C. Building Permutation

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Permutation p *is an ordered set of integers* $p_1, p_2, \ldots, p_n$, *consisting of* n *distinct positive integers, each of them doesn't exceed* n. *We'll denote the* i*-th element of permutation* p *as* p_i. *We'll call number* n *the size or the length of permutation* $p_1, p_2, \ldots, p_n$.

You have a sequence of integers $a_1, a_2, \ldots, a_n$. In one move, you are allowed to decrease or increase any number by one. Count the minimum number of moves, needed to build a permutation from this sequence.

Input

The first line contains integer n ($1 \le n \le 3 \cdot 10^5$) — the size of the sought permutation. The second line contains n integers $a_1, a_2, \ldots, a_n$ ($-10^9 \le a_i \le 10^9$).

Output

Print a single number — the minimum number of moves.

Please, do not use the %lld specifier to read or write 64-bit integers in C++. It is preferred to use the cin, cout streams or the %I64d specifier.

Examples

input
2 3 0
output
2

input
3 -1 -1 2
output
6

Note

In the first sample you should decrease the first number by one and then increase the second number by one. The resulting permutation is $(2, 1)$.

In the second sample you need 6 moves to build permutation $(1, 3, 2)$.