

## Problem G. Be Positive

**Time Limit** 1000 ms

**Mem Limit** 262144 kB

Given an array  $a$  of  $n$  elements, where each element is equal to  $-1$ ,  $0$ , or  $1$ . In one operation, you can choose an index  $i$  and increase  $a_i$  by  $1$  (that is, assign  $a_i := a_i + 1$ ). Operations can be performed any number of times, choosing any indices.

The goal is to make the product of all elements in the array strictly positive with the minimum number of operations, that is,  $a_1 \cdot a_2 \cdot a_3 \cdot \dots \cdot a_n > 0$ . Find the minimum number of operations.

It is guaranteed that this is always possible.

### Input

Each test consists of several test cases.

The first line contains one integer  $t$  ( $1 \leq t \leq 10^4$ ) — the number of test cases. The description of the test cases follows.

The first line of each test case contains one integer  $n$  ( $1 \leq n \leq 8$ ) — the length of the array  $a$ .

The second line contains  $n$  integers  $a_1, a_2, \dots, a_n$ , where  $-1 \leq a_i \leq 1$  — the elements of the array  $a$ .

### Output

For each test case, output one integer — the minimum number of operations required to make the product of the elements in the array strictly positive.

### Examples

Input	Output
3 3 -1 0 1 4 -1 -1 0 1 5 -1 -1 -1 0 0	3 1 4

## Note

In the first test case: from  $[-1, 0, 1]$ , you can obtain  $[1, 1, 1]$  in 3 operations.

In the second test case: it is enough to perform  $0 \rightarrow 1$  (1 operation). In the resulting array  $a = [-1, -1, 1, 1]$ , the product of all elements is 1.

In the third test case: turning two zeros into ones (2 operations), and one  $-1$  into 1 (another 2 operations), for a total of 4.