There is a garden of n flowers, and each flower has an integer beauty value. The flowers are arranged in a line. You are given an integer array flowers of size n and each flowers[i] represents the beauty of the ith flower.

A garden is **valid** if it meets these conditions:

* The garden has at least two flowers.
* The first and the last flower of the garden have the same beauty value.

As the appointed gardener, you have the ability to **remove** any (possibly none) flowers from the garden. You want to remove flowers in a way that makes the remaining garden **valid**. The beauty of the garden is the sum of the beauty of all the remaining flowers.

Return the maximum possible beauty of some **valid** garden after you have removed any (possibly none) flowers.

**Example 1:**

**Input:** flowers = [1,2,3,1,2]

**Output:** 8

**Explanation:** You can produce the valid garden [2,3,1,2] to have a total beauty of 2 + 3 + 1 + 2 = 8.

**Example 2:**

**Input:** flowers = [100,1,1,-3,1]

**Output:** 3

**Explanation:** You can produce the valid garden [1,1,1] to have a total beauty of 1 + 1 + 1 = 3.

**Example 3:**

**Input:** flowers = [-1,-2,0,-1]

**Output:** -2

**Explanation:** You can produce the valid garden [-1,-1] to have a total beauty of -1 + -1 = -2.

**Constraints:**

* 2 <= flowers.length <= 105
* -104 <= flowers[i] <= 104
* It is possible to create a valid garden by removing some (possibly none) flowers.