

# 2628. JSON Deep Equal

## Description

Given two values `o1` and `o2` , return a boolean value indicating whether two values, `o1` and `o2` , are **deeply equal** .

For two values to be **deeply equal** , the following conditions must be met:

- If both values are primitive types, they are **deeply equal** if they pass the `===` equality check.
- If both values are arrays, they are **deeply equal** if they have the same elements in the same order, and each element is also **deeply equal** according to these conditions.
- If both values are objects, they are **deeply equal** if they have the same keys, and the associated values for each key are also **deeply equal** according to these conditions.

You may assume both values are the output of `JSON.parse` . In other words, they are valid JSON.

Please solve it without using lodash's `_.isEqual()` function

### Example 1:

**Input:** `o1 = {"x":1,"y":2}, o2 = {"x":1,"y":2}`  
**Output:** `true`  
**Explanation:** The keys and values match exactly.

### Example 2:

**Input:** `o1 = {"y":2,"x":1}, o2 = {"x":1,"y":2}`  
**Output:** `true`  
**Explanation:** Although the keys are in a different order, they still match exactly.

### Example 3:

**Input:** `o1 = {"x":null,"L":[1,2,3]}, o2 = {"x":null,"L":["1","2","3"]}`  
**Output:** `false`  
**Explanation:** The array of numbers is different from the array of strings.

### Example 4:

**Input:** `o1 = true, o2 = false`  
**Output:** `false`  
**Explanation:** `true !== false`

### Constraints:

- `1 <= JSON.stringify(o1).length <= 105`
- `1 <= JSON.stringify(o2).length <= 105`
- `maxNestingDepth <= 1000`

