

6067 Bit Magic

Yesterday, my teacher taught me about bit operators: and (&), or (|), xor (^). I generated a number table $a[N]$, and wrote a program to calculate the matrix table $b[N][N]$ using three kinds of bit operator. I thought my achievement would get teacher's attention.

The key function is the code showed below.

```
void calculate(int a[N], int b[N][N]) {
    for (int i = 0; i < N; ++i) {
        for (int j = 0; j < N; ++j) {
            if (i == j) b[i][j] = 0;
            else if (i % 2 == 1 && j % 2 == 1) b[i][j] = a[i] | a[j];
            else if (i % 2 == 0 && j % 2 == 0) b[i][j] = a[i] & a[j];
            else b[i][j] = a[i] ^ a[j];
        }
    }
}
```

There is no doubt that my teacher raised lots of interests in my work and was surprised to my talented programming skills. After deeply thinking, he came up with another problem: if we have the matrix table $b[N][N]$ at first, can you check whether corresponding number table $a[N]$ exists?

Input

There are multiple test cases.

For each test case, the first line contains an integer N , indicating the size of the matrix. ($1 \leq N \leq 500$).

The next N lines, each line contains N integers, the j -th integer in i -th line indicating the element $b[i][j]$ of matrix. ($0 \leq b[i][j] \leq 2^{31} - 1$)

Output

For each test case, output 'YES' if corresponding number table $a[N]$ exists; otherwise output 'NO'.

Sample Input

```
2
0 4
4 0
3
0 1 24
1 0 86
24 86 0
```

Sample Output

```
YES
NO
```