Matrix Reduction (`matred`)

Alex and Andrei are playing a game on an $N \times M$ matrix filled with non-negative integers. The rows of the matrix are numbered from 1 to N (from top to bottom), and the columns are numbered from 1 to M (from left to right).

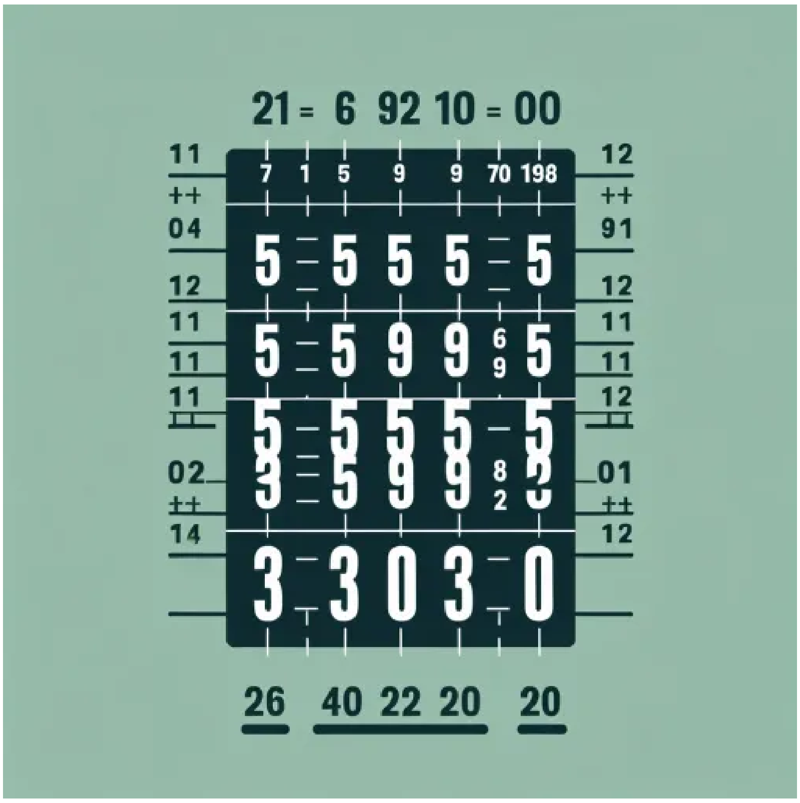

Figure 1: A matrix of numbers.

Alex is allowed to perform the following operation any number of times:

- Choose a horizontal or vertical 1×3 subrectangle.
- Select an integer V.
- Add V to all numbers in the chosen subrectangle, ensuring that no number becomes negative.

Andrei believes that Alex **cannot** make all the numbers in the matrix zero within at most $2 \cdot N \cdot M$ operations.

Alex is determined to prove him wrong – but since he hasn't learned addition at school yet, he needs your help. Can you determine whether Alex can reduce the entire matrix to zero within the given limit?

> ☞ Among the attachments of this task you may find a template file `matred.*` with a sample incomplete implementation.

Input

The first line of the input contains two integers: N and M.

Each of the next N lines contains M non-negative integers, representing the elements of the matrix.

Output

First, print "YES" if it is possible to make all $A_{i,j}$ equal to zero within at most $2 \cdot N \cdot M$ operations. Otherwise, print "NO".

If the answer is "YES", proceed with the following output:

- Print an integer R, the number of operations perfomed.
- Print R lines, each describing an operation in the format:
  X_1 Y_1 X_2 Y_2 V
  where (X_1,Y_1) are the coordinates of the top-left corner of the chosen subrectangle, (X_2,Y_2) are the coordinates of the bottom-right corner of the subrectangle (that is, $X_1 \leq X_2$ and $Y_1 \leq Y_2$), and V is the integer added to all elements within the subrectangle.

If there are multiple valid solutions, you can print any of them.

Constraints

- $3 \leq N \leq 500$.
- $3 \leq M \leq 500$.
- $0 \leq A_{i,j} \leq 1000$ for each $i = 1 \ldots N$ and $j = 1 \ldots M$.
- $-10^9 \leq V \leq 10^9$.

Scoring

Your program will be tested against several test cases grouped in subtasks. In order to obtain the score of a subtask, your program needs to correctly solve all of its test cases.

– **Subtask 1** (0 points) Examples.

– **Subtask 2** (14 points) $N \leq 6$, $M \leq 6$.

– **Subtask 3** (25 points) $N = 3$ or $M = 3$.

– **Subtask 4** (7 points) All numbers $A_{i,j}$ are equal.

– **Subtask 5** (54 points) No additional limitations.

Examples

input	output
3 3 1 5 1 2 6 2 8 12 8	YES 6 1 1 1 3 8 2 1 2 3 7 3 1 3 3 1 1 1 3 1 -9 1 2 3 2 -13 1 3 3 3 -9
3 3 1 5 0 2 6 2 8 12 8	NO

Explanation

In the **first sample case**, there are multiple valid solutions, with the sample output being one of them.

In the **second sample case** it can be proven that no sequence of at most $2 \cdot N \cdot M = 18$ operations will lead to a matrix containing only zeros.