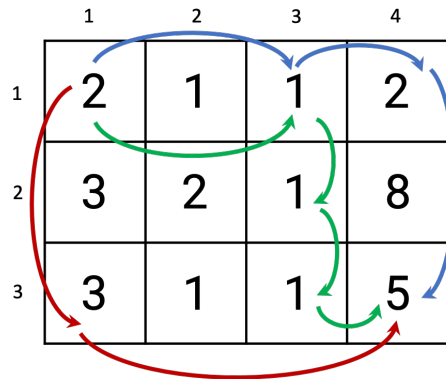


Ejercicio 2. Programación dinámica (3.5 puntos)

Tenemos una matriz de tamaño $n \times m$, donde cada celda (i, j) contiene un número positivo $a_{i,j}$. Desde la celda (i, j) se puede saltar hacia abajo o hacia la derecha, pero la longitud del salto debe ser $a_{i,j}$ (es decir, puedes saltar a la celda $(i + a_{i,j}, j)$ o $(i, j + a_{i,j})$). Y no se puede saltar fuera de la matriz.

Tu tarea consiste en calcular el número de formas distintas de llegar a la celda (n, m) desde la celda $(1, 1)$. Por ejemplo, en la siguiente figura están señalados los 3 caminos diferentes con los que se puede recorrer esta matriz:



Entrada

La entrada está formada por una serie de casos de prueba. Cada caso comienza con una línea con dos números n y m , que representan el tamaño de la matriz ($1 \leq n, m \leq 50$). Las siguientes n líneas contienen cada una m números, los valores de la matriz.

Salida

Para cada caso de prueba se escribirá una línea con el número de formas diferentes de ir desde la celda $(1, 1)$ a la celda (n, m) . Se garantiza que este número nunca será mayor que 10^9 .

Entrada de ejemplo

```
3 4
2 1 1 2
3 2 1 8
3 1 1 5
3 3
1 1 1
1 1 1
1 1 1
```

Salida de ejemplo

```
3
6
```