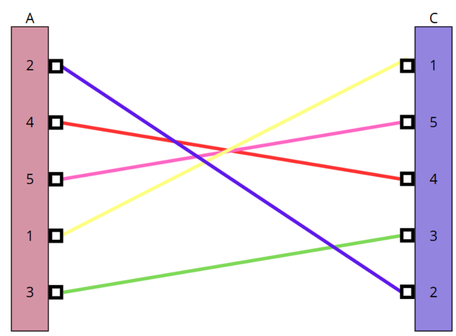
Problem D - Desembaraçar Cabos

> Check our instructions page for detailed information on the qualification and the format of this problem.

Tiago is trying to build his own computer, and now he just needs to connect some cables. He has two columns of $\boldsymbol{N}$ entries each, $\boldsymbol{A}$ and $\boldsymbol{C}$, both with entries numbered from 1 to $\boldsymbol{N}$ in distinct positions $a_1, a_2, \ldots, a_n$ and $c_1, c_2, \ldots, c_n$ respectively. Tiago's task is to connect the i-th entries of each column, located at positions a_i and c_i, respectively.

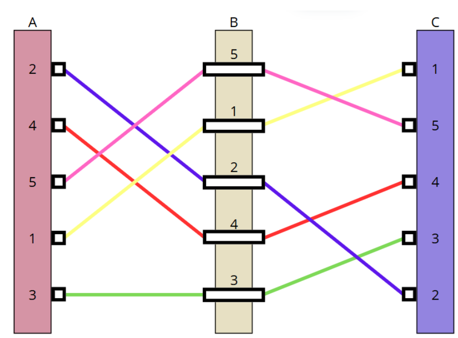
To clarify, when we say $a_i = j$, it means that the entry in row j of column $\boldsymbol{A}$ is labeled with the number i. Similarly, if $c_i = k$, it means the entry in row k of column $\boldsymbol{C}$ has label i. Tiago's task is to connect all corresponding entries from the two columns. More precisely, for each label i (with $1 \leq i \leq N$), he needs to connect row j of column $\boldsymbol{A}$ (where $a_i = j$) to row k of column $\boldsymbol{C}$ (where $c_i = k$).

However, he noticed that since the columns were not arranged in the same order, there were many cable crossings, creating considerable confusion and potentially becoming dangerous.



For example, connecting columns A and C with entries $1, 2, 3, 4, 5$ at positions $\{4, 1, 5, 2, 3\}$ and $\{1, 5, 4, 3, 2\}$ respectively, as shown in the image to the right (counting from top to bottom), resulted in 7 cable crossings, 6 of which were very close to each other.

Luís then recommended adding an intermediate column $\boldsymbol{B}$ with $\boldsymbol{N}$ double entries, to space out the crossings. Tiago could freely choose the positions of these entries to minimize confusion.



For instance, if Tiago chooses the following numbering: $b_1 = 2, b_2 = 3, b_3 = 5, b_4 = 4, b_5 = 1$, he manages to reduce crossings to just 4 between columns $\boldsymbol{A}$ and $\boldsymbol{B}$, and 3 crossings between columns $\boldsymbol{B}$ and $\boldsymbol{C}$. Formally, we say cables i, j cross between columns $\boldsymbol{A}$ and $\boldsymbol{B}$ if $a_i > a_j$ and $b_i < b_j$ or vice versa, and that they cross between columns $\boldsymbol{B}$ and $\boldsymbol{C}$ if $b_i > b_j$ and $c_i < c_j$.

Tiago's goal is to minimize the number of crossings on each side, ensuring neither side has too many crossings. Explicitly, he wants to minimize the following value:

$$\max\{\text{crossings between columns A and B}, \text{crossings between columns B and C}\}$$

Part I

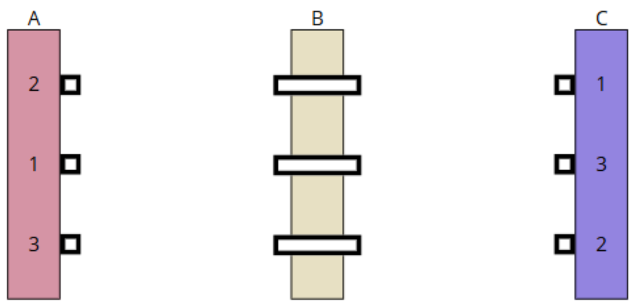
In Part I of this problem, your goal is to help Tiago calculate the optimal value for an unknown configuration of column $\boldsymbol{B}$; that is, you must calculate:

$$\max\{\text{crossings between columns A and B}, \text{crossings between columns B and C}\}$$

However, you don't have to find values for $b_1, b_2, b_3, \ldots, b_N$ that satisfy the equality.

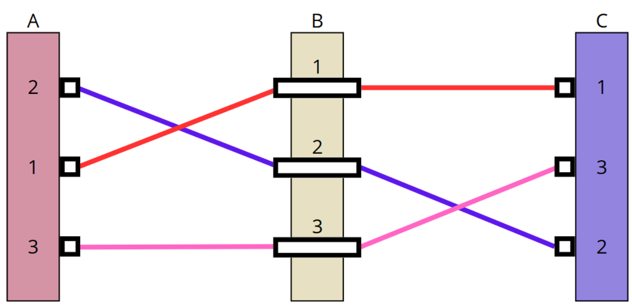
Example

Suppose we have $\boldsymbol{N} = 3, a_1 = 2, a_2 = 1, a_3 = 3, c_1 = 1, c_2 = 3$, and $c_3 = 2$, as illustrated in the image.



