Problema O: El ritual de las campanas invertidas

Archivo: `o.cpp` **Juez:** `make o`

Para despertar al antiguo reloj del campus, deben sonar N campanas, cada una exactamente una vez. Sin embargo, un pergamino establece ciertas reglas sobre el orden en que deben tocarse. Algunas reglas pueden repetirse e incluso contradecirse, haciendo imposible completar el ritual.

Tu tarea será contar cuántos ordenamientos diferentes de las campanas, identificadas del 1 al N, cumplen todas las reglas establecidas. Cada regla (a, b) indica que la campana a debe sonar antes que la campana b, aunque no necesariamente de forma consecutiva. Las reglas repetidas no generan nuevas restricciones y, si existen exigencias contradictorias que impiden cualquier ordenamiento, la respuesta debe ser 0.

Entrada

La primera línea contiene dos enteros N y M, correspondientes a la cantidad de campanas y de reglas de precedencia, respectivamente. Las siguientes M líneas contienen dos enteros a y b, indicando que la campana a debe sonar antes que la campana b. Las reglas pueden repetirse. Considere $1 \leq N \leq 10$, $0 \leq M \leq 100$, $1 \leq a, b \leq N$ y $a \neq b$. Si $N = 1$, entonces $M = 0$.

Salida

Un entero que indique la cantidad de ordenamientos diferentes que cumplen todas las reglas de precedencia.

Ejemplos

Ejemplo 1

Entrada	Salida
3 2 1 3 2 3	2

Explicación: Las campanas 1 y 2 deben sonar antes que la campana 3, por lo que esta última debe ocupar la posición final. Las campanas 1 y 2 pueden intercambiarse libremente, dando lugar a los ordenamientos $(1, 2, 3)$ y $(2, 1, 3)$. Por lo tanto, existen 2 ordenamientos válidos.

Ejemplo 2

Entrada	Salida
2 2 1 2 2 1	0

Ejemplo 3

Entrada	Salida
3 0	6