← A_j
17:        while i < j and A_i < pivot do
18:            i ← i + 1
19:        if i < j then
20:            A_j ← A_i
21:    A_i ← pivot
22:    return i
```

The number of inversions of a permutation $[p_1, p_2, \ldots, p_n]$ is the number of integer pairs (u, v) such that $1 \le u < v \le n$ and $p_u > p_v$.

To avoid any precision issue, you are asked to report the product of $n!$, the factorial of n, and the expected number in modulo 998244353, which ought to be an integer.

Input

The input contains several test cases. The first line contains an integer T indicating the number of test cases. The following describes all test cases. For each test case:

The only line contains two integers n and k.

- $1 \le T \le 3 \times 10^5$
- $1 \le n, k \le 6000$

Output

For each test case, output a line containing "`Case #x: y`" (without quotes), where `x` is the test case number starting from 1, and `y` is the answer to this test case.

Example

standard input	standard output
5 5 1 5 2 5 3 5 4 5 5	Case #1: 600 Case #2: 240 Case #3: 64 Case #4: 8 Case #5: 0