# Problem 2 – Bridges

You’ll be given two sequences of **natural numbers** – we’ll call the sequences **north** and **south**. A sequence will be given on a single line; numbers will be separated by a **single space**.

A bridge can be built between the north sequence and the south sequence by **connecting equal numbers** – 1 with 1, 5 with 5, etc. Your task is to **find the maximum number of bridges that can be built without crossing any bridges**. Print the result as a single number on the console.

### Input

* The input data should be read from the console. It will consist of exactly two lines:
* On the first line you’ll be given the north sequence of numbers.
* On the second line you’ll be given the south sequence of numbers.
* All numbers will be separated from each other by a single space.
* The input data will always be valid and in the format described. There is no need to check it explicitly.

### Output

* The output should be printed on the console. It should consist of only one line.
* On the only output line print a single number – the maximum number of non-crossing bridges that can be built between the north and south sequences.

### Constraints

* Each sequence will contain between 1 and 50 numbers **with possible duplicates**. **Sequences may be of different length.**
* The numbers in the sequences will be in the range [0 … 10].
* Allowed working time for your program: 0.1 seconds. Allowed memory: 16 MB.

### Examples

|  |  |  |
| --- | --- | --- |
| **Input** | **Output** | **Comments** |
| 1 2 3 4  1 2 3 4 | 4 | Each of the four numbers in the north sequence has a counterpart in the south sequence; all four couples can be connected without crossing bridges. |
| 1 2 3 4  4 3 2 1 | 1 | Once any bridge is built, no others can be built without crossing it. |
| 1  1 1 1 | 3 |  |
| 2 5 3 3 3 1 8 2 6 7 6  1 2 5 3 4 1 7 8 2 5 6 | 10 |  |