***Minimum operations***

C371\_Coding\_October2022

**Topic**: Queue

**Difficulty Level:** Easy

**Question / Problem Statement**:

Caleb gives Jason two arrays, **A** and **B** each of size **N** consisting of the first **N** positive integers each exactly once, that is, they are permutations.

Jason's task is to find the minimum number of operations required to make both the arrays **A** and **B** empty.

The following two types of operations can be performed any number of times:

* Jason is allowed to rotate the array **A** clockwise.
* when the first element of both the arrays is the same, then they are removed from both the arrays and the process continues.

Write a program to find the minimum number of operations performed by Jason to make both the arrays **A** and **B** empty.

**Note**

**A[i]** and **B[i]** range from 1 to **N** (both included) in any order.

**Function Description**

In the provided code snippet, implement the provided **minimumOperationOnArray(...)** method using the variables to find the minimum number of operations performed by Jason to make both the arrays **A** and **B** empty. You can write your code in the space below the phrase **“WRITE YOUR LOGIC HERE”**.   
  
There will be multiple test cases running so the Input and Output should match exactly as provided.  
The base Output variable **result** is set to a default value of **-404** which can be modified. Additionally, you can add or remove these output variables.

**Input Format**

First line contains an integer **N**.

Second line contains **N** integers separated by a single space - array **A**.

Third line contains **N** integers separated by a single space - array **B**.

**Sample Input**

3 –denotes N.

1 3 2 –denotes array A.

2 3 1 –denotes array B.

**Constraints**

1 <= **N <=** 100.

**Output Format**

Output should return the minimum number of operations performed by Jason to make both the arrays **A** and **B** empty.

**Sample Output**

6

**Explanation**

Perform operation 1 to make A = [3, 2, 1].

Perform operation 1 to make A = [2, 1, 3].

Now perform operation 2 to make A = [1, 3] and B = [3, 1].

Perform operation 1 to make A = [3, 1].

Now perform operation 2 to make A = [1] and B = [1].

Now perform operation 2 to make A = [] and B = [].

Total operation performed by Jason to make both array A and B empty is 6.

**Solution Steps**

1. This problem can be easily solved using a well known data structure known as queue.

2. First insert all the elements of array A and B in the queue qa and qb.

3. Now, start iterating through the queue qa from the front and check if the front element of the queue qa and front element of queue qb are similar or not, if they are same then pop the front element of qa and qb, else remove the front element of queue qa and push it again in queue qa, repeat this process until the queue qa becomes empty.

4. Finally return the count of operations performed.

**Running Solution in C++**:

#include<bits/stdc++.h>

using namespace std;

int main(){

int N;

cin>>N;

//create array A.

int A[N];

for(int idx=0;idx<N;idx++)

cin>>A[idx];

//create array B.

int B[N];

for(int idx=0;idx<N;idx++)

cin>>B[idx];

//push elements of array A and B to queue qa, qb.

queue<int> qa,qb;

for(int idx=0;idx<N;idx++){

qa.push(A[idx]);

qb.push(B[idx]);

}

//calculate the min\_cnt.

int min\_cnt=0;

while(!qa.empty()){

//if both values match, remove from the array.

if(qa.front()==qb.front()){

qa.pop();

qb.pop();

}

//else remove and again push back to qa.

else{

int front\_val=qa.front();

qa.pop();

qa.push(front\_val);

}

min\_cnt++;

}

cout<<min\_cnt<<"\n";

}

Input:

10

5 4 8 9 1 6 3 2 7 10

1 6 8 9 5 4 10 3 2 7

Output:

27

**Test Cases [ Qty: 12]**

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case No** | **Input** | **Output** | **Score** |
| 1 | 3  1 3 2  2 3 1 | 6 | 0 |
| 2 | 10  5 4 8 9 1 6 3 2 7 10  1 6 8 9 5 4 10 3 2 7 | 27 | 0 |
| 3 | 15  5 11 12 13 15 6 14 2 7 10 4 8 9 3 1  4 15 8 2 6 9 11 10 7 5 13 14 3 1 12 | 75 | 1 |
| 4 | 30  5 29 12 16 25 17 18 30 27 10 4 23 20 3 1 24 26 19 14 9 6 22 8 13 15 21 28 7 11 2  17 20 6 18 21 5 22 24 28 7 23 3 27 19 10 30 15 25 12 16 2 1 11 9 4 8 29 14 13 26 | 226 | 1 |
| 5 | 40  5 29 12 16 25 36 18 37 27 32 34 40 20 3 1 24 26 19 33 9 6 22 8 13 15 21 28 7 11 2 31 39 14 38 4 17 30 35 10 23  14 39 4 5 23 7 40 8 36 17 30 21 3 35 33 32 12 16 20 25 31 13 22 34 19 18 29 11 27 15 1 38 26 6 28 10 9 37 2 24 | 491 | 1 |
| 6 | 60  5 52 12 16 25 36 18 37 27 51 34 40 20 3 48 57 60 19 33 41 6 22 58 13 15 43 28 7 46 59 31 39 14 38 4 55 56 54 10 23 9 24 21 35 2 11 32 1 47 42 49 29 53 44 17 30 50 8 45 26  55 15 44 60 7 33 14 27 16 1 25 59 48 18 35 39 30 37 11 34 45 17 22 28 21 5 23 41 51 26 13 19 36 3 12 24 53 43 38 4 10 31 57 20 29 49 2 6 46 56 40 52 54 42 47 9 58 8 50 32 | 1056 | 1 |
| 7 | 70  5 52 12 70 25 36 18 37 27 51 68 40 20 3 48 57 60 19 33 41 6 22 58 13 15 43 28 63 64 59 31 39 14 69 4 55 56 65 10 23 67 24 21 35 2 11 32 1 47 42 66 29 53 44 17 61 50 8 45 26 30 62 7 46 54 49 9 34 38 16  46 8 20 39 51 34 59 22 21 5 66 41 11 65 33 44 2 40 68 31 58 50 23 29 4 38 17 49 30 48 67 18 27 62 60 26 14 64 28 7 56 12 63 16 69 35 43 15 36 45 55 19 54 61 9 3 37 1 53 10 47 52 42 24 32 6 57 70 25 13 | 1515 | 1 |
| 8 | 80  5 52 12 70 25 36 18 37 27 51 68 40 20 3 76 57 60 19 33 41 6 22 58 13 15 43 28 63 64 59 31 79 14 69 4 55 72 65 10 23 67 24 21 80 2 75 74 1 47 42 66 29 53 44 17 61 50 8 45 73 30 62 7 46 54 77 9 34 38 16 26 56 71 32 78 48 49 11 39 35  38 51 9 47 3 45 72 46 30 23 13 35 52 80 70 39 14 75 62 29 55 78 17 73 67 42 60 10 7 24 50 26 20 61 33 76 2 21 16 15 22 32 43 18 41 28 79 48 68 36 12 27 54 49 6 34 64 31 53 19 57 58 74 40 44 63 59 37 69 5 8 1 77 11 25 4 71 65 66 56 | 1733 | 1 |
| 9 | 90  5 52 12 70 25 36 18 37 27 51 68 40 84 3 76 57 60 19 33 41 6 87 58 13 15 43 28 63 64 59 31 79 14 69 4 88 72 65 10 23 67 81 21 80 90 82 74 1 47 42 89 29 53 44 17 61 50 8 85 73 30 62 7 46 54 77 9 34 38 16 26 56 71 32 83 48 49 11 39 35 24 75 78 20 86 45 22 55 66 2  3 15 86 44 14 46 32 58 53 69 20 56 50 88 83 62 80 34 19 81 31 65 45 26 10 74 72 22 9 38 64 7 90 42 18 30 78 39 21 63 36 54 84 47 24 85 82 68 48 67 37 25 28 61 60 49 29 23 57 55 35 16 4 1 5 12 13 17 59 70 33 66 41 51 11 73 6 76 52 8 89 87 75 43 71 2 79 77 40 27 | 2304 | 1 |
| 10 | 100  93 52 12 70 25 36 18 37 27 99 68 40 84 3 76 57 60 19 33 41 92 87 58 13 15 43 28 63 64 59 31 97 14 69 4 88 72 65 10 23 67 81 21 80 90 82 74 1 95 42 89 29 53 44 17 61 50 8 85 73 30 62 7 46 54 77 9 34 38 16 26 56 71 32 83 48 49 11 91 35 24 75 78 20 86 45 94 55 98 2 39 96 5 22 100 6 79 66 51 47  40 6 94 44 48 5 28 68 77 78 18 47 4 92 37 30 10 50 42 38 100 88 31 65 32 8 81 82 49 27 60 29 69 41 70 1 33 87 26 54 89 67 39 12 99 24 34 14 51 64 53 11 15 23 58 63 43 71 74 22 97 2 83 55 93 98 80 79 84 72 45 36 75 95 61 56 25 86 9 91 35 76 66 7 73 3 90 21 57 59 52 17 62 96 16 85 19 13 46 20 | 2615 | 1 |
| 11 | 1  1  1 | 1 | 1 |
| 12 | 2  1 2  2 1 | 3 | 1 |

Plagiarism found – No

Clarity of the problem statement - Yes

Clarity of the example in the problem statement - Yes

Clarity of sample test cases - Yes

Clarity of test cases (Dual output) – Yes

Clarity of explanations - Yes

Provided Solution running – Yes

EEOC complaint (using abusive words/Indian Names/) - No

Similar Question in System - No

Difficulty Level – Easy

Question w.r.t Queue concepts- Yes

Final Comment: **Accepted**