

## A. Pupils Redistribution

time limit per test: 1 second

memory limit per test: 256 megabytes

input: standard input

output: standard output

In Berland each high school student is characterized by *academic performance* — integer value between 1 and 5.

In high school `0xFF` there are two groups of pupils: the group *A* and the group *B*. Each group consists of exactly  $n$  students. An academic performance of each student is known — integer value between 1 and 5.

The school director wants to redistribute students between groups so that each of the two groups has the same number of students whose academic performance is equal to 1, the same number of students whose academic performance is 2 and so on. In other words, the purpose of the school director is to change the composition of groups, so that for each value of academic performance the numbers of students in both groups are equal.

To achieve this, there is a plan to produce a series of exchanges of students between groups. During the single exchange the director selects one student from the class *A* and one student of class *B*. After that, they both change their groups.

Print the least number of exchanges, in order to achieve the desired equal numbers of students for each academic performance.

### Input

The first line of the input contains integer number  $n$  ( $1 \leq n \leq 100$ ) — number of students in both groups.

The second line contains sequence of integer numbers  $a_1, a_2, \dots, a_n$  ( $1 \leq a_i \leq 5$ ), where  $a_i$  is academic performance of the  $i$ -th student of the group *A*.

The third line contains sequence of integer numbers  $b_1, b_2, \dots, b_n$  ( $1 \leq b_i \leq 5$ ), where  $b_i$  is academic performance of the  $i$ -th student of the group *B*.

### Output

Print the required minimum number of exchanges or  $-1$ , if the desired distribution of students can not be obtained.

### Examples

input
4 5 4 4 4 5 5 4 5
output
1
input
6 1 1 1 1 1 1 5 5 5 5 5 5
output
3
input
1 5 3

output

-1

input

9

3 2 5 5 2 3 3 3 2

4 1 4 1 1 2 4 4 1

output

4