

# 2755. Deep Merge of Two Objects

## Description

Given two values `obj1` and `obj2` , return a **deepmerged** value.

Values should be **deepmerged** according to these rules:

- If the two values are objects, the resulting object should have all the keys that exist on either object. If a key belongs to both objects, **deepmerge** the two associated values. Otherwise, add the key-value pair to the resulting object.
- If the two values are arrays, the resulting array should be the same length as the longer array. Apply the same logic as you would with objects, but treat the indices as keys.
- Otherwise the resulting value is `obj2` .

You can assume `obj1` and `obj2` are the output of `JSON.parse()` .

### Example 1:

**Input:** `obj1 = {"a": 1, "c": 3}, obj2 = {"a": 2, "b": 2}`

**Output:** `{"a": 2, "c": 3, "b": 2}`

**Explanation:** The value of `obj1["a"]` changed to 2 because if both objects have the same key and their value is not an array or object then we change the `obj1` value to the `obj2` value. Key "b" with value was added to `obj1` as it doesn't exist in `obj1`.

### Example 2:

**Input:** `obj1 = [{}, 2, 3], obj2 = [[], 5]`

**Output:** `[[], 5, 3]`

**Explanation:** `result[0] = obj2[0]` because `obj1[0]` and `obj2[0]` have different types. `result[2] = obj1[2]` because `obj2[2]` does not exist.

### Example 3:

**Input:**  
`obj1 = {"a": 1, "b": {"c": [1 , [2, 7], 5], "d": 2}},`  
`obj2 = {"a": 1, "b": {"c": [6, [6], [9]], "e": 3}}`

**Output:** `{"a": 1, "b": {"c": [6, [6, 7], [9]], "d": 2, "e": 3}}`

**Explanation:**  
Arrays `obj1["b"]["c"]` and `obj2["b"]["c"]` have been merged in way that `obj2` values overwrite `obj1` values deeply only if they are not arrays or objects.  
`obj2["b"]["c"]` has key "e" that `obj1` doesn't have so it's added to `obj1`.

### Example 4:

**Input:** `obj1 = true, obj2 = null`

**Output:** `null`

### Constraints:

- `obj1` and `obj2` are valid JSON values
- `1 <= JSON.stringify(obj1).length <= 5 * 105`
- `1 <= JSON.stringify(obj2).length <= 5 * 105`