Clearly, we need at least one crossing in total. The question is whether we can have at most one crossing on each side.

Indeed, we can show this is possible; for instance, by setting $b_1 = 1, b_2 = 2$, and $b_3 = 3$.



However, note that for this Part, it would be sufficient to output 1; we do not need to indicate the values of b_1, b_2, b_3 that achieve this value.

Constraints

The following limits are guaranteed in all test cases provided to the program for this Part:

$1 \leq \boldsymbol{N} \leq 10^5$ Number of entries in each column

The test cases for this Part of the problem are organized into a single group:

Subtask	Points	Additional Constraints
1	20	No additional restrictions

Part II

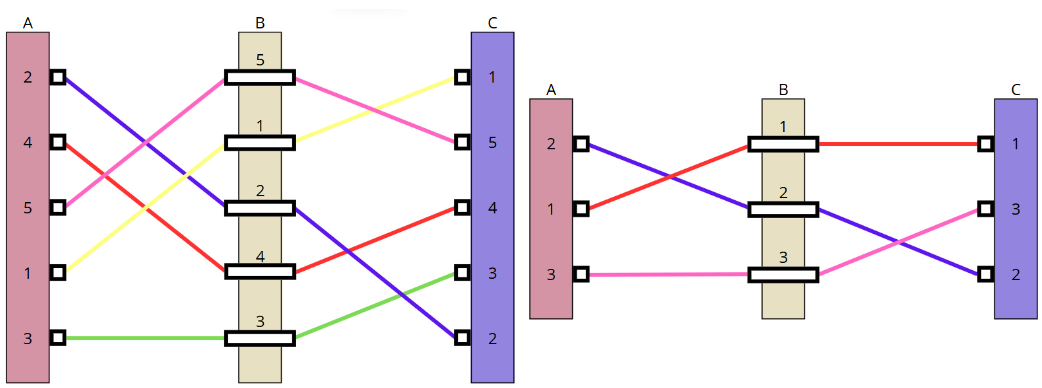
Now that we know the value to achieve, Tiago wants your help to find an optimal configuration for column $\boldsymbol{B}$. That is, given two columns $\boldsymbol{A}$ and $\boldsymbol{C}$ defined by $a_1, a_2, a_3, \ldots, a_n$ and $c_1, c_2, c_3, \ldots, c_N$, find values $b_1, b_2, b_3, \ldots, b_N$ for column $\boldsymbol{B}$ that minimize:

$$\max\{\text{number of crossings between columns A and B}, \text{number of crossings between columns B and C}\}$$

Example

Consider the examples already given in the statement, that is, $\boldsymbol{N} = 5$, $a_1 = 4, a_2 = 1, a_3 = 5, a_4 = 2, a_5 = 3, c_1 = 1, c_2 = 5, c_3 = 4, c_4 = 3, c_5 = 2$, and $\boldsymbol{N} = 3, a_1 = 2, a_2 = 1, a_3 = 3, c_1 = 1, c_2 = 3, c_3 = 2$.

As previously seen, these cases could have possible solutions: $b_1 = 2, b_2 = 3, b_3 = 5, b_4 = 4, b_5 = 1$ and $b_1 = 1, b_2 = 2, b_3 = 3$ respectively, but note that these aren't necessarily the only solutions.



Constraints

The limits guaranteed in all test cases provided to the program in Part II are identical to those in Part I:

$1 \leq \boldsymbol{N} \leq 10^5$ Number of entries in each column

The test cases for this Part of the problem are organized into three groups with different additional restrictions:

Subtask	Points	Additional Constraints
3	20	$\boldsymbol{N} \leq 8$
4	30	$\boldsymbol{N} \leq 500$
5	30	No additional restrictions

Input Format

The first line contains an integer $\boldsymbol{P}$, indicating the Part this test case represents (1 for Part I, 2 for Part II).

Next line contains an integer $\boldsymbol{N}$, representing the number of entries in each column.

Next two lines each contain $\boldsymbol{N}$ numbers:

- The first line has positions $a_1, a_2, \ldots, a_N$ of column $\boldsymbol{A}$.
- The second line has positions $c_1, c_2, \ldots, c_N$ of column $\boldsymbol{C}$.

Output Format

For Part I, output only one integer, the optimal value of:

$$\max\{\text{crossings between columns A and B}, \text{crossings between columns B and C}\}$$

For Part II, output a single line with $\boldsymbol{N}$ integers $b_1, b_2, \ldots, b_N$, separated by spaces, which minimize the above expression.

If multiple optimal solutions exist, output any one of them.

Note: Answers may exceed $2^{31} - 1$. Use `long long int` in `C/C++`, `long` in `Java`, or `Longint` in `Pascal`.

Example 1 Input

```
1
5
4 1 5 2 3
1 5 4 3 2
```

Example 1 Output

```
4
```

Example 1 Description

This example corresponds to Part I applied to the first example in the problem statement.

Example 2 Input

```
1
3
2 1 3
1 3 2
```

Example 2 Output

```
1
```

Example 2 Description

This example corresponds to the example given in Part I of the problem statement.

Example 3 Input

```
2
3
2 1 3
1 3 2
```

Example 3 Output

```
1 2 3
```

Example 3 Description

This example corresponds to Part II applied to the previous example.

Example 4 Input

```
2
5
4 1 5 2 3
1 5 4 3 2
```

Example 4 Output

```
2 3 5 4 1
```

Example 4 Description

This example corresponds to Part II applied to the first example in the problem statement.