Question 1

Due on 2019-08-29, 23:59 IST
Given two arrays of integers output the smallest number in the first array not present in the second one.

Input Specification:
The first line contains the size $N_1$ of the first array. Next line give the contents of the first array. Next line contains the size $N_2$ of the second array. Next line give the contents of the second array.

Output Format:
Output must be a single number which is the smallest number occurring in the first array that does not occur in the second. In case there is no such number, output NO.

Variable Constraints:
The sizes of the arrays are smaller than 20. Each array entry is an integer which fits an int data type.

Example:
Input:
3
2 3 4
4
1 3 5 7

Output: 2

Input
1
1
2
1 2

Output: NO

Sample Test Cases

<table>
<thead>
<tr>
<th>Test Case 1</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1 2 3 4 5</td>
</tr>
<tr>
<td></td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>2 7 9</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Test Case 2</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>3 5 7 9</td>
</tr>
<tr>
<td></td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>7 9 10 55</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Output</th>
<th>1</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Output</th>
<th>3</th>
</tr>
</thead>
</table>
The due date for submitting this assignment has passed.
As per our records you have not submitted this assignment.

Sample solutions (Provided by instructor)

```c
#include<stdio.h>
#define MAX 20

int read_array(int arr[])
{
    int i, n;
    scanf("%d", &n);
    for (i = 0; i < n; i++)
        scanf("%d", &arr[i]);
    return n;
}

int present(int arr[], int n, int elt) {
    int i;
    for (i = 0; i < n; i++) {
        if (arr[i] == elt) {
            return 1;
        }
    }
    return 0;
}

int main() {
    int arr1[MAX], n1;
    int arr2[MAX], n2;
    n1 = read_array(arr1);
    n2 = read_array(arr2);
    int i, small_np = 0, flag = 0;
    for (i = 0; i < n1; i++){
        if (!present(arr2, n2, arr1[i])) {
            if (!flag || (small_np > arr1[i])) {
                flag = 1;
                small_np = arr1[i];
            }
        }
    }
    if (flag) {
        printf("%d", small_np);
    } else {
        printf("NO");
    }
    return 0;
}
```
return 0;
}