Output :

```
YES
YES
NO
NO
```

Note:

9	4	7	8
6	5	0	3
11	4	2	5
10	3	8	14

In the **first** query the answer is YES because both sum are equal to 0 (there are no elements in the upper or lower part of the diagonal).
In the **second** query the answer is YES because: 9 + 5 = 14. (m[1][1] + m[2][2] = 14 and m[4][4] = 14)
In the **third** query the answer is NO because: m[2][1] = 6 $\neq$ m[4][3] = 8.
In the **fourth** query the answer is NO bcause: m[3][1] = 11 $\neq$ 0 (there are no elements in the lower part of the diagonal).

Important

Since the input/output for this problem can be very large, you need to use **fast input/output** to optimize performance, it is listed below how to do so for the available programming languages:

- **C++**:

Add these lines to your code:

```
ios::sync_with_stdio(false);
cin.tie(NULL);
cout.tie(NULL);
```

Instead of using **endl** for a new line, use **"\n"**

- **Python**: Import sys library

```
//input
sys.stdin.readline()
//output
sys.stdout.write("YES\n")
sys.stdout.write("NO\n")
```

- **Java**: Use the buffered reader and buffered writer classes

Import these java classes: java.io.BufferedReader; java.io.InputStreamReader; java.io.BufferedWriter; import java.io.OutputStreamWriter; java.util.StringTokenizer;

```
BufferedReader reader = new BufferedReader(new InputStreamReader(System.in));
BufferedWriter writer = new BufferedWriter(new OutputStreamWriter(System.out));
```

For each line, create a string tokenizer:

```
StringTokenizer tokenizer = new StringTokenizer(reader.readLine());
```

To get a value use:

```
tokenizer.nextToken();
Integer.parseInt(tokenizer.nextToken()); //for example
```