# 3117. Minimum Sum of Values by Dividing Array

## Description

You are given two arrays `nums` and `andValues` of length `n` and `m` respectively.

The **value** of an array is equal to the **last** element of that array.

You have to divide `nums` into `m` **disjoint contiguous** subarrays such that for the $i^{th}$ subarray $[l_i, r_i]$, the bitwise `AND` of the subarray elements is equal to `andValues[i]`, in other words, $nums[l_i] \& nums[l_i + 1] \& \dots \& nums[r_i] == andValues[i]$ for all `1 <= i <= m`, where `&` represents the bitwise `AND` operator.

Return *the* ***minimum*** *possible sum of the* ***values*** *of the* `m` *subarrays* `nums` *is divided into*. *If it is not possible to divide* `nums` *into* `m` *subarrays satisfying these conditions, return* `-1`.

### Example 1:

**Input:** nums = [1,4,3,3,2], andValues = [0,3,3,2]

**Output:** 12

**Explanation:**

The only possible way to divide `nums` is:

1. `[1,4]` as `1 & 4 == 0`.
2. `[3]` as the bitwise `AND` of a single element subarray is that element itself.
3. `[3]` as the bitwise `AND` of a single element subarray is that element itself.
4. `[2]` as the bitwise `AND` of a single element subarray is that element itself.

The sum of the values for these subarrays is `4 + 3 + 3 + 2 = 12`.

### Example 2:

**Input:** nums = [2,3,5,7,7,7,5], andValues = [0,7,5]

**Output:** 17

**Explanation:**

There are three ways to divide `nums`:

1. `[[2,3,5],[7,7,7],[5]]` with the sum of the values `5 + 7 + 5 == 17`.
2. `[[2,3,5,7],[7,7],[5]]` with the sum of the values `7 + 7 + 5 == 19`.
3. `[[2,3,5,7,7],[7],[5]]` with the sum of the values `7 + 7 + 5 == 19`.

The minimum possible sum of the values is `17`.

### Example 3:

**Input:** nums = [1,2,3,4], andValues = [2]

**Output:** -1

**Explanation:**

The bitwise `AND` of the entire array `nums` is `0`. As there is no possible way to divide `nums` into a single subarray to have the bitwise `AND` of elements `2`, return `-1`.

### Constraints:

- $1 <= n == nums.length <= 10^4$
- $1 <= m == andValues.length <= min(n, 10)$
- $1 <= nums[i] < 10^5$
- $0 <= andValues[j] < 10^5$