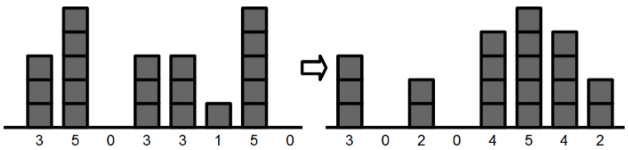
beCP
2023
Task 1.1: Stack city (stackcity)

Author: Damien Leroy Preparation: Damien Leroy and Robin Jadoul
Time limit: 2 s Memory limit: 64 MB

A young architect with a vivid imagination designed a city where the houses consist of swappable, cube shaped modules, allowing the city to be continually transformed. You are the programmer of the robot that's responsible for the movement of the modules, according to the mood of the inhabitants.

The city is represented as a list of columns, each consisting of a stack with some amount of modules.



Starting from the initial configuration and the requested placement of the modules, your program should determine the minimum number of movements necessary to go from the initial configuration to the requested one. A movement consists of moving a single module to the next or the previous column.

Input

Your program receives the input in the following format:

- The first line contains an integer W: the number of columns in the city.
- The second line describes the initial configuration as W nonnegative integers, separated by spaces. The ith number is the amount of modules in the ith column.
- The third line describes the requested final configuration with W nonnegative integers, in the same manner.

Output

The output of your program is a single number: the minimal number of movements needed.

General limits

- $1 \leq W \leq 100\,000$, the number of columns;
- $0 \leq m_i \leq 500\,000\,000$, the number of modules in column i;

Additional constraints

Subtask	Points	Constraints
A	32	$W \leq 100$, $m_i \leq 100$, the modules only have to move to the right
B	19	$W \leq 100$, $m_i \leq 100$
C	49	No additional constraint

Example 1

sample1.in
```
8
3 5 0 3 3 1 5 0
3 0 2 0 4 5 4 2
```

sample1.out
```
22
```

This corresponds to the earlier illustration. The city consists of 8 columns with a total of 20 modules. It's possible to move from the initial to the final configuration in 22 movements.

Example 2

sample2.in
```
10
0 0 0 0 0 0 0 0 0 500000000
```

sample2.out
```
4500000000
```

Remark: The number of movements can be very big, too big for a regular *integer* datatype. Use variables of the type `long long`.