

Security Plan

Input file: **standard input**
Output file: **standard output**
Time limit: 4 seconds
Memory limit: 512 megabytes

Nikuniku wants to make a security plan for a warehouse. The warehouse can be represented as a grid with n rows and m columns. The upper-left cell is $(1, 1)$, while the bottom-right cell is (n, m) .

A security plan is to choose some cells in the grid and place exactly one camera in each of the selected cells.

When placing a camera in cell (i, j) ($1 \leq i \leq n, 1 \leq j \leq m$), Nikuniku can choose a cell (p, q) ($1 \leq p \leq n, 1 \leq q \leq m$) as its parameter, and a cell (x, y) is protected by this camera if and only if both of the following conditions satisfied:

1. $\min(i, p) \leq x \leq \max(i, p)$
2. $\min(j, q) \leq y \leq \max(j, q)$

Notice that (p, q) is not necessarily different from (i, j) and can't be modified once configured. Different cameras can be configured with different parameters.

A security plan is perfect if and only if the chosen cameras can be configured with appropriate parameters and all of the $n \times m$ cells can be protected by at least one camera.

For a perfect security plan, if Nikuniku can not remove any camera from the plan and change the parameters of the remaining cameras such that the plan remains perfect, we call it a minimal security plan.

Nikuniku wants to know the number of different minimal security plans.

Two plans are different if and only if one cell is chosen in one plan but not in the other, regardless of the parameters.

Input

The first line of the input contains one integer t ($1 \leq t \leq 10^5$) — the number of test cases.

Each test case contains two integers n and m ($1 \leq n, m \leq 10^9$) in one line — the grid size.

Output

Print a single integer for each test case — the answer to the test case modulo 998244353.

Example

standard input	standard output
6	4
2 2	129
4 4	78769
8 8	10
3 3	80
3 5	273
3 7	