B. Progressive Square

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

A *progressive square* of size n is an $n \times n$ matrix. Maxim chooses three integers $a_{1,1}$, c, and d and constructs a *progressive square* according to the following rules:

$$a_{i+1,j} = a_{i,j} + c$$

$$a_{i,j+1} = a_{i,j} + d$$

For example, if $n = 3$, $a_{1,1} = 1$, $c = 2$, and $d = 3$, then the *progressive square* looks as follows:

$$\begin{pmatrix} 1 & 4 & 7 \\ 3 & 6 & 9 \\ 5 & 8 & 11 \end{pmatrix}$$

Last month Maxim constructed a *progressive square* and remembered the values of n, c, and d. Recently, he found an array b of n^2 integers in random order and wants to make sure that these elements are the elements of **that specific** square.

It can be shown that for any values of n, $a_{1,1}$, c, and d, there exists exactly one *progressive square* that satisfies all the rules.

Input

The first line contains an integer t ($1 \le t \le 10^4$) — the number of test cases.

The first line of each test case contains three integers n, c, and d ($2 \le n \le 500$, $1 \le c, d \le 10^6$) — the size of the square and the values of c and d as described in the statement.

The second line of each test case contains $n \cdot n$ integers $b_1, b_2, \ldots, b_{n \cdot n}$ ($1 \le b_i \le 10^9$) — the elements found by Maxim.

It is guaranteed that the sum of n^2 over all test cases does not exceed $25 \cdot 10^4$.

Output

For each test case, output "YES" in a separate line if a *progressive square* for the given n, c, and d can be constructed from the array elements a, otherwise output "NO".

You can output each letter in any case (lowercase or uppercase). For example, the strings "yEs", "yes", "Yes", and "YES" will be accepted as a positive answer.

Example

input

```
5
3 2 3
3 9 6 5 7 1 10 4 8
3 2 3
3 9 6 5 7 1 11 4 8
2 100 100
400 300 400 500
3 2 3
3 9 6 6 5 1 11 4 8
4 4 4
15 27 7 19 23 23 11 15 7 3 19 23 11 15 11 15
```

output

```
NO
YES
```

```
YES
NO
NO
```