Problem G. Ihsodoih the Charmer

Time limit:	1 seconds
Memory limit:	2048 MB

Ihsodoih is both handsome and popular, being a real charmer whom many people pursue.

She is now playing a game alone while awaiting anyone calling in during a live streaming. The rules of the game are as follows:

Given an $n \times m$ grid, where each cell contains a number $w_{i,j}$. The player draws a single line starting at the $(n, 1)$ cell and ending at any cell on the first row of the grid; that is, the player is allowed to end the line at any one of $(1, 1), (1, 2), \ldots, (1, m)$. Each segment of the line must be either vertical or horizontal, segments cannot overlap with each other, each cell can be visited at most once, and segments cannot be drawn downward.

The score is the sum of the values of each cell visited by the line. Your task is to assist Ihsodoih in computing the maximum score she can achieve, allowing her to boost herself in case of any calling in though there is none so far.

Input

The first line of the input contains two positive integers n, m — the size of the grid.

The ith line of the next n lines contains m integers $w_{i,1}, w_{i,2}, \ldots, w_{i,m}$ — the value on each cell.

- $1 \leq n \times m \leq 200000$
- $-10^9 \leq w_{i,j} \leq 10^9$ for $1 \leq i \leq n$ and $1 \leq j \leq m$

Output

You should print one integer representing the maximum value Ihsodoih can achieve.

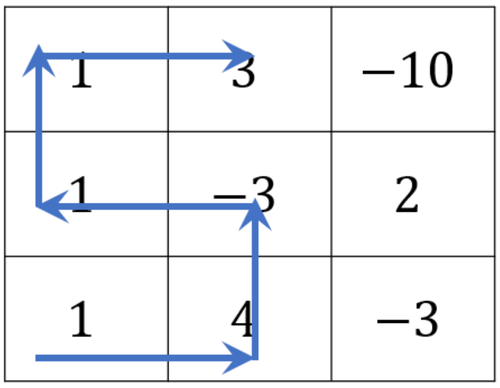
Examples

Standard Input	Standard Output

3 3
1 3 -10
1 -3 2
1 4 -3
	7

2 2
1 -1
1 -1
	2

The best line of the example 1 is showed in the following figure. $1 + 4 + (-3) + 1 + 1 + 3 = 7$.



This page is intentionally left blank

Problem H. White

Time limit:	2 seconds
Memory limit:	2048 MB

Two good things doesn't necessarily lead to good result, as 1 OP 1 doesn't necessarily equal to 2 if the operation OP is not $+$.

There are n photos in an album S. The ith of the photos represents a good memory with happiness value w_i.

For each subset of $T \subseteq S$, let

$$f(T) = \left(\sum_{i \in T} w_i\right) \bmod 10^9+7 \text{ and } F = \bigoplus_{T \subseteq S} f(T)$$

where $f(\emptyset) = 0$ and

$$g(T) = \left(\prod_{i \in T} w_i\right) \bmod 10^9+7 \text{ and } G = \bigoplus_{T \subseteq S} g(T)$$

where $g(\emptyset) = 1$.

Your task is to compute $F \oplus G$, which is the eventual happiness value.

Note that $\oplus$ is the xor operation, which is denoted by ^ in C++ language.

Input

The first line of the input contains an integer n — the size of S.

The second line of the input contains n integers $w_1, w_2, \ldots, w_n$ — the happiness value of each photo.

- $1 \le n \le 26$
- $1 \le w_i \le 10^9$ for $1 \le i \le n$

Output

Print an integer representing $F \oplus G$ — the eventual happiness value.

Examples

Standard Input	Standard Output
3 1 2 3	4
1 100000	1

In example 1,

$$F = ((0) \oplus (1) \oplus (2) \oplus (3) \oplus (1+2) \oplus (1+3) \oplus (2+3) \oplus (1+2+3)) \bmod 10^9+7 = 4$$

$$G = ((1) \oplus (1) \oplus (2) \oplus (3) \oplus (1 \times 2) \oplus (1 \times 3) \oplus (2 \times 3) \oplus (1 \times 2 \times 3)) \bmod 10^9+7 = 0$$

Hence, $F \oplus G = 4$.

This page is intentionally left blank

Problem I. CPCP

Time limit:	1 seconds
Memory limit:	512 MB

CP (Competitive Penguins) is a virtual YouTuber company that owns two popular penguin groups, NYCU and NTHU. Both of them have n charming penguins.

To make the penguins more attractive, the company is organizing a new event called CPCP (Competitive Penguins Couple). In this event, pairs will be formed, each consisting of one penguin from NYCU and one from NTHU. Each penguin can be in at most one pair. The company believes these couples will increase the popularity of the groups.

However, there's a catch. Though the penguins seems to be friends with each other, penguins from different groups don't get along well in private actually. Each penguin in NYCU has a tolerance value a_i, and each penguin in NTHU has a tolerance value b_i. If you pair the ith penguin from NYCU with the jth penguin from NTHU, the pair will have an anger value of $a_i \times b_j$.

The company can handle conflicts between penguins if the sum of the anger values of the pairs doesn't exceed k. Given k, the task is to help the CP company find the maximum number of pairs that can be formed while keeping the total anger value below or equal to k.

Input

The first line of the input lines contains three integers n, k — the number of penguins in NYCU/NTHU, and the upper bound of the total anger value the company can deal with.

The second line contains n integers $a_1, a_2, \ldots, a_n$ — the tolerance value of the penguins in NYCU.

The third line contains n integers $b_1, b_2, \ldots, b_n$ — the tolerance value of the penguins in NTHU.

- $1 \le n \le 10^5$
- $0 \le k \le 10^{18}$
- $0 \le a_i, b_j \le 10^6$ for $i = 1, 2, \ldots, n$ and $j = 1, 2, \ldots, n$

Output

You should print one integer representing the maximum number of pairs that can be formed while keeping the total anger value below or equal to k.

Examples

Standard Input	Standard Output

3 14
1 2 4
3 1 5
	2

2 20
2 2
3 3
	2

In example 1, we can form the two pairs (1st of NYCU, 3rd of NTHU), (2nd of NYCU, 1st of NTHU). The eventual anger value will be $1 \times 5 + 2 \times 3 = 11 \le 14$. Hence, forming two pairs is possible. However, one can verify that there is no way to form three pairs while keeping the total anger value not exceed 14.

Note

Actually, the company can force the penguins to get along with each other by a horrible technique called FFT (Frozen Frustrating Tolerance). Nevertheless, it hurts the penguins so that we should prevent from using it.

This page is intentionally left blank

Problem J. Social Network

Time limit:	3 seconds
Memory limit:	512 MB

The relationship between individuals need not be mutually acknowledged. The statement ”A regards B as a friend” is not necessarily reciprocated by ”B regards A as a friend.” This asymmetry has unsettled the problem setter, and thus, we avoid it in this problem.

In this problem, a matrix A is termed a daydreaming matrix if it satisfies the following three conditions:

- A is binary, meaning each entry is either 0 or 1.
- A is symmetric, implying $A_{i,j} = A_{j,i}$ for $i = 0, 1, \ldots, n-1$ and $j = 0, 1, \ldots, n-1$.
- $Tr(A) = n$, indicating $A_{i,i} = 1$ for $i = 0, 1, \ldots, n-1$.

For two $n \times n$ daydreaming matrices A and B, the operation $\text{Sleep}(A, B)$ results in an $n \times n$ daydreaming matrix C defined as follows:

$$C_{i,j} = \vee_{k=0}^{n-1}(A_{i,k} \wedge B_{k,j}) = (A_{i,0} \wedge B_{0,j}) \vee (A_{i,1} \wedge B_{1,j}) \vee \cdots \vee (A_{i,n-1} \wedge B_{n-1,j})$$

Here, $\wedge$ represents the logical AND operation, and $\vee$ represents the logical OR operation. For two numbers $a, b \in \{0, 1\}$, the operations are defined as:

$$a \wedge b = \begin{cases} 1, & \text{If } a = 1 \text{ and } b = 1 \\ 0, & \text{Otherwise} \end{cases}$$

$$a \vee b = \begin{cases} 0, & \text{If } a = 0 \text{ and } b = 0 \\ 1, & \text{Otherwise} \end{cases}$$

$\text{Sleep}(A, B)$ can be interpreted as the matrix multiplication of A and B, with addition replaced by $\vee$ and multiplication replaced by $\wedge$.

For a daydreaming matrix M and a non-negative integer d, the function $\text{Popular}(M, d)$ is defined as $\text{Sleep}(\text{Popular}(M, d-1), M)$, and $\text{Popular}(M, 1) = M$.

Given an $n \times n$ daydreaming matrix A, your task is to determine the number of entries equal to 1 in $\text{Popular}(A, n)$.

Input

The first line contains an integer n ($1 \leq n \leq 1500$) representing the size of matrix A.

The subsequent $n-1$ lines each represent a row of the matrix A. The ith line contains $n-i$ integers $A_{i,i+1}, A_{i,i+2}, \ldots, A_{i,n-1}$, denoting the right-hand side values of the ith row of A.

Output

You should print a number representing the number of entries equal to 1 in $\text{Popular}(A, n)$.

Example

Standard Input	Standard Output

3
1 1
1
	9

4
1 0 0
0 0
1
	8

In example 1, the result is

$$\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

In example 2, the result is

$$\begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 \end{bmatrix}$$