B. Make Product Equal One

time limit per test

1 second

memory limit per test

256 megabytes

input

standard input

output

standard output

You are given n numbers $a_1, a_2, \dots, a_n$. With a cost of one coin you can perform the following operation:

Choose one of these numbers and add or subtract 1 from it.

In particular, we can apply this operation to the same number several times.

We want to make the product of all these numbers equal to 1, in other words, we want $a_1 \cdot a_2 \dots \cdot a_n = 1$.

For example, for $n = 3$ and numbers $[1, -3, 0]$ we can make product equal to 1 in 3 coins: add 1 to second element, add 1 to second element again, subtract 1 from third element, so that array becomes $[1, -1, -1]$. And $1 \cdot (-1) \cdot (-1) = 1$.

What is the minimum cost we will have to pay to do that?

Input

The first line contains a single integer n ($1 \le n \le 10^5$) — the number of numbers.

The second line contains n integers $a_1, a_2, \dots, a_n$ ($-10^9 \le a_i \le 10^9$) — the numbers.

Output

Output a single number — the minimal number of coins you need to pay to make the product equal to 1.

Examples

input

Copy

```
2
-1 1
```

output

Copy

```
2
```

input

Copy

```
4
0 0 0 0
```

output

Copy

```
4
```

input

Copy

```
5
-5 -3 5 3 0
```

output

Copy

```
13
```

Note

In the first example, you can change 1 to -1 or -1 to 1 in 2 coins.

In the second example, you have to apply at least 4 operations for the product not to be 0.

In the third example, you can change -5 to -1 in 4 coins, -3 to -1 in 2 coins, 5 to 1 in 4coins, 3 to 1 in 2 coins, 0 to 1 in 1 coin.