

Exam

Input file: **standard input**
Output file: **standard output**
Time limit: 1 second
Memory limit: 1024 megabytes

Professor Deokin lectures on algorithms. For the upcoming midterm exam, he decided to add the maximum subarray sum problem taught in his lecture, which is the following problem:

- Given the array $A = [a_1, a_2, \dots, a_n]$, compute the value $MSS(A) = \max_{1 \leq i \leq j \leq n} (\sum_{k=i}^j a_k)$

Deokin wants to force the students to solve the maximum subarray sum problem by hand. For this, he uses the following procedure.

- Prepare a two-dimensional array of integers of size $N \times N$.
- Start from the cell $(1, 1)$, and repeatedly move either right or down to reach the cell (N, N) . If the current cell is (i, j) , he can move right to the cell $(i, j + 1)$ if $j < N$, or move down to the cell $(i + 1, j)$ if $i < N$.
- Write a sequence of length $2N - 1$ which contains the elements of the cells in the order that the cells were visited.

Professor Deokin wants to add an inside joke to the problem so that the students who paid attention to his lectures can have confidence in their answers. He wants to create a sequence where the value of $MSS(A)$ is exactly K . Given a two-dimensional array of size $N \times N$, you need to find the number of different paths in the grid which yield sequences that have a maximum subarray sum value of exactly K .

Input

The first line contains two integers N denoting the size of the two-dimensional array, and K denoting the desired value ($1 \leq N \leq 20, -4 \times 10^{10} \leq K \leq 4 \times 10^{10}$).

The next N lines contain N integers denoting the values in the two-dimensional array. The j -th integer in the i -th line denotes the value $A_{i,j}$. All indices used in the array are 1-indexed ($-10^9 \leq A_{i,j} \leq 10^9$).

Output

Output a single integer denoting the number of different paths in a grid which yield a sequence with a maximum subarray sum value of exactly K .

Examples

standard input	standard output
3 3 1 2 -5 -2 3 0 -1 -1 1	2
4 3 1 -1 1 1 1 1 1 1 1 1 1 1 1 1 -1 1	4