A zero-indexed array A consisting of N integers is given. The *dominator* of array A is the value that occurs in more than half of the elements of A.

For example, consider array A such that

A[0] = 3 A[1] = 4 A[2] = 3

A[3] = 2 A[4] = 3 A[5] = -1

A[6] = 3 A[7] = 3

The dominator of A is 3 because it occurs in 5 out of 8 elements of A (namely in those with indices 0, 2, 4, 6 and 7) and 5 is more than a half of 8.

Write a function

class Solution { public int solution(int[] A); }

that, given a zero-indexed array A consisting of N integers, returns index of any element of array A in which the dominator of A occurs. The function should return −1 if array A does not have a dominator.

Assume that:

* N is an integer within the range [0..100,000];
* each element of array A is an integer within the range [−2,147,483,648..2,147,483,647].

For example, given array A such that

A[0] = 3 A[1] = 4 A[2] = 3

A[3] = 2 A[4] = 3 A[5] = -1

A[6] = 3 A[7] = 3

the function may return 0, 2, 4, 6 or 7, as explained above.

Complexity:

* expected worst-case time complexity is O(N);
* expected worst-case space complexity is O(1), beyond input storage (not counting the storage required for input arguments).

Elements of input arrays can be modified.